 **`**

**Deliverable 02**

**Functional Specification**

The **functional specification** document covers the most important parts of **what** our system, **Aegis**, will accomplish to meet the business requirements provided for the client, **Jesster FIT**. The **overview of this document** covers a full set of **use case diagrams** and **narratives** which summarizes how each subsystem will operate, **process models** illustrate the various processes within the system along with **input, output and validation specifications.**

Table of Contents

[2. 1. Deliverable 02 Introduction 2](#_Toc8619682)

[3. Use Cases 5](#_Toc8619683)

[4. Use Case Diagrams 6](#_Toc8619684)

[1. Admin Subsystem 6](#_Toc8619685)

[2. Product Subsystem 7](#_Toc8619686)

[3. User Subsystem 8](#_Toc8619687)

[4. Employee Subsystem 9](#_Toc8619688)

[5. Client Subsystem 10](#_Toc8619689)

[6. Reporting Subsystem 11](#_Toc8619690)

[7. Sale Subsystem 11](#_Toc8619691)

[8. Exercise Subsystem 12](#_Toc8619692)

[8. Exercise Subsystem Part 2 13](#_Toc8619693)

[13](#_Toc8619694)

[5. Use Case Narratives 14](#_Toc8619695)

[High Level Data Flow Diagrams 210](#_Toc8619696)

1. Deliverable 02 Introduction

##### Synopsis of previous deliverable: Deliverable 01

The previous deliverable required the team to present their project proposal. A brief overview of the document contents will be provided as it serves as the base for deliverable 02 – Functional Specification.

**Client Information:**

The company name is JessterFIT. It was established by Jesse Pretorius August 2018. There are brand ambassadors that promote the company on social media. The marketing of the company is mostly on social media. The company offers workouts for its client. They also sell merchandise such as hoodies and vests. They provide exercise plans as well as challenges that clients can sign up for and win prizes. Clients can sign up for the subscription which grants them additional advantages.

**Project Request:**

The system is designed to solve data management problems currently faced by the client. The solution presented will put the company at a competitive advantage. The company currently works from the owner’s laptop. Work is assigned to brand ambassadors by the owner. The solution we present will organize all management reports and essential information in a single place, separating it from the day to day operational sections of the systems for the technicians to manage. The system will keep track of client information, subscriptions and challenges the client is registered for. It will organize all information that the owner will use to make their company more competitive.

**Preliminary** **Investigation:**

Team Aegis conducted a preliminary investigation using a **problem statement matrix** with a brief description of current problems in the system. The company currently works from the owner’s laptop and uses his personal email account to store all relevant company information. Preliminary constraints include budget and scope as the main areas to focus on through-out this project.

**Problem Analysis:**

Team Aegis used a **problems, opportunities, objectives or constraints matrix**. The results of the matrix identified that dynamically creating an exercise plan, challenge creation and keeping track of clients’ progress are key requirements to be addressed by the proposed system

**Requirement** **Analysis:**

A complete list of user requirements totaling to roughly 88 requirements which are divided into 8 subsystems. The subsystems include: **Admin, Product, User, Employee, Client, Reporting, Sale** and **Exercise.**

**Decision** **Analysis**:

The decision analysis incorporates the purpose, background and scope of the **feasibility analysis**. The feasibility analysis is completed according to a feasibility matrix containing feasibility descriptions for operational, technical, legal, economic and schedule constraints. Detailed research of alternatives considered, and recommendations are based of the competed feasibility analysis

**Appendices**:

Client documents, other systems investigated, and complexity requirements and a client sign-off form is added to the appendix

##### Synopsis of current deliverable: Deliverable 02

This deliverable reveals the **functional specifications** of the proposed system. This document illustrates the proposed system’s capabilities but will not focus on any technical aspects of the capabilities. This document leaves the interested stakeholders with an in-depth understanding of **what** the system will be able to do and how the data will be transformed for better decision making. A description of the different actor-system interactions and expected system response is provided via Systems Analysis Methods:

A complete set of use **case diagrams** along with their accompanying use **case narratives** to describe the roles of each actor and system requirements needed to complete a use case.

**Process models** display the flow of information into and out of the system between use cases, actors and entities

A complete, fully attributed **logical data model** demonstrating how the data will be structured and stored with its relevant attributes.

The final section of the document will contain an overview of all inputs, outputs and validation of the functional requirements required by the system established in deliverable 01 and updated for deliverable 02. A complexity matrix is added to show the level of complexity Aegis is aiming to achieve.

Use Cases

##### Introduction

The following part of the functional specification document contains two main sections namely:

Complete set of UML Use Case Diagrams.

* + Illustrating actor roles in the Jesster FIT system.

Complete list of Use Case Narratives

* + Containing details of each activity and process needed to complete a use case. Each process within the use case is explained step by step detailing how the actors and system interact and respond to each other.

Use Case Diagrams

1. Admin Subsystem

2. Product Subsystem



3. User Subsystem



4. Employee Subsystem



5. Client Subsystem



6. Reporting Subsystem



7. Sale Subsystem



8. Exercise Subsystem



8. Exercise Subsystem Part 2



Use Case Narratives

##### 1. Admin Subsystem

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Iain Steyn**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Add New Subscription Type** | | | **Use Case Type:** |
| **USE CASE ID:** | 1.1 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee would like to add a new subscription type to the system. The employee will add a new subscription type on the add new subscription type page which will be displayed by the system when the employee wants to add a new subscription type. The employee will add all the required details for the new subscription type such as:  **Subscription\_Type\_Description,**  **Subscription\_Type\_Price,**  which will be saved in the **Subscription\_Type table.** The system will then capture and validate the entered details and check to see if the subscription type - provided by the user – doesn’t already exist on the system. If it does not exist, the system will save it as a new type and display a success message to the employee. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to add a new subscription type to the system. | | | |
| **TRIGGER:** | The employee selects to add a new subscription type to the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee requests to add a new subscription type to the system. | | **Step 2:** The system will display the add new subscription type screen and inform the employee to enter all the required subscription type details such as:  **Subscription\_Type\_Description,**  **Subscription\_Type\_Price.** | |
|  | **Step 3:** The employee will enter all the required details such as:  **Subscription\_Type\_Description**  **Subscription\_Type\_Price**  And submit them to the system. | | **Step 4:** The system will capture the entered details:  **Subscription\_Type\_Description,**  **Subscription\_Type\_Price.** | |
|  |  | | **Step 5:** The system will validate the entered subscription type details so that they are valid details. | |
|  |  | | **Step 6:** The system will check to see if the type added does not already exist in the system. | |
|  |  | | **Step 7:** The system will automatically generate a unique **Subscription\_Type\_ID** and add it to the **Subscription\_Type table.** | |
|  |  | | **Step 8:** The system will save all the details to the **Subscription\_Type table** as:  **Subscription\_Type\_Description,**  **Subscription\_Type\_Price.** | |
|  |  | | **Step 9:** The system will display a success message to the employee. | |
| **ALTERNATE COURSES:** | **Alt Step 5:** One or more of the entered subscription type details are invalid so the system will display an error message.   * Return the user to step 3. | | | |
| **CONCLUSION:** | The use case concludes when a new subscription type has been successfully added to the system and a success message is displayed for the employee. | | | |
| **POST-CONDITION:** | A new subscription type has been successfully added to the system. | | | |
| **BUSINESS RULES** | Only an employee with the required access level can add a subscription type. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None | | | |
| **ASSUMPTIONS:** | The subscription should be approved by the owner | | | |
| **OPEN ISSUES:** | None | | | |

Table 1.1.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Iain Steyn** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Maintain Subscription Type** | | | **Use Case Type:** |
| **USE CASE ID:** | 1.2 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee would like to maintain a subscription type on the system. The employee will request to maintain the subscription type and the system will display the maintain subscription type page. The system will then prompt the user to search for an existing subscription type on the maintain subscription type page and invoke use case **1.18 Search Subscription Type** and display the output on the page. The employee will then select the type they would like to maintain by selecting either the edit or delete option. All changes will be saved on the system and a success message will be displayed. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to maintain a subscription type. | | | |
| **TRIGGER:** | The employee selects to edit a subscription type from the edit subscription type option. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee requests to maintain a subscription type from the maintain subscription option. | | **Step 2:** The system will display the maintain subscription type page and prompt the employee to search for an existing subscription type on the page. | |
|  |  | | **Step 3:** The system will invoke use case **1.3 Search Subscription Type** and display all the results to the user in a list on the maintain subscription type page as well as an edit and remove option for each type in the list. | |
|  | **Step 4:** The employee will select the edit subscription type option for the subscription type that they would like to edit. | | **Step 5:** The system will retrieve all the details gathered from use case **1.3 Search Subscription** **Type** for the selected type such as:  **Subscription\_Type\_Description,**  **Subscription\_Type\_Price,**  and display it on the maintain subscription type page into an editable field for each type detail. | |
|  | **Step 6:** The employee will update the subscription type and save it to the system. | | **Step 7:** The system will capture the edited type as: **Subscription\_Type\_Description,**  **Subscription\_Type\_Price.** | |
|  |  | | **Step 8:** The system will validate the type so that it is a is a valid description. | |
|  |  | | **Step 9:** The system will then update the type for the selected **Subscription\_Type\_ID** in the **Subscription\_Type table**. | |
|  |  | | **Step 10:** The system will display a success message to the employee.   * Return the employee to the home page. | |
| **ALTERNATE COURSES:** | **Alt Step 4a:** The employee selects the remove subscription type option for the subscription type that they would like to remove. | | | |
|  | **Alt Step 4b:** The system prompts the user to confirm if they would like to delete the selected subscription type. | | | |
|  | **Alt Step 4c:** The user confirms that they would like to delete the selected subscription type. | | | |
|  | **Alt Step 4cc:** The User revokes the decision to remove the selected subscription type and selects the option to cancel confirmation.   * Return to Alt Step 4b. | | | |
|  | **Alt Step 4d:** The system removes the selected subscription type and its relevant details from the **Subscription\_Type table**, display an error message   * Continue at step 10. | | | |
|  | **Alt Step 8:** The updated subscription type description is not a valid value, display an error message.   * Return to step 6. | | | |
| **CONCLUSION:** | The selected subscription type has been successfully maintained and all changes have been saved on the system and a success message has been displayed. | | | |
| **POST-CONDITION:** | Subscription type details have been successfully maintained. | | | |
| **BUSINESS RULES** | Only an employee with the required access levels can maintain a subscription type. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The subscription type exists on the system | | | |
| **OPEN ISSUES:** | None. | | | |

Table 1.1.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| JessterFIT | | Date: 2019/03/30 | | |
| Authors(s): Iain Steyn | | Version 1.3 | | |
| **USE CASE NAME:** | **Search Subscription Type** | | | **Use Case Type:** |
| **USE CASE ID:** | 1.3 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee needs to search for a subscription type on the system. The user will search for a subscription type by entering a keyword or phrase that matches any existing:  **Subscription\_Type\_ID,**  **Subscription\_Type\_Description,**  **Subscription\_Type\_Price,**  in the **Subscription\_Type table**, into the provided (by the system) search bar.The system will then verify and search for any existing subscription type and display a subscription type that matches the given search criteria. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to search a subscription type on the system. | | | |
| **TRIGGER:** | The employee needs to search for a subscription type on the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee needs to search for a subscription type on the system. | | **Step 2**: The system displays the search criteria and prompts the employee to enter a keyword or phrase | |
|  | **Step 3:** The employee will enter any keyword or phrase that will match any existing:  **Subscription\_Type\_ID,**  **Subscription\_Type\_Description,**  **Subscription\_Type\_Price,**  in the **Subscription\_Type table** on the system. | |  | |
|  | **Step 4:** The employee will request to search. | | **Step 5:** The system will capture the entered search keyword or phrase. | |
|  |  | | **Step 6:** The system will validate the entered search criteria so that it is a valid search keyword or phrase. | |
|  |  | | **Step 7:** The captured keyword or phrase is valid, and the system will check the:  **Subscription\_Type\_ID,**  **Subscription\_Type\_Description,**  **Subscription\_Type\_Price,**  details for all existing subscription types in the **Subscription\_Type table** to see if any match the captured search criteria. | |
|  |  | | **Step 8:** The system returns a list of all subscription type(s) matching the given search criteria. | |
| **ALTERNATE COURSES:** | **Alt Step 6:** The captured keyword or phrase is invalid, display an error message.   * Return to step 2. | | | |
|  | **Alt Step 7:** No values on the system match the given search criteria. The system will display an error message.   * Return to step 2. | | | |
| **CONCLUSION:** | The system has successfully searched for the required subscription type based off the given search criteria and displayed the results to the employee. | | | |
| **POST-CONDITION:** | The system successfully retrieves the subscription type/s on the system. | | | |
| **BUSINESS RULES** | Only an employee with the required access level can search for a subscription type. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | None. | | | |
| **OPEN ISSUES:** | None. | | | |
|  |  | | | |

Table 1.1.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/04/12** | | |
| **Authors: Iain Steyn**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Add new Challenge** | | | **Use Case Type:** |
| **USE CASE ID:** | 1.4 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: **◻** |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None | | | |
| **DESCRIPTION:** | This use case describes when the employee requests to add a new challenge to the system. The employee must be logged into the system and have the required access level functionality to be able to select the option to a add a new challenge, then enter details:  **Challenge\_Start\_Date, Challenge\_End\_Date,**  which will be added to the **Challenge table**. The system will check if the details – provided by the user – are valid. The use case ends when the challenge is added to the system or if the process is cancelled. A success message will be displayed if the use case has been completed successfully | | | |
| **PRE-CONDITION:** | The challenge must be approved by the owner.  -The user must have the required access level | | | |
| **TRIGGER:** | The employee selects to add a new challenge to the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1:** The employee requests to add a new challenge to the system. | |  | |
|  |  | | **Step 2**: The system prompts the employee to enter the required details:  **Challenge\_Start\_Date** and **Challenge\_End\_Date.** | |
|  | **Step3:** The employee enters the required details. | |  | |
|  |  | | **Step 4**: The system captures the data entered by the employee. | |
|  |  | | **Step 5:** The system will validate the entered details to ensure all details are valid challenge details. | |
|  |  | | Step 6: The system will check to see if the details added does not already exist in the system. | |
|  |  | | **Step 7:** The system will automatically generate a unique **Challenge\_ID** and add it to the **Challenge table**. | |
|  |  | | **Step 8:** The system will add the challenge details to the **Challenge table** and save it as a new challenge. | |
|  |  | | **Step 9:** The system will display a success message to the employee and return the employee to the challenge page. | |
| **ALTERNATE COURSES:** | **Alt Step 5:** The entered **Challenge\_Start\_Date**, **Challenge\_End\_Date** are invalid. The system will display an error message.  Return the user to step 3. | | | |
|  | **Alt step 6:** Challenge details already exist on the system. Error message return to step 3. | | | |
| **CONCLUSION:** | The use case ends when a new challenge has been successfully added to the system and a success message is displayed for the employee. | | | |
| **POST-CONDITION:** | A new challenge has been successfully added to the system. | | | |
| **BUSINESS RULES** | Only employees with the required access level can add a new challenge to the system. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | None. | | | |
| **OPEN ISSUES:** | None. | | | |

Table 1.1.4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/04/12** | | |
| **Authors: Iain Steyn**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Maintain challenge** | | | **Use Case Type:** |
| **USE CASE ID:** | 1.5 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis:◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee would like to maintain a challenge on the system. The employee will request to maintain a challenge and the system will display the maintain challenge page. The system will then prompt the user to search for an existing challenge by invoking use case **1.6 Search Challenge** and display the output on the page. The employee will then select the challenge they would like to maintain by selecting either the edit or delete option. All changes will be saved on the system and a success message will be displayed. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to maintain a challenge. | | | |
| **TRIGGER:** | The employee selects to edit a challenge from the edit challenge option. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee requests to maintain a challenge from the maintain challenge option. | | **Step 2:** The system will display the maintain challenge page and prompt the employee to search for an existing challenge on the system. | |
|  |  | | **Step 3:** The system will invoke use case **1.6** **Search Challenge** and display all the results to the user in a list on the maintain product as well as an edit and remove option for each product in the list. | |
|  | **Step 4:** The employee will select the edit challenge option for the challenge they would like to edit. | | **Step 5:** The system will retrieve all the details gathered from use case 1.6 **Search Challenge** for the selected challenge such as: **Challenge\_ID, Challenge\_Start\_Date and Challenge\_End\_Date** and display it on the maintain challenge page. | |
|  | **Step 6:** The employee will update the challenge and save it to the system. | | **Step 7:** The system will capture the edited challenge details: **Challenge\_ID, Challenge\_Start\_Date and Challenge\_End\_Date**. | |
|  |  | | **Step 8:** The system will validate the challenge to ensure they are valid challenge details. | |
|  |  | | **Step 9:** The system will then update the challenge for the selected **Challenge\_ID** in the **Challenge** **table**. | |
|  |  | | **Step 10:** The system will display a success message to the employee.  Return the employee to the home page. | |
| **ALTERNATE COURSES:** | **Alt Step 4a:** The employee selects the remove challenge option for the challenge that they would like to remove. | | | |
|  | **Alt Step 4b:** The system prompts the user to confirm if they would like to delete the selected challenge. | | | |
|  | **Alt Step 4c:** The user confirms that they would like to delete the selected challenge. | | | |
|  | **Alt Step 4cc:** The User revokes the decision to remove the selected challenge and selects the option to cancel confirmation.  Return to Step 3. | | | |
|  | **Alt Step 4d:** The system removes the selected challenge and its relevant details from the **Challenge table.**  Continue at step 10. | | | |
|  | **Alt Step 8:** The updated product details not a valid value.  Return to step 6. | | | |
| **CONCLUSION:** | The use case ends when a new challenge has been successfully added to the system and a success message is displayed to the employee. | | | |
| **POST-CONDITION:** | A new challenge has been successfully added to the system. | | | |
| **BUSINESS RULES** | None | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None | | | |
| **ASSUMPTIONS:** | None | | | |

Table 1.1.5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Joshua Eales**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Search Challenge** | | | **USE CASE TYPE** |
| **USE CASE ID:** | 1.6 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis:◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee needs to search for a challenge on the system. The user will search for a challenge by entering a keyword or phrase that matches any existing:  **Challenge\_ID, Challenge\_Start\_Date, Challenge\_End\_Date**  in the **Challenge table**, into the provided (by the system) search bar.The system will then verify and search for any existing challenge and display any challenges that matches the given search criteria. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the needed access levels and functionality to be able to search a challenge on the system. | | | |
| **TRIGGER:** | The employee needs to search for a challenge on the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee needs to search for a challenge on the system. | |  | |
|  | **Step 2:** The employee will enter any keyword or phrase that will match any existing: **Challenge\_ID, Challenge\_Start\_Date, Challenge\_End\_Date**  in the **Challenge table** on the system. | |  | |
|  | **Step 3:** The employee will request to search. | | **Step 4:** The system will capture the entered search keyword or phrase. | |
|  |  | | **Step 5:** The system will validate the entered search criteria so that it is a valid search keyword or phrase. | |
|  |  | | **Step 6:** The system will check the:  **Challenge\_ID, Challenge\_Start\_Date, Challenge\_End\_Date**  details for all existing challenges in the **Challenge table** to see if any match the captured search criteria. | |
|  |  | | **Step 7:** The system returns a list of the challenge(s) matching the given search criteria. | |
| **ALTERNATE COURSES:** | **Alt Step 5:** The captured keyword or phrase is invalid.   * Return to step 2. | | | |
|  | **Alt Step 6:** No values on the system match the given search criteria. The system will display an error message.   * Return to step 2. | | | |
| **CONCLUSION:** | The system has successfully searched for the required challenge based off the given search criteria and displayed the results to the employee. | | | |
| **POST-CONDITION:** | The system successfully retrieves the challenge(s) on the system. | | | |
| **BUSINESS RULES** | Only an employee with the required access level functionality can maintain a challenge. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None | | | |
| **ASSUMPTIONS:** | None | | | |
| **OPEN ISSUES:** | None | | | |

Table 1.1.6

|  |  |  |
| --- | --- | --- |
| **JessterFIT** | | **Date: 2019/04/28** |
| **Authors(s): Joshua Eales**  **Janco Steenkamp (checked)** | | **Version 1.3** |
| **USE CASE NAME:** | **Add Personal Training Booking Slots** | **USE CASE TYPE** |
| **USE CASE ID:** | 1.7 | Business Requirements: ◻ |
| **PRIORITY:** | High | System Analysis: |
| **SOURCE:** | JessterFIT | System Design: |
| **PRIMARY BUSINESS ACTOR** | Employee | |
| **PRIMARY SYSTEM ACTOR** | None | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **OTHER INTERESTED STAKEHOLDERS:** | Owner | |
| **DESCRIPTION:** | The use case describes the event where an employee would like to add personal booking slots to the system. The system will return a calendar allowing the employee to select day. Once they have selected a day the employee will proceed to enter the personal trainer’s details at hourly intervals for the specified day. Personal training slots will then be saved by the system in the **Personal Training Slots table**. Once the system has successfully saved the personal training slots a success message will be displayed and employee will be returned to the calendar to allow further personal training slots to be allocated if needed. | |
| **PRE-CONDITION:** | Employee must be logged into the system and have the required access level functionality. | |
| **TRIGGER:** | An employee wishes to add new personal training booking slots. | |
| **TYPICAL COURSE** | Actor Action | System Response |
| **OF EVENTS:** | **Step 1**: The employee requests to add personal training booking slots to the system. |  |
|  |  | **Step 2**: System prompts the user to select a date. |
|  | **Step 3**: Employee selects the date on which they choose to add the new personal training booking slots. |  |
|  |  | **Step 4**: System captures the date selected. The system prompts the employee to enter personal trainer details. |
|  | **Step 5**: Employee enters personal trainer details. For the selected date. |  |
|  |  | **Step 6**: System captures entered personal booking slots details. |
|  |  | **Step 7**: System validates the entered information. |
|  |  | **Step 8**: System checks to ensure slots have not been over booked. |
|  |  | **Step 9**: System makes the user aware of available dates not allocated and prompts the user if he wishes to add more booking slots or complete the use case. |
|  | **Step 10**: Employee selects to complete the use case. |  |
|  |  | **Step 11**: System adds the new personal training booking date in the **Personal\_Training\_Booking table.** |
|  |  | **Step 12**: Success message is displayed to employee and the employee is returned to the calendar page allowing for further personal training booking slots to be added if needed. |
| **ALTERNATE COURSES:** | **Alt-step 7**: Personal trainer information entered was invalid or incomplete. Display error message, return to Step 5 making employee aware of information needing attention. | |
|  | **Alt-step 9**: Employee chooses to add more personal training slots. Return to step 2. | |
| **CONCLUSION:** | When the new personal training booking slots are successfully added to the system and a success message is displayed. | |
| **POST-CONDITION:** | Personal training booking slot details are completed fully and successfully added to the system. | |
| **BUSINESS RULES** | Employee can only assign personal training slots if they have the required access level. | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | Cannot book a slot that is already taken. | |
| **ASSUMPTIONS:** | Personal training booking details do not already exist on the system. | |

Table 1.1.7

|  |  |  |
| --- | --- | --- |
| **JessterFIT** | | **Date: 2019/04/28** |
| **Authors(s): Joshua Eales**  **Janco Steenkamp (checked)** | | **Version 1.3** |
| **USE CASE NAME:** | **Add challenge consultation booking slots** | **USE CASE TYPE** |
| **USE CASE ID:** | 1.8 | Business Requirements: X |
| **PRIORITY:** | High | System Analysis: |
| **SOURCE:** | JessterFIT | System Design: |
| **PRIMARY BUSINESS ACTOR** | Employee | |
| **PRIMARY SYSTEM ACTOR** | None | |
| **OTHER PARTICIPATING ACTORS:** | None | |
| **OTHER INTERESTED STAKEHOLDERS:** | Owner | |
| **DESCRIPTION:** | The use case describes the event where an employee would like to add challenge consultation booking slots to the system. The system will return a calendar allowing the employee to select a date. Once they have selected a date the employee will proceed to enter the personal trainer details. Challenge consultation details:  **Consultation\_Venue, Consultation\_Date,**  will then be added to the system in the **Consultation\_Booking table**. Once the system has successfully added the challenge consultation details a success message will be displayed and employee will be returned to the calendar to allow further challenge consultation slots to be allocated if needed. | |
| **PRE-CONDITION:** | Employee must be logged into the system and have the required access level functionality. | |
| **TRIGGER:** | An employee wishes to add new challenge consultation booking slots. | |
| **TYPICAL COURSE** | Actor Action | System Response |
| **OF EVENTS:** | **Step 1**: The employee requests to add challenge booking slots to the system. |  |
|  |  | **Step 2**: System prompts the user to select a date. |
|  | **Step 3**: Employee selects the date on which they choose to add the new consultation booking slots. |  |
|  |  | **Step 4**: System captures the date selected. The system prompts the employee to enter personal trainer details. |
|  | **Step 5**: Employee enters personal trainer details. For the selected date. |  |
|  |  | **Step 6**: System captures entered challenge consultation booking slots information. |
|  |  | **Step 7**: System validates the entered information. |
|  |  | **Step 8**: System checks to ensure slots have not been double booked for a specific trainer. |
|  |  | **Step 9**: System makes the user aware of available dates not allocated and prompts the user if he wishes to add more booking slots or complete the use case. |
|  | **Step 10**: Employee selects to complete the use case. |  |
|  |  | **Step 11**: System adds the challenge consultation booking details to the **Consultation\_Booking table.** |
|  |  | **Step 12**: Success message is displayed to employee and the employee is returned to the calendar page allowing for further challenge consultation booking slots to be added if needed. |
| **ALTERNATE COURSES:** | **Alt-step 7**: Personal trainer information entered was invalid or incomplete. Display error message, return to Step 5 making employee aware of information needing attention. | |
|  | **Alt-step 9**: Employee chooses to add more personal training slots. Return to step 2. | |
| **CONCLUSION:** | This use case concludes when the new challenge consultation booking details are successfully saved and added to the system and success message is received. | |
| **POST-CONDITION:** | Challenge consultation booking slots details are completed fully and successfully added to the system. | |
| **BUSINESS RULES** | Employee can only assign personal trainers to challenge consultation booking if they have the required access level functionality | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | Cannot book a slot that is already taken. | |
| **ASSUMPTIONS:** | Challenge consultation booking details do not already exist on the system. | |
| **OPEN ISSUES:** | None | |

Table 1.1.8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Iain Steyn**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Confirm Payment** | | | **Use Case Type:** |
| **USE CASE ID:** | 1.9 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Time | | | |
| **PRIMARY SYSTEM ACTOR** | Employee | | | |
| **OTHER PARTICIPATING ACTORS:** | Banking Institution | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | Owner | | | |
| **DESCRIPTION:** | The Use Case describes the event in which the system must wait for a payment confirmation from a banking institution. The system will notify an employee that payment has been confirmed from the banking institution. The employee will confirm the payment and its related details:  **Sale\_ID**  The system will then capture the payment information and conclude the payment. If everything is done correctly the system will display a success message to the employee and notify the client of successful payment. | | | |
| **PRE-CONDITION:** | The employee and customer/ client must be logged into the system and have the right access levels and functionality to be able to add a new exercise plan type to the system. | | | |
| **TRIGGER:** | The system needs to confirm a payment made by a customer. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The client made a payment to the business. | | **Step 2:** The system waits for a notification from the banking institution in the form of a **Proof of Payment**. | |
|  | **Step 3:** The employee confirms the **Proof of Payment** along with the **Sale\_ID** | | **Step 4:** The system will capture the confirmed payment and conclude the transaction. | |
|  |  | | **Step 5:** The system will display a confirmation message on the screen and notify the customer/ client via email. | |
| **ALTERNATE COURSES:** | **Alt Step 3:** The payment is invalid. The system will display an error message.   * Return the user to the relevant payment step. | | | |
|  | **Alt-Step 4:** The system is unable to confirm the payment. An error message is displayed on the screen advising the employee on further action | | | |
| **CONCLUSION:** | The use case concludes when payment is confirmed, and its details is added. | | | |
| **POST-CONDITION:** | The purchase details have been added to the | | | |
| **BUSINESS RULES** | Only employees with the required access levels can confirm payments and VAT is added at 15%. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | Client has correctly made the payment. | | | |
| **OPEN ISSUES:** | None. | | | |

Table 1.1.9

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Iain Steyn**  **Janco Steenkamp (checked)** | | **Version 1.1** | | |
| **USE CASE NAME:** | **Search Subscription** | | | **USE CASE TYPE** |
| **USE CASE ID:** | 1.10 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee needs to search for a subscription on the system. The user will search for a subscription by entering a keyword or phrase that matches any existing:  **Subscription\_ID, Subscription\_Type\_ID,**  in the **Subscription table**, into the provided (by the system) search bar.The system will then verify and search for any existing subscription and display subscription details that match the given search criteria. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to search a subscription on the system. | | | |
| **TRIGGER:** | The employee needs to search for a subscription on the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee requests to search for a subscription on the system. | | **Step 2:** The system prompts the user to enter a phrase or keyword | |
|  | **Step 3:** The employee will enter any keyword or phrase that will match any existing:  **Subscription\_ID, Subscription\_Type\_ID,**  in the **Subscription table** on the system. | |  | |
|  | **Step 4:** The client will request to search. | | **Step 5:** The system will capture the entered search keyword or phrase. | |
|  |  | | **Step 6:** The system will validate the entered search criteria and confirm that it is a valid search keyword or phrase. | |
|  |  | | **Step 7:** The system will check the:  **Subscription\_ID, Subscription\_Type\_ID,**  details for all existing subscriptions in the **Subscription table** to see if any match the captured search criteria. | |
|  |  | | **Step 8:** The system returns a list of the subscription(s) matching the given search criteria. | |
| **ALTERNATE COURSES:** | **Alt Step 6:** The captured keyword or phrase is invalid, an error message is displayed.   * Return to step 2. | | | |
|  | **Alt Step 7:** No values on the system match the given search criteria. The system will display an error message.   * Return to step 2. | | | |
| **CONCLUSION:** | The system has successfully retrieved the required subscription based off the given search criteria and displayed the results to the employee. | | | |
| **POST-CONDITION:** | The system successfully retrieves the subscription(s) details on the system. | | | |
| **BUSINESS RULES** | Only an employee with the required access levels can maintain a subscription. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The subscription exists and the employee has the required access level. | | | |
| **OPEN ISSUES:** | None. | | | |

Table 1.1.10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Iain Steyn**  **Janco Steenkamp (checked)** | | **Version 1.1** | | |
| **USE CASE NAME:** | **Search Personal Training Booking** | | | **USE CASE TYPE** |
| **USE CASE ID:** | 1.11 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee needs to search for a personal training booking on the system. The user will search for a personal training booking by entering a keyword or phrase that matches any existing:  **Client\_ID, Employee\_ID, Training\_Date\_Booked, Training\_Venue, Training\_Booking\_ID**  in the **Personal\_Training\_Booking table**, into the provided (by the system) search bar.The system will then verify and search for any existing personal training bookings and display details that match the given search criteria. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to search a personal training booking on the system. | | | |
| **TRIGGER:** | The employee needs to search for a personal training booking on the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee needs to search for a personal training booking on the system. | | **Step 2:** The system prompts the user to enter a phrase or keyword | |
|  | **Step 3:** The employee will enter any keyword or phrase that will match any existing:  **Client\_ID, Employee\_ID, Training\_Date\_Booked, Training\_Venue,**  in the **Personal\_Training\_Booking table** on the system. | |  | |
|  | **Step 4:** The client will request to search. | | **Step 5:** The system will capture the entered search keyword or phrase. | |
|  |  | | **Step 6:** The system will validate the entered search criteria to confirm that it is a valid search keyword or phrase. | |
|  |  | | **Step 7:** The system will check the:  **Client\_ID, Employee\_ID, Training\_Date\_Booked, Training\_Venue,**  details for all existing subscriptions in the **Personal\_Training\_Booking table** to see if any match the captured search criteria. | |
|  |  | | **Step 8:** The system returns a list of all personal training booking(s) matching the given search criteria. | |
| **ALTERNATE COURSES:** | **Alt Step 6:** The captured keyword or phrase is invalid; the system will display an error message.   * Return to step 2. | | | |
|  | **Alt Step 7:** No values on the system match the given search criteria. The system will display an error message.   * Return to step 2. | | | |
| **CONCLUSION:** | The system has successfully retrieved the required personal training booking based off the given search criteria and displayed the results to the employee. | | | |
| **POST-CONDITION:** | The system successfully retrieves the personal training booking(s) on the system. | | | |
| **BUSINESS RULES** | Only an employee with the required access levels can search a personal training booking. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The personal training booking exists and the employee has the required access levels. | | | |
| **OPEN ISSUES:** | None. | | | |

Table 1.1.11

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Iain Steyn**  **Janco Steenkamp (checked)** | | **Version 1.1** | | |
| **USE CASE NAME:** | **Search Challenge Consultation Booking** | | | **USE CASE TYPE** |
| **USE CASE ID:** | 1.12 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee needs to search for a challenge consultation booking on the system. The user will search for a challenge consultation booking by entering a keyword or phrase that matches any existing:  **Client\_ID, Employee\_ID, Training\_Date\_Booked, Consultation\_ID, Training\_Venue**  in the **Consultation\_Booking table**, into the provided (by the system) search bar.The system will then verify and search for any existing challenge consultation bookings and display details that match the given search criteria. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the required access levels and functionality to be able to search a challenge consultation booking on the system. | | | |
| **TRIGGER:** | The employee needs to search for a challenge consultation booking on the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee requests to search for a challenge consultation booking on the system. | | **Step 2:** The system prompts the user to enter a phrase or keyword | |
|  | **Step 3:** The employee will enter any keyword or phrase that will match any existing:  **Client\_ID, Employee\_ID, Training\_Date\_Booked, Training\_Venue, Consultation\_ID**  in the **Consultation\_Booking table** on the system. | |  | |
|  | **Step 4:** The client will request to search. | | **Step 5:** The system will capture the entered search keyword or phrase. | |
|  |  | | **Step 6:** The system will validate the entered search criteria so that it is a valid search keyword or phrase. | |
|  |  | | **Step 7:** The system will check the:  **Client\_ID, Employee\_ID, Training\_Date\_Booked, Training\_Venue, Training\_Booking\_ID**  details for all existing subscriptions in the **Consultation\_Booking table** to see if any match the captured search criteria. | |
|  |  | | **Step 8:** The system returns a list of all consultation booking (s) matching the given search criteria. | |
| **ALTERNATE COURSES:** | **Alt Step 6:** The captured keyword or phrase is invalid; an error message will be displayed.   * Return to step 2. | | | |
|  | **Alt Step 7:** No values on the system match the given search criteria. The system will display an error message.   * Return to step 2. | | | |
| **CONCLUSION:** | The system has successfully retrieved the required consultation booking based off the given search criteria and displayed the results to the employee. | | | |
| **POST-CONDITION:** | The system successfully retrieves the consultation booking(s) on the system. | | | |
| **BUSINESS RULES** | Only an employee with the required access levels can search for a challenge consultation booking | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The relevant challenge consultation booking exists on the system and the employee has the required access level. | | | |
| **OPEN ISSUES:** | None. | | | |

Table 1.1.12

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/04/12** | | |
| **Authors: Iain Steyn**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Maintain Company** | | | **Use Case Type:** |
| **USE CASE ID:** | 1.13 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis:◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None | | | |
| **DESCRIPTION:** | The use case describes the event that an employee would like to maintain the company details on the system. The employee will request to maintain the company and the system will display the maintain company page. The system will then prompt the user to edit **Company\_Name, Company\_Address, Company\_RegisterNo, Company\_Email, Company\_Contact\_No, Account\_Name, Account\_No, Account\_Type, Account\_Branch, Account\_Branch\_Code**. The employee will then select the details they would like to update by selecting the edit option. All changes will be saved on the system and a success message will be displayed. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to maintain the company record. | | | |
| **TRIGGER:** | The employee selects to the company record from the edit option. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee requests to maintain the company from the maintain company option. | | **Step 2:** The system will display the maintain company page and prompt the employee to select a detail to update:  **Company\_Name, Company\_Address, Company\_RegisterNo, Company\_Email, Company\_Contact\_No, Account\_Name, Account\_No, Account\_Type, Account\_Branch, Account\_Branch\_Code**. | |
|  | **Step 3:** The employee will select the edit detail option for the details they would like to edit. | | **Step 5:** The system will display editable fields for the details selected. | |
|  | **Step 6:** The employee will update the company and save it to the system. | | **Step 7:** The system will capture the edited company details: **Company\_Name, Company\_Address, Company\_RegisterNo, Company\_Email, Company\_Contact\_No, Account\_Name, Account\_No, Account\_Type, Account\_Branch, Account\_Branch\_Code**. | |
|  |  | | **Step 8:** The system will validate the company to ensure they are valid company details. | |
|  |  | | **Step 9:** The system will then update the company for the selected **Company\_ID** in the **Company** **table**. | |
|  |  | | **Step 10:** The system will display a success message to the employee.  Return the employee to the home page. | |
|  | **Alt Step 8:** The updated company details not a valid value.  Return to step 6. | | | |
| **CONCLUSION:** | The use case ends when the company table has been successfully updated on the system and a success message is displayed to the employee. | | | |
| **POST-CONDITION:** | The company table has been successfully updated on the system. | | | |
| **BUSINESS RULES** | None | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None | | | |
| **ASSUMPTIONS:** | The company table exists | | | |
| **OPEN ISSUES:** | None | | | |

Table 1.1.13

2. Product Subsystem

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/04/12** | | |
| **Authors: Doreen Maloka** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Add New Product Type** | | | **Use Case Type:** |
| **USE CASE ID:** | 2.1 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: 🞎 |
| **SOURCE:** | JessterFIT | | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee would like to add a new product type to the system. The employee will add a new product type on the add new product type page which will be displayed by the system when the employee wants to add a new product type. The employee will add all the required details for the new product type such as: **Product\_Type\_Description,**  which will be saved in the **Exercise\_Plan\_Type table.** If everything is done correctly the system will display a success message to the employee. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to add a new product type to the system. | | | |
| **TRIGGER:** | The employee wants to add a new product type to the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee wants to add a new product type to the system | | **Step 2:** The system will display the add new product type screen and inform the employee to enter a new product type detail:  **Product\_Type\_Description**. | |
|  | **Step 3:** The employee will enter a new **Product\_Type\_Description** on the add product type page and save it on the system. | | **Step 4:** The system will capture the entered detail:  **Product\_Type\_Description.** | |
|  |  | | **Step 5:** The system will validate the entered type so that it is a valid product type description. | |
|  |  | | **Step 6:**  The system will check for duplicate product types from the database. | |
|  |  | | **Step 7:** The system will automatically generate a unique **Product\_Type\_ID** and add it to the **Product\_Type table**. | |
|  |  | | **Step 8:** The system will add the type to the **Product\_Type table** and save it as an:  **Product\_Type\_Description**. | |
|  |  | | **Step 9:** The system will display a success message to the employee and return the employee to the home page. | |
| **ALTERNATE COURSES:** | **Alt Step 5:** The entered **Product\_Type\_Description** is invalid. The system will display an error message.  Return the user to step 3. | | | |
|  | **Alt Step 6:** The entered **Product\_Type\_Description** is a duplicate. The system will display an error message.  Return the user to step 3. | | | |
| **CONCLUSION:** | The use case ends when the new product type has been successfully added to the system and a success message is displayed for the employee. | | | |
| **POST-CONDITION:** | A new product type has been successfully added to the system. | | | |
| **BUSINESS RULES** | Only a manager or owner can edit a product type. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None | | | |
| **ASSUMPTIONS:** | None | | | |
| **OPEN ISSUES:** | None | | | |

Table 2.2.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/04/12** | | |
| **Authors: Doreen Maloka** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Maintain product type** | | | **Use Case Type:** |
| **USE CASE ID:** | 2.2 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: 🞎 |
| **SOURCE:** | JessterFIT | | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None | | | |
| **DESCRIPTION:** | The use case describes the process where an employee wants to update or delete a product type from the system. The system **invokes use case 2.3 search product type** and retrieves the information from the database. The employee then updates the relevant information and saves it. The new information is updated on the system and displays a success message. The employee could alternatively delete the information from the system. The information will then be removed from the database and a success message will be displayed | | | |
| **PRE-CONDITION:** | The Employee must be logged into the system and have the right access levels and functionality to be able to add a new product type. There are product types saved on the system. | | | |
| **TRIGGER:** | The employee wants to update or delete product type information from the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee requests to maintain a product type from the maintain product type option. | | **Step 2:** The system displays a page where the employee can add search criteria for a product type. | |
|  | **Step 3:** The employee enters criteria for product type they want to update | |  | |
|  |  | | **Step 4:** The system invokes **use case 2.3 search product type** and displays information retrieved | |
|  | **Step 5:** The employee enters the information that requires changing and save them on the system. | |  | |
|  |  | | **Step 6:** The system will validate the entered product type details so that they are valid details. | |
|  |  | | **Step 7:** The system will save all the details to the **Product\_Type table** such as: | |
|  |  | | **Step 8:** The system will display a success message to the employee. | |
|  |  | |  | |
| **ALTERNATE COURSES:** | **Alt Step 4:** The employee selects to remove the product type details. | | | |
|  | **Alt step 3:** The system invokes use case Add exercise | | | |
|  | **Alt Step 6:** The entered details are not valid   * Return the user to step 3. | | | |
| **CONCLUSION:** | The use case ends when product type details are updated on the system or when they are successfully removed from the system. | | | |
| **POST-CONDITION:** | Updated product type details are saved on the system. | | | |
| **BUSINESS RULES** | Only a employee can update product type details. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None | | | |
| **ASSUMPTIONS:** | None | | | |
| **OPEN ISSUES:** | None | | | |

Table 2.2.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/04/12** | | |
| **Authors: Doreen Maloka** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Search Product Type** | | | **Use Case Type:** |
| **USE CASE ID:** | 2.3 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: 🞎 |
| **SOURCE:** | JessterFIT | | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Manager | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee needs to search for an exercise plan on the system. The user will search for an exercise plan by entering a keyword or phrase that matches any existing:  **Product\_Type\_ID,**  **Product\_Type\_Description,**  in the **Product\_Type table.** The system will then verify and search for any existing exercise plan and display an exercise plan that matches the given search criteria. | | | |
| **PRE-CONDITION:** | The Employee must be logged into the system and have the right access levels and functionality to be able to add a new exercise plan. There are exercise plans saved on the system. | | | |
| **TRIGGER:** | The employee wants to search for an exercise plan information from the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee selects to search for an exercise plan from the system | | **Step 2:** The system will prompt the employee to enter any keywords that match any existing:  **Exercise\_Plan\_Type\_ID,**  **Exercise\_Plan\_Type\_Name.** | |
|  | **Step 3:** The employee will enter any keyword or phrase that will match any existing:  **Exercise\_Plan\_Type\_ID,**  **Exercise\_Plan\_Type\_Name,**  on the system. | |  | |
|  | **Step 4:** The employee will request to search | | **Step 5:** The system will capture the entered search keyword or phrase | |
|  |  | | **Step 6:** The system will validate the entered search criteria so that it is a valid search keyword or phrase. | |
|  |  | | **Step 7:** The captured keyword or phrase is valid, and the system will check the:  **Exercise\_Plan\_Type\_ID,**  **Exercise\_Plan\_Type\_Name,**  details for all existing exercise plans in the **Exercise\_Plan\_Type table** to see if any match the captured search criteria. | |
|  |  | | **Step 8:** The system returns a list of all exercise plan (s) matching the given search criteria. | |
| **ALTERNATE COURSES:** | **Alt Step 6:** The captured keyword or phrase is invalid.  Return to step 4. | | | |
|  | **Alt Step 8:** No values on the system match the given search criteria. The system will display an error message.   * Return to step 4. | | | |
| **CONCLUSION:** | The system has successfully searched for the required exercise plan based off of the given search criteria and displayed the results to the employee. | | | |
| **POST-CONDITION:** | The system successfully finds the exercise plan/s on the system. | | | |
| **BUSINESS RULES** | None | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None | | | |
| **ASSUMPTIONS:** | None | | | |
| **OPEN ISSUES:** | None | | | |

Table 2.2.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Joshua Eales**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Add New Product** | | | **Use Case Type:** |
| **USE CASE ID:** | 2.4 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis:◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee would like to add a new product to the system. The employee will add a new product on the add new product page which will be displayed by the system when the employee wants to add a new product. The employee will add all the required details for the new product such as: **Product\_Name**, **Product\_Description**, **Product\_Price,** **Product\_Quantity, Product\_Type, Product\_Color** which will be saved in the **Product** table**.**  The system will then capture and validate the entered details and check to see if the product - provided by the user – doesn’t already exist on the system. If everything is done correctly the system will display a success message to the employee. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the required access levels and functionality to be able to add a new product to the system. | | | |
| **TRIGGER:** | The employee requests to add a new product to the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee requests to add a new product to the system. | | **Step 2:** The system will display the add new product screen and prompt t the employee to enter the new product details: **Product\_Name**, **Product\_Description**, **Product\_Price,Product\_Quantity, Product\_Type, Product\_Color**. | |
|  | **Step 3:** The employee will enter a new **Product\_Name**, **Product\_Description**, **Product\_Price,** **Product\_Quantity, Product\_Type, Product\_Color** on the add product page and save it on the system. | | **Step 4:** The system will capture the entered details. | |
|  |  | | **Step 5:** The system will validate the entered product details to ensure that it is a valid products detail. | |
|  |  | | **Step 6:** The system will check to see if the product details added does not already exist in the system. | |
|  |  | | **Step 7:** The system will automatically generate a unique **Product\_ID** and add it to the **Product table**. | |
|  |  | | **Step 8:** The system will add the product details to the **Product table.** | |
|  |  | | **Step 9:** The system will display a success message to the employee and return the employee to the product page. | |
| **ALTERNATE COURSES:** | **Alt Step 5:** The entered product details are invalid. The system will display an error message.   * Return the user to step 3. | | | |
|  | **Alt Step 6:** The product details already exist on the system; the system will display an error message | | | |
| **CONCLUSION:** | The use case ends when the new product has been successfully added to the system and a success message is displayed to the employee. | | | |
| **POST-CONDITION:** | A new product has been successfully added to the system. | | | |
| **BUSINESS RULES** | Only an employee with the required access level functionality can add a product. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None | | | |
| **ASSUMPTIONS:** | The employee has the required access level and the product does not exist on the system before. | | | |
| **OPEN ISSUES:** | None | | | |

Table 2.2.4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Joshua Eales**  **Janco Steenkamp (checked)** | | **Version 1.1** | | |
| **USE CASE NAME:** | **Maintain Product** | | | **USE CASE TYPE** |
| **USE CASE ID:** | 2.5 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis:◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | Owner. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee would like to maintain a product on the system. The employee will request to maintain a product and the system will display the maintain product page. The system will then prompt the user to search for an existing product and invoke use case 2.6 **Search Product** and display the output on the page. The employee will then select the product they would like to maintain by selecting either the edit or delete option. All changes will be saved on the system and a success message will be displayed. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to maintain a product. | | | |
| **TRIGGER:** | The employee selects to edit a product from the edit product option. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: Employee requests to maintain a product on the system | | **Step 2:** The system will display the maintain product page and prompt the employee to search for an existing product on the page. | |
|  |  | | **Step 3:** The system will invoke use case 2.6 **Search Product** and display all the results to the user in a list on the maintain product page as well as an edit and remove option for each product in the list. | |
|  | **Step 4**: Employee will enter the **Product\_ID,** **Product\_Name, Product\_Description, Product\_Type\_ID, Product\_Color\_ID** to search by. | | **Step 5:** The system will retrieve all the details gathered from use case 2.6 **Search Product** for the selected product | |
|  | **Step 6:** The employee will select the edit product option for the product that they would like to update. | |  | |
|  | **Step 7:** The employee will update the product details and save it to the system. | | **Step 8:** The system will capture the edited product details. | |
|  |  | | **Step 9:** The system will validate the product to ensure they are valid product details. | |
|  |  | | **Step 10:** The system will then update the product for the selected **Product\_ID** in the **Product** **table**. | |
|  |  | | **Step 11:** The system will display a success message to the employee.   * Return the employee to the product page. | |
| **ALTERNATE COURSES:** | **Alt Step 4a:** The employee selects the remove product option for the product that they would like to remove. | | | |
|  | **Alt Step 4b:** The system prompts the user to confirm if they would like to delete the selected product. | | | |
|  | **Alt Step 4c:** The user confirms that they would like to delete the selected product. | | | |
|  | **Alt Step 4cc:** The User revokes the decision to remove the selected product and selects the option to cancel confirmation.   * Return to step 4. | | | |
|  | **Alt Step 4d:** The system removes the selected product and its relevant details from the **Product table.**   * Continue at step 10. | | | |
|  | **Alt Step 9:** The updated product details are not valid values.   * Return to step 7. | | | |
| **CONCLUSION:** | The selected product has been successfully maintained and all changes have been added on the system and a success message has been displayed. | | | |
| **POST-CONDITION:** | Product details have been successfully updated on the **Product table**. | | | |
| **BUSINESS RULES** | Only an employee with the required access level can maintain a product. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The employee has the required access levels and the product exists on the system | | | |
| **OPEN ISSUES:** | None. | | | |

Table 2.2.5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Joshua Eales**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Search Product** | | | **USE CASE TYPE** |
| **USE CASE ID:** | 2.6 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis:◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee needs to search for a product on the system. The user will search for a product by entering a keyword or phrase that matches any existing:  **Product\_ID,** **Product\_Name, Product\_Description, Product\_Type\_ID, Product\_Color\_ID** in the **Product table**, into the provided (by the system) search bar.The system will then verify and search for any existing product and display any product that matches the given search criteria. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the required access levels and functionality to be able to search a product on the system. | | | |
| **TRIGGER:** | The employee needs to search for a product on the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee requests to search for a product on the system. | | **Step 2:** Prompt user to enter search criteria | |
|  | **Step 3:** The employee will enter any keyword or phrase that will match any existing:  **Product\_ID,** **Product\_Name, Product\_Description, Product\_Type\_ID, Product\_Color\_ID** in the **Product table** on the system. | |  | |
|  | **Step 4:** The employee will proceed with the search. | | **Step 5:** The system will capture the entered search keyword or phrase. | |
|  |  | | **Step 6:** The system will validate the entered search criteria to confirm that it is a valid search keyword or phrase. | |
|  |  | | **Step 7:** The system will check the  details for all existing products in the **Product table** to see if any match the captured search criteria. | |
|  |  | | **Step 8:** The system returns a list of all product(s) matching the given search criteria. | |
| **ALTERNATE COURSES:** | **Alt Step 6:** The captured keyword or phrase is invalid.   * Return to step 2. | | | |
|  | **Alt Step 7:** No values on the system match the given search criteria. The system will display an error message.   * Return to step 2. | | | |
| **CONCLUSION:** | The system has successfully retrieved the product based off the given search criteria and displayed the results to the employee. | | | |
| **POST-CONDITION:** | The system successfully retrieves the product(s) on the system. | | | |
| **BUSINESS RULES** | Only an employee with the required access level can maintain a product. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None | | | |
| **ASSUMPTIONS:** | The product exists on the system and the employee has the required access level. | | | |
| **OPEN ISSUES:** | None | | | |

Table 2.2.6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Iain Steyn** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Add New Supplier** | | | **Use Case Type:** |
| **USE CASE ID:** | 2.7 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee would like to add a new supplier type to the system. The employee will request to add a new supplier on the add new supplier page which will be displayed by the system when the employee wants to add a new supplier. The employee will add all the required details for the new supplier such as:  **Supplier\_Description,**  **Supplier\_Email,**  **Supplier\_Phone,**  **Supplier\_Address,**  **Supplier\_Contact,**  which will be saved in the **Supplier table.** The system will then capture and validate the entered details and check to see if the supplier - provided by the user – doesn’t already exist on the system. If it does not exist, the system will save it as a new supplier and display a success message to the employee. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to add a new supplier to the system. | | | |
| **TRIGGER:** | The employee requests to add a new supplier to the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee wants to add a new supplier to the system. | | **Step 2:** The system will display the add new supplier page and inform the employee to enter all the required supplier details such as:  **Supplier\_Description,**  **Supplier\_Email,**  **Supplier\_Phone,**  **Supplier\_Address,**  **Supplier\_Contact.** | |
|  | **Step 3:** The employee will enter all the new details such as:  **Supplier\_Description,**  **Supplier\_Email,**  **Supplier\_Phone,**  **Supplier\_Address,**  **Supplier\_Contact,**  on the add supplier page and save them on the system. | | **Step 4:** The system will capture the entered details as:  **Supplier\_Description,**  **Supplier\_Email,**  **Supplier\_Phone,**  **Supplier\_Address,**  **Supplier\_Contact.** | |
|  |  | | **Step 5:** The system will validate the entered supplier details so that they are valid supplier details. | |
|  |  | | **Step 6:** The system will check to see if the supplier added does not already exist in the system. | |
|  |  | | **Step 7:** The system will automatically generate a unique **Supplier\_ID** and add it to the **Supplier table**. | |
|  |  | | **Step 8:** The system will add all the details to the **Supplier table** as:  **Supplier\_Description,**  **Supplier\_Email,**  **Supplier\_Phone,**  **Supplier\_Address,**  **Supplier\_Contact.** | |
|  |  | | **Step 9:** The system will display a success message to the employee and return the employee to the home page. | |
| **ALTERNATE COURSES:** | **Alt Step 5:** One or more of the entered supplier details are invalid so the system will display an error message.   * Return the user to step 3. | | | |
| **CONCLUSION:** | The use case ends when a new supplier has been successfully added to the system and a success message is displayed for the employee. | | | |
| **POST-CONDITION:** | A new supplier has been successfully added to the system. | | | |
| **BUSINESS RULES** | Only a manager or owner can add a supplier. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | None. | | | |
| **OPEN ISSUES:** | None. | | | |

Table 2.2.7

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Iain Steyn** | | **Version 1.1** | | |
| **USE CASE NAME:** | **Maintain Supplier** | | | **Use Case Type** |
| **USE CASE ID:** | 2.8 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee would like to maintain a supplier on the system. The employee will request to maintain a supplier and the system will display the maintain supplier page. The system will then prompt the user to search for an existing supplier on the maintain supplier page and then invoke use case **2.9 Search Supplier** and display the output on the page. The employee will then select the supplier they would like to maintain and edit or delete the type and all changes will be saved on the system and a success message will be displayed. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to maintain a supplier. | | | |
| **TRIGGER:** | The employee requests to maintain a supplier from the maintain supplier option. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee requests to maintain a supplier from the maintain supplier option. | | **Step 2:** The system will display the maintain supplier page and prompt the employee to search for an existing supplier on the page. | |
|  |  | | **Step 3:** The system will invoke use case **2.9 Search Supplier** and display all the results to the user in a list on the maintain supplier page as well as an edit and remove option for each supplier in the list. | |
|  | **Step 4:** The employee will select the edit supplier option for the supplier that they would like to edit. | | **Step 5:** The system will retrieve all the details gathered from use case 2.9 Search Supplier for the selected supplier such as:  **Supplier\_Description,**  **Supplier\_Email,**  **Supplier\_Phone,**  **Supplier\_Address,**  **Supplier\_Contact,**  and display them on the maintain supplier page into an editable field for each supplier detail. | |
|  | **Step 6:** The employee will update any supplier details that they need too and save all the changes to the system. | | **Step 7:** The system will capture the details as:  **Supplier\_Description,**  **Supplier\_Email,**  **Supplier\_Phone,**  **Supplier\_Address,**  **Supplier\_Contact.** | |
|  |  | | **Step 8:** The system will validate all supplier details so that they are still valid details. | |
|  |  | | **Step 9:** The system will then update all the subscription type details for the selected **Supplier\_ID** in the **Supplier Table**. | |
|  |  | | **Step 10:** The system will display a success message to the employee.   * Return the employee to the home page. | |
| **ALTERNATE COURSES:** | **Alt Step 4a:** The employee selects the remove supplier option for the supplier that they would like to remove. | | | |
|  | **Alt Step 4b:** The system prompts the user to confirm if they would like to delete the selected supplier. | | | |
|  | **Alt Step 4c:** The user confirms that they would like to delete the selected supplier and display success message. | | | |
|  | **Alt Step 4cc:** The User revokes the decision to remove the selected Supplier and selects the option to cancel confirmation.   * Return to Alt Step 4b. | | | |
|  | **Alt Step 4d:** The system removes the selected supplier and its relevant details from the **Supplier Table.**   * Continue at step 10. | | | |
|  | **Alt Step 8:** The updated Subscription type description or price is not a valid value.   * Return to step 6. | | | |
| **CONCLUSION:** | The selected subscription type has been successfully maintained and all changes have been saved on the system and a success message has been displayed. | | | |
| **POST-CONDITION:** | Supplier details have been successfully maintained. | | | |
| **BUSINESS RULES** | Only a manager or an owner can maintain a supplier. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | None. | | | |
| **OPEN ISSUES:** | None. | | | |

Table 2.2.8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Iain Steyn** | | **Version 1.1** | | |
| **USE CASE NAME:** | **Search Supplier** | | | **Use Case Type:** |
| **USE CASE ID:** | 2.9 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee needs to search for a supplier on the system. The user will search for a supplier by entering a keyword or phrase that matches any existing:  **Supplier\_ID,**  **Supplier\_Description,**  **Supplier\_Email,**  in the **Supplier Table**, into the provided (by the system) search bar.The system will then verify and search for any existing supplier and display a supplier that matches the given search criteria. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to search a supplier on the system. | | | |
| **TRIGGER:** | The employee needs to search for a supplier on the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee needs to search for a supplier on the system. | |  | |
|  |  | | Step 2: System prompts employee enter a phrase or keyword. | |
|  | **Step 3:** The employee will enter any keyword or phrase that will match any existing:  **Supplier\_ID,**  **Supplier\_Description,**  **Supplier\_Email,**  in the **Supplier Table** on the system. | |  | |
|  | **Step 3:** The employee will request to search. | | **Step 4:** The system will capture the entered search keyword or phrase. | |
|  |  | | **Step 5:** The system will validate the entered search criteria so that it is a valid search keyword or phrase. | |
|  |  | | **Step 6:** The captured keyword or phrase is valid, and the system will check the:  **Supplier\_ID,**  **Supplier\_Description,**  **Supplier\_Email,**  details for all existing suppliers in the **Supplier Table** to see if any match the captured search criteria. | |
|  |  | | **Step 7:** The system returns a list of all Supplier(s) matching the given search criteria. | |
| **ALTERNATE COURSES:** | **Alt Step 6:** The captured keyword or phrase is invalid. Display error message   * Return to step 2. | | | |
|  | **Alt Step 7:** No values on the system match the given search criteria. The system will display an error message.   * Return to step 2. | | | |
| **CONCLUSION:** | The system has successfully searched for the required supplier based off the given search criteria and displayed the results to the employee. | | | |
| **POST-CONDITION:** | The system successfully finds all Supplier(s) on the system. | | | |
| **BUSINESS RULES** | Only a manager or an owner can search for a supplier. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | None. | | | |
| **OPEN ISSUES:** | None. | | | |

Table 2.2.9

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Iain Steyn**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Add New Product Order** | | | **Use Case Type:** |
| **USE CASE ID:** | 2.10 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | Supplier | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | Owner | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee would like to add a new product order to the system. The employee will add a new product order on the add new product order page. The employee will add all the required details for the new product order such as:  **Supplier\_ID, Product\_ID, Ordered\_Quantity, Ordered\_Amount, Quantity\_Level, Date\_Ordered**  which will be saved in the **Supplier\_Product table.** The system will then capture and validate the entered details - provided by the user. If everything is done correctly the system will display a success message to the employee. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to add a new product order to the system. | | | |
| **TRIGGER:** | The product level of a certain product has dropped below a pre-defined value. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** |  | | **Step 1:** The system will notify an employee that a product’s quantity level has dropped below a pre-defined value and update the **Quantity\_Level** to **Low.** | |
|  | **Step 2:** The employee will request to create a new supplier order | | **Step 3:** The system will display the add new product order screen and prompt the user to enter the product order details**.** | |
|  | **Step 4:** The employee will enter the needed product order details:  **Supplier\_ID, Product\_ID, Ordered\_Quantity, Ordered\_Amount, Date\_Ordered** | | **Step 5:** The system will validate the entered details to confirm that it is a valid product order | |
|  |  | | **Step 6:** The system will automatically generate a unique **Order\_ID** and add it to the **Order table**. | |
|  |  | | **Step 7:** The system will calculate the total amount of the order and display order details on the screen. | |
|  | **Step 8:** The employee will confirm the product order details and submit the product order. | | **Step 9:** The system will notify the supplier and the owner of the order. The system will add the product order to the **Order table** and update the **Order­\_Status** to **Ordered**. | |
|  |  | | **Step 10:** The system will display a success message to the employee and return the employee to the product page. | |
| **ALTERNATE COURSES:** | **Alt Step 5:** The entered **Product Order details** are invalid. The system will display an error message.   * Return the user to step 4. | | | |
|  | **Alt Step 7:** The calculated amount is incorrect and could not be confirmed by the employee   * Return user to step 4 | | | |
|  | **Alt Step 9:** The system is unable to notify the supplier of the product order. The system will display an error message and advise the user to follow the provided action. | | | |
| **CONCLUSION:** | The use case concludes when the new product order has been successfully added to the system and the **Order\_Status** has been updated | | | |
| **POST-CONDITION:** | A new product order has been successfully added to the system. | | | |
| **BUSINESS RULES** | Only an employee with the required access level may add a new product order. The quantity level of a product must be below a pre-defined amount before an order is created. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The employee has the required access level to create a supplier order  The quantity level is below the specified amount  The supplier is trusted | | | |

Table 2.2.10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Janco Steenkamp**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Pay Supplier** | | | **Use Case Type:** |
| **USE CASE ID:** | 2.11 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | Supplier | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | Owner  Banking Institution | | | |
| **DESCRIPTION:** | The Use Case describes the event that the business would like to make a payment to a supplier. An employee will make a new payment to a supplier on the supplier payment page which will be displayed by the system when the employee wants to make a supplier payment. The employee will add all the required details for the new supplier order such as:  **Order\_ID**  which will be saved in the **Order table.** The system will then capture and validate the entered details - provided by the user. If everything is done correctly the system will display a success message to the employee. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to add a new supplier payment to the system. | | | |
| **TRIGGER:** | A product order is created, and the supplier is awaiting payment. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee receives confirmation of a successful product order. | |  | |
|  | **Step 2:** The employee will search the **Order\_ID** | | **Step 3:** The system invokes **Use Case 2.9 search supplier order**, calculates the amount to be paidand displays the details on the screen | |
|  | **Step 4:** The employee will confirm the order payment | | **Step 5:** The system will validate the order details | |
|  |  | | **Step 6:** The system will check the **Order\_Status** to confirm that the order has not yet been paid. | |
|  |  | | **Step 7:** The system will automatically update the **Order\_Status** in the **Order table**. | |
|  |  | | **Step 8:** The system will notify the supplier and the owner of the recent payment, display a success message to the employee and return the employee to the product order page. | |
|  |  | |  | |
| **ALTERNATE COURSES:** | **Alt Step 3:** The system could not retrieve the requested order details   * Return to step 2. | | | |
| **CONCLUSION:** | The use case ends when the new supplier payment has been added and the order status has been updated | | | |
| **POST-CONDITION:** | A new supplier payment has been successfully added to the system and the order status has been updated. | | | |
| **BUSINESS RULES** | Only an employee with the required access levels can add a new supplier payment. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The supplier has been successfully rated and is trusted | | | |
| **OPEN ISSUES:** | Suppliers should be rated and trusted | | | |

Table 2.2.11

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Janco Steenkamp**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Confirm supplier order** | | | **Use Case Type:** |
| **USE CASE ID:** | 2.12 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | Owner | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | Supplier | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee would receive a supplier order and confirm it on the system. The employee will update the received supplier order status on the order page. The employee will add all the required details for the received supplier order such as:  **Employee\_ID, Date\_Received, Quantity\_Received,**  which will be saved in the **Order table.** The system will then capture and validate the entered details and check to see if the supplier order received - provided by the user – doesn’t already exist on the system. If everything is done correctly the system will display a success message to the employee. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to confirm a supplier order received to the system. | | | |
| **TRIGGER:** | A product order has been received from a supplier. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee receives a supplier order | |  | |
|  | **Step 2:** The employee will search the **Order\_ID** | | **Step 3:** The system will display the details related to the **Order\_ID** | |
|  | **Step 4:** The employee will enter the supplier order details received to the system | | **Step 5:** The system will validate the entered details so that it corresponds with the supplier order created. | |
|  |  | | **Step 6:** The system will check to see if the supplier order received does not already exist in the system. | |
|  |  | | **Step 7:** The system will automatically update the **Order\_Status** to **Completed** in the **Order table**. | |
|  |  | | **Step 8:** The system will display a success message to the employee and return the employee to the order page. | |
|  |  | |  | |
| **ALTERNATE COURSES:** | **Alt Step 3:** The **Order\_ID** does not exist on the system.   * Notify the supplier, terminate the use case. | | | |
|  | **Alt Step 5:** The supplier order received does not match the supplier order placed   * Notify the supplier of the incident, terminate the use case | | | |
|  | **Alt Step 6:** The **Order\_ID** exists on the system with a completed status   * Notify the supplier, terminate the use case | | | |
| **CONCLUSION:** | The use case ends when the received order details have been added to the system and the **Order\_Status** has been updated to **Completed**; a success message is displayed to the employee. | | | |
| **POST-CONDITION:** | A supplier order has been received and the status has been updated. | | | |
| **BUSINESS RULES** | Only an employee with the required access levels may receive a supplier order. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | A valid order is received | | | |
| **OPEN ISSUES:** | None. | | | |

Table 2.2.12

3. User Subsystem

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors: Dalana Fhatuwani**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Login** | | | **Use Case Type** |
| **USE CASE ID:** | 3.1 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | User | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None | | | |
| **DESCRIPTION:** | This use case describes the process where the user logs into the system. The user enters login details such **Username** and **Password.**  This is where the user’s access level is checked to see what functions the user can access. The use case concludes once the user has logged onto the system and has access to the various functions of the system which are available to the user | | | |
| **PRE-CONDITION:** | The user must not be logged on to the system | | | |
| **TRIGGER:** | The user requests to login | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The user requests to access their account, by selecting to log in. | |  | |
|  |  | | **Step 2**: The system displays the login page and prompts the user for their:  **Username, Password** | |
|  | **Step 3**: The user provides their current username and password as requested by the system | | **Step 4**: The system captures the entered login details as specified | |
|  |  | | **Step 5:** The system validates the entered details by the user, comparing it to the usernames and passwords stored in the **Client or Employee table**. | |
|  |  | | **Step 6**: They system determines the access level of the user based on the user’s access level. | |
|  |  | | **Step 7**: The system logs the user into the system and navigates to the home page | |
| **ALTERNATE COURSES:** | **ALT Step 5**: The user login details do not exist in the system, display error.   * Invoke use case **5.1 Add New Client** | | | |
| **CONCLUSION:** | The use case concludes when user is logged onto the system and has access to their various functionalities of the system | | | |
| **POST-CONDITION:** | The user has access to the various functionalities based on their access level. | | | |
| **BUSINESS RULES** | There are two types of users:  **Employees** and **Clients,**  The access level of an employee is determined by the **Employee\_Type**. The access level of a client is determined by the contents the clients has signed up for on the system. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None | | | |
| **ASSUMPTIONS:** | Users exist on the system | | | |
| **OPEN ISSUES:** | None | | | |

Table 3.3.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/04/12** | | |
| **Authors: Dalana Fhatuwani**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Update Password** | | | **Use Case Type:** |
| **USE CASE ID:** | 3.2 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis:◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | User | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | Owner | | | |
| **DESCRIPTION:** | This use case describes the process where a user wants to update their password. The user proceeds to update the details on the system by selecting to update their password on the client details page. The system validates the current password. The user will be able to update their password. The use case concludes once the employee has updated their password. | | | |
| **PRE-CONDITION:** | The user must be registered on the system.  The user must be logged into the system. | | | |
| **TRIGGER:** | The user requests to update their password. | | | |
| **TYPICAL COURSE**  **OF EVENTS:** | Actor Action | | System Response | |
| **Step 1:** The user requests to update their password. | |  | |
|  | | **Step 2:** The system requests the user to enter their current password | |
| **Step 3**: The user provides current password as requested by the system | |  | |
|  | | **Step 4**: The system captures the entered login details as specified | |
|  | | **Step 5**: The system validates the entered password by comparing it to the password linked with the user’s ID:  **Client\_ID** or **Employee\_ID** | |
|  | | **Step 6:** The system requests the user to input a new password. | |
| **Step 7:** The user inputs the new password | | **Step 8**: The system captures the new password | |
|  | | **Step 9:** The system validates the new password and checks to see if it has not previously been used. | |
|  | | **Step 10:** The system encrypts the entered password and updates the relevant details:  **Password** in either the **Employee** or **Client table** | |
|  |  | | **Step 11:** The system displays a confirmation message that the user has successfully updated their password. | |
| **ALTERNATE COURSES:** | **Alt Step 1:** The user has forgotten their password   * Invoke use case **3.3 Forgotten Password** | | | |
|  | **ALT Step 4:** The user password details do not match the data in the **Employee** or **Client table**, display error.   * Return to Step 2 | | | |
|  | **ALT Step 7:** The new password matches old password, display error   * Return to Step 6 | | | |
| **CONCLUSION:** | The user’s password has been updated | | | |
| **POST-CONDITION:** | The **Password** has been updated in either the **Client** or **Employee table** andis up to date | | | |
| **BUSINESS RULES** | Passwords are hashed and stored in encrypted form | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None | | | |
| **ASSUMPTIONS:** | The user knows his current password | | | |
| **OPEN ISSUES:** | None | | | |

Table 3.3.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/04/12** | | |
| **Authors: Dalana Fhatuwani**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Forgotten Password** | | | **Use Case Type:** |
| **USE CASE ID:** | 3.3 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: **◻** |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | User | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | Owner | | | |
| **DESCRIPTION:** | This use case describes the process where a user has forgotten his/her password. The user will select that they have forgotten the password. The system will request the user’s email address, the user will provide their valid email address and the system will send a link to a password recovery page via email. | | | |
| **PRE-CONDITION:** | The user must be registered on the system. | | | |
| **TRIGGER:** | The user has forgotten their password login details. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1:** The user requests for new password since they have forgotten their password | |  | |
|  |  | | **Step 2:** The system displays forgot password screen | |
|  | **Step 3**: The user enters their email address | |  | |
|  |  | | **Step 4**: The system captures the entered details as specified. | |
|  |  | | **Step 5**: The system validates the format of the entered email address format; The system checks if the email address exists on the system in either the **Employee table** or **Client table** | |
|  |  | | **Step 6:** The system sends an email to the user with a link to update their password | |
|  |  | | **Step 7:** The system invokes use case **3.2 Update password step 6** | |
|  |  | | **Step 8:** The system displays a confirmation message that the user has successfully updated their password. | |
| **ALTERNATE COURSES:** | **ALT Step 5:** The email does not match any data in the **Employee** or **Client table**, display error.   * Return to Step 2 | | | |
| **CONCLUSION:** | The user has updated their password. | | | |
| **POST-CONDITION:** | The user has access to the system using their new password and the respective tables have been updated. | | | |
| **BUSINESS RULES** | Passwords are hashed and stored in encrypted form | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None | | | |
| **ASSUMPTIONS:** | The user knows its email address | | | |
| **OPEN ISSUES:** | None | | | |

Table 3.3.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/04/12** | | |
| **Authors: Dalana Fhatuwani**  **Janco Steenkamp (checked)** | | **Version 1.1** | | |
| **USE CASE NAME:** | **Logout** | | | **Use Case Type:** |
| **USE CASE ID:** | 3.4 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: **◻** |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | User | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | Owner | | | |
| **DESCRIPTION:** | This use case describes the process where the user is done using the system for the time being and he/she wants to log out of the system. The employee must select the option to logout. The use case concludes when the employee is logged out of the system. | | | |
| **PRE-CONDITION:** | The user must be logged in to the system | | | |
| **TRIGGER:** | The user requests to log out of the system | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The user requests to log out of the system by selecting the log out option | |  | |
|  |  | | **Step 2:** The system displays the log out screen | |
|  | **Step 3:**  The user selects confirm to log out | |  | |
|  |  | | **Step 4**: The system captures the date the user logged out at, and updates it in either the **Client** or **Employee table** | |
|  |  | | **Step 5**: The system redirects the user to the home page. | |
|  |  | |  | |
| **ALTERNATE COURSES:** | **ALT 3Alt Step 3**: The user selects to cancel log out   * Terminate use case, redirect user to home page | | | |
| **CONCLUSION:** | The user is logged off | | | |
| **POST-CONDITION:** | Information about the user logging off is saved onto the system | | | |
| **BUSINESS RULES** | There are two types of users:  **Employees** and **Clients,**  The access level of an employee is determined by the **Employee\_Type**. The access level of a client is determined by the contents the clients has signed up for on the system. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None | | | |
| **ASSUMPTIONS:** | The user is logged in to the system | | | |
| **OPEN ISSUES:** | None | | | |

Table 3.3.4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/04/12** | | |
| **Authors: Iain Steyn**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Add Access Level** | | | **Use Case Type:** |
| **USE CASE ID:** | 3.5 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: **◻** |
| **SOURCE:** | JessterFit | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | Owner | | | |
| **DESCRIPTION:** | This use case describes when an employee with the required access level wants to add a new user access level to the system. The employee has to login into the system and navigate to the admin page, then enter details regarding the new access level. The use case ends when the access level has been added to the system or if the process is cancelled | | | |
| **PRE-CONDITION:** | The access level to be added must be approved by the owner.  The user must have logged in with the required access level | | | |
| **TRIGGER:** | The employee selects to add a new access level to the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1:** An Employee wants to add a new Access level to the system. | |  | |
|  |  | | **Step 2**: The system prompts the employee to enter the information required | |
|  | **Step3:** The Employee enters the details:  **Access\_Description** as requested by the system | |  | |
|  |  | | **Step 4**:  The system captures the details for a new Access Level entered by the employee | |
|  |  | | **Step 5**: The system validates the format of the captured details to check if it is in the correct format. | |
|  |  | | **Step 6**: The system check if the access level details does not already exist on the system by checking the **Access\_Level table** for duplicate access level details | |
|  |  | | **Step 7:** The system will automatically generate a unique **Access\_Level** ID and add it to the **Access\_Level table** | |
|  |  | | **Step 8:** The system will add the **Access\_Description** to the **Access\_Level table.** | |
|  | . | | **Step 9:** The system will display a success message on the screen and navigate the employee to the admin screen | |
|  |  | |  | |
| **ALTERNATE COURSES:** | **Alt Step 5**: The information entered fails the validation test of the data format. The system displays an error message stating that the data format is not correct   * Return to step 3 | | | |
|  | **Alt Step 6**: The details entered already exists, display an error message   * Return to step 3 | | | |
| **CONCLUSION:** | The use case ends when a new Access Level has been successfully added to the **Access\_Level table** and a success message is displayed to the employee. | | | |
| **POST-CONDITION:** | A new Access Level has been successfully added to the system. | | | |
| **BUSINESS RULES** | None | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None | | | |
| **ASSUMPTIONS:** | The access level does not exist on the system, the employee has the required access level functionality. | | | |

Table 3.3.5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/04/12** | | |
| **Authors: Iain Steyn**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Maintain Access Level** | | | **Use Case Type:** |
| **USE CASE ID:** | 3.6 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis:◻ |
| **SOURCE:** | JessterFit | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | Owner | | | |
| **DESCRIPTION:** | This use case describes when an employee with the required access level wants to maintain an existing user access level to the system. The employee has to login into the system and navigate to the admin page, then enter a keyword or phrase to search for the required access level. The employee retrieved will have the option to edit or delete the access level. The use case ends when the access level has been successfully maintained on the system or if the process is cancelled | | | |
| **PRE-CONDITION:** | The access level exists on the system.  The employee must have logged in with the required user access level | | | |
| **TRIGGER:** | Employee requests to maintain an Access Level. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: An Employee requests to maintain an access level on the system. | |  | |
|  |  | | **Step 2**: The system Invokes **UC 3.7 Search Access Level** and display the relevant access levels. | |
|  | **Step3:** The employee selects the edit access level option. | |  | |
|  |  | | **Step 4**: The system displays the maintain access level screen. In which the details:  **Access\_Description**  of the access level is editable. | |
|  | **Step 5**: The employee edits the details which needed to be updated and select to update | | . | |
|  |  | | **Step 6**: The system captures the updated information entered by the employee | |
|  |  | | **Step 7**:  The system validates the format of the captured information to check if it is in the correct format. | |
|  |  | | **Step 8**: The system will update the needed access level details in the **Access\_Level table** | |
|  |  | | **Step 9**: The system displays a message stating the that table has been updated successfully and the employee is navigated to the admin page | |
| **ALTERNATE COURSES:** | **Alt Step 5a**: The employee selects to remove the access level | | | |
|  | **Alt Step 5b**: The system prompts to confirm the removal. | | | |
|  | **Alt Step 5bb**: The employee cancels the removal   * Return to admin page | | | |
|  | **Alt Step 5c:** The system removes the selected **Access\_Level** and **Access\_Description** relevant from the **Access\_Level table**. | | | |
|  | **Alt Step 5d**: The system displays a confirmation message and navigates the employee to the admin page | | | |
|  | **Alt Step 7**: The information entered fails the validation test of the data format. The system displays a message stating that the data format is not correct   * Return to step 3 | | | |
| **CONCLUSION:** | The use case ends when the access level has been updated in the **Access\_Level table** and a success message is displayed to the employee. | | | |
| **POST-CONDITION:** | An access level has been successfully updated in the **Access\_Level table** and the table is up to date. | | | |
| **BUSINESS RULES** | Only an employee with the required access level may maintain an access level. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None | | | |
| **ASSUMPTIONS:** | The relevant access level exists on the system | | | |
| **OPEN ISSUES:** | None | | | |

Table 3.3.6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/04/12** | | |
| **Authors: Iain Steyn**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Search Access Level** | | | **Use Case Type:** |
| **USE CASE ID:** | 3.7 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis:◻ |
| **SOURCE:** | JessterFit | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee needs to search for an access level on the system. The user will search for an access level by entering a keyword or phrase that matches any existing:  **Access\_Level, Access\_Description** in the **Access\_Level table**, into the provided (by the system) search bar.The system will then verify and search for any existing access level and display its relevant details that matches the given search criteria. | | | |
| **PRE-CONDITION:** | The access level must exist on the system  The employee must be logged in with the required access level | | | |
| **TRIGGER:** | Employee request to search for an access level | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: Employee requests to search for an access level. | |  | |
|  |  | | **Step 2**: The system displays the search criteria for the access level | |
|  | **Step3:** The Employee enters the details:  **Access\_Level, Access\_Description** and submits the search | |  | |
|  |  | | **Step 4**: The system captures the entered details provided by the employee | |
|  |  | | **Step 5**: The system retrieves the relevant details from the **Access\_Level table** and displays the results in a list on the screen | |
|  | **Step 6**: The employee selects the specific access level they are looking for. | |  | |
|  |  | | **Step 7**: The system displays the details of the selected access level | |
| **ALTERNATE COURSES:** | **Alt Step 5**: The system displays a message saying no results found matching the entered criteria   * Return to step 3 | | | |
| **CONCLUSION:** | The use case ends when an access level with its details have been successfully retrieved from the **Access\_Level table** and displayed on the screen | | | |
| **POST-CONDITION:** | An access level has been successfully retrieved by the system. | | | |
| **BUSINESS RULES** | Only an employee with the required access level may search an access level | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None | | | |
| **ASSUMPTIONS:** | The access level exists, and the search criteria is valid | | | |
| **OPEN ISSUES:** | None | | | |

Table 3.3.7

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/04/12** | | |
| **Authors: Iain Steyn**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Add functionality** | | | **Use Case Type:** |
| **USE CASE ID:** | 3.8 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: **◻** |
| **SOURCE:** | JessterFit | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | Owner | | | |
| **DESCRIPTION:** | This use case describes when an employee wants to add a new functionality to the system. The employee must be logged in to the system and have the required access level to be able select the option to add a new functionality. The use case ends when the functionality is added to the system or if the process is cancelled | | | |
| **PRE-CONDITION:** | The functionality must not exist on the system  The employee must have the required access level | | | |
| **TRIGGER:** | The employee selects to add a new functionality to the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1:** An employee requests to add a new functionality to the system. | |  | |
|  |  | | **Step 2**: The system requests the employee to enter the information required | |
|  | **Step 3:** The employee enters details:  **Function\_Description** as requested by the system | |  | |
|  |  | | **Step 4**:  The system captures the data for a new functionality entered by the employee | |
|  |  | | **Step 5**: The system validates the format of the captured information to check if it is in the correct forma. | |
|  |  | | **Step 6**: The system check if the functionality does not already exist on the system by checking the **Functionality** table for same **Functionality\_Description** | |
|  |  | | **Step 7:** The system will automatically generate a unique **Function\_ID** and add it to the **Functionality table.** | |
|  |  | | **Step 8:** The system will add the functionality details to the **Functionality table** | |
|  |  | | **Step 9:** The system displays a confirmation message to the employee and navigate to the functionality page | |
| **ALTERNATE COURSES:** | **Alt Step 5**: The information entered fails the validation test of the data format. The system displays a message stating that the data format is not correct   * Returns to step 3 | | | |
|  | **Alt Step 6**: The Functionality entered already exists   * Return to step 3 | | | |
| **CONCLUSION:** | The use case ends when a new functionality has been successfully added to the **Functionality table** and a success message is displayed to the employee. | | | |
| **POST-CONDITION:** | A new functionality has been successfully added to the system and the **Functionality table** has been updated. | | | |
| **BUSINESS RULES** | Only an employee with the required access level an add a functionality | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None | | | |
| **ASSUMPTIONS:** | The functionality does not exist on the system | | | |
| **OPEN ISSUES:** | None | | | |

Table 3.3.8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/04/12** | | |
| **Authors: Iain Steyn**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Maintain Functionality** | | | **Use Case Type:** |
| **USE CASE ID:** | 3.9 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis:◻ |
| **SOURCE:** | JessterFit | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | Owner | | | |
| **DESCRIPTION:** | This use case describes when an employee wants to maintain an existing functionality on the system. The employee must be logged in to the system and have the required access level to be able select the option to edit or remove a functionality. The use case ends when the functionality has been maintained on the system or if the process is cancelled | | | |
| **PRE-CONDITION:** | The functionality exists on the system.  The user must be logged in with the required access level | | | |
| **TRIGGER:** | Employee requests to maintain a functionality. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: An Employee requests to maintain functionality on the system. | |  | |
|  |  | | **Step 2**: The system Invokes **UC 3.7 Search Functionality** and displays the retrieved records in a list | |
|  | **Step3:** The employee selects the edit option for the relevant functionality | |  | |
|  |  | | **Step 4**: The system displays the maintain Functionality screen. In which the details of the Functionality are editable. | |
|  | **Step 5**: The employee edits the details:  **Functionality\_Description** selects to update | | . | |
|  |  | | **Step 6**: The system captures the updated information entered by the employees | |
|  |  | | **Step 7**:  The system validates the format of the updated captured information to check if it is in the correct format | |
|  |  | | **Step 8**: The system updates the **Functionality table** | |
|  |  | | **Step 9**: The system displays a message stating the that updating has been successful and navigates the user to the functionality page | |
| **ALTERNATE COURSES:** | **Alt Step 5a**: The Employee selects to remove the functionality | | | |
|  | **Alt Step 5b**: The system prompts to confirm the removal. | | | |
|  | **Alt Step 5bb**: The user cancels the removal   * Return to admin page | | | |
|  | **Alt Step 5c:** The system removes the selected **Functionality\_ID** and details relevant from the **Functionality table**. | | | |
|  | **Alt Step 5d**: The system displays a confirmation message and navigates the employee to the admin page | | | |
|  | **Alt Step 7**: The information entered fails the validation test of the data format. The system displays an error message stating that the data format is not correct   * Returns to step 3 | | | |
| **CONCLUSION:** | The use case ends when the record has been updated in the **Functionality table** on the system and a success message is displayed to the employee. | | | |
| **POST-CONDITION:** | A functionality record has been updated in the **Functionality table** on the system. | | | |
| **BUSINESS RULES** | Only an employee with the needed access level may maintain a functionality | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None | | | |
| **ASSUMPTIONS:** | The functionality record exists in the **Functionality table** | | | |

Table 3.3.9

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | **Date: 2019/04/12** | | |
| **Authors: Iain Steyn**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Search Functionality** | | | **Use Case Type:** |
| **USE CASE ID:** | 3.10 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis:◻ |
| **SOURCE:** | JessterFit | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee needs to search for a functionality on the system. The user will search for a functionality by entering a keyword or phrase that matches any existing:  **Function\_ID, Function\_Description** in the **Functionality table**, into the provided (by the system) search bar.The system will then verify and search for any existing functionality and display its relevant details that matches the given search criteria. | | | |
| **PRE-CONDITION:** | The function must exist on the system  The employee must be logged in to the system with the required access level | | | |
| **TRIGGER:** | Employee requests to search a functionality | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: Employee requests to search for a functionality. | |  | |
|  |  | | **Step 2**: The system displays the search criteria for the functionality | |
|  | **Step3:** The Employee enters the details:  **Function\_ID, Function\_Description** and submits the search | |  | |
|  |  | | **Step 4**: The system captures the entered details provided by the employee | |
|  |  | | **Step 5**: The system retrieves the relevant details from the **Functionality table** and displays the results in a list on the screen | |
|  | **Step 6**: The employee selects the specific functionality they are looking for. | |  | |
|  |  | | **Step 7**: The system displays the details of the selected functionality | |
|  |  | |  | |
|  |  | |  | |
| **ALTERNATE COURSES:** | **Alt Step 5**: The system displays an error message saying no results found matching the entered criteria   * Return to step 3 | | | |
| **CONCLUSION:** | The use case ends when an access level with its details have been successfully retrieved from the **Access\_Level table** and displayed on the screen | | | |
| **POST-CONDITION:** | A functionality has been successfully retrieved by the system. | | | |
| **BUSINESS RULES** | Only an employee with the required access level may search a functionality | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None | | | |
| **ASSUMPTIONS:** | The functionality exists, and the search criteria is valid | | | |
| **OPEN ISSUES:** | None | | | |

Table 3.3.10

4. Employee Subsystem

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Joshua Eales**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Add New Employee** | | | **Use Case Type:** |
| **USE CASE ID:** | 4.1 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis:◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Owner | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | Owner | | | |
| **DESCRIPTION:** | The Use Case describes the event that an owner would like to add a new employee to the system. The owner will add a new employee on the add new employee page which will be displayed by the system when the employee wants to add a new employee. The owner will add all the required details for the new employee such as: **Employee\_Name**, **Employee\_Surname**, **Employee\_Email**, **Employee\_Address**, **Employee\_Contact, Employee\_Password, Access\_Level, Employee\_ID\_Number** which will be saved in the **Employee table.**  The system will then capture and validate the entered details and check to see if the employee - provided by the user – doesn’t already exist on the system. If everything is done correctly the system will display a success message to the employee. | | | |
| **PRE-CONDITION:** | The owner must be logged into the system and have the right access levels and functionality to be able to add an employee to the system. | | | |
| **TRIGGER:** | The owner wants to add a new employee to the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The owner requests to add a new employee to the system. | | **Step 2:** The system will display the add new employee screen and inform the owner to enter the new employee’s detail’s: **Employee\_Name**, **Employee\_Surname**, **Employee\_Email**, **Employee\_Address**, **Employee\_Contact, Employee\_Password, Access\_Level, Employee\_ID\_Number** | |
|  | **Step 3:** The owner will enter a new **Employee\_Name**, **Employee\_Surname**, **Employee\_Email**, **Employee\_Address**, **Employee\_Contact, Employee\_Password, Access\_Level, Employee\_ID\_Number** on the add employee page and save it on the system. | | **Step 4:** The system will capture the entered detail:  **Employee\_Name**, **Employee\_Surname**, **Employee\_Email**, **Employee\_Address**, **Employee\_Contact, Access\_Level, Employee\_ID\_Number,**  Encrypt the entered **Employee\_Password** | |
|  |  | | **Step 5:** The system will validate the entered employee details to ensure that they are valid employee detail. | |
|  |  | | **Step 6:** The system will check to see if the employee added does not already exist in the system. | |
|  |  | | **Step 7:** The system will automatically generate a unique **Employee\_ID** and add it to the **Employee table**. | |
|  |  | | **Step 8:** The system will add the employee to the **Employee table** and save it as a new employee. | |
|  |  | | **Step 9:** The system will display a success message to the owner and return the employee to the home page. | |
| **ALTERNATE COURSES:** | **Alt Step 5:** The entered **Employee\_Name**, **Employee\_Surname**, **Employee\_Email**, **Employee\_Address**, **Employee\_Contact, Employee\_Password, Access\_Level, Employee\_ID\_Number** are invalid. The system will display an error message.   * Return the user to step 3. | | | |
|  | **Alt Step 6:** The entered **Employee\_Email**, **Employee\_Contact, Employee\_ID\_Number** already exist on the system, display an error message.   * Return the user to step 3. | | | |
| **CONCLUSION:** | The use case ends when the new employee has been successfully added to the **Employee table** and a success message is displayed to the employee. | | | |
| **POST-CONDITION:** | A new employee has been successfully added to the system. | | | |
| **BUSINESS RULES** | Only users with the required access level may add a new employee | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The employee does not exist in the system | | | |
| **OPEN ISSUES:** | None. | | | |

Table 4.4.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Joshua Eales**  **Janco Steenkamp (checked)** | | **Version 1.1** | | |
| **USE CASE NAME:** | **Maintain Employee** | | | **USE CASE TYPE** |
| **USE CASE ID:** | 4.2 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis:◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Owner | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that an owner would like to maintain an owner on the system. The owner will request to maintain an employee and the system will display the maintain employee page. The system will then prompt the user to search for an existing employee on the employee page and invoke use case **4.2 Search Employee** and display the output on the page. The owner will then select the employee they would like to maintain by selecting either the edit or delete option. All changes will be updated on the system and a success message will be displayed. | | | |
| **PRE-CONDITION:** | The owner must be logged into the system and have the right access levels and functionality to be able to maintain an employee. | | | |
| **TRIGGER:** | The owner selects to edit an employee from the edit option. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The owner requests to maintain an employee’s details. | | **Step 2:** The system will display the maintain employee page and prompt the owner to search for an existing employee on the page. | |
|  |  | | **Step 3:** The system will invoke use case **4.3 Search Employee** and display the results in a list | |
|  | **Step 4:** The owner will select the edit employee option for the employee that they would like to edit. | | **Step 5:** The system will retrieve all the details gathered from use case **4.3 Search Employee** for the selected employee such as: **Employee\_Name**, **Employee\_Surname**, **Employee\_Email**, **Employee\_Address**, **Employee\_Contact, Employee\_Password, Access\_Level, Employee\_ID\_Number** and display it on the maintain employee page into an editable field for the employee. | |
|  | **Step 6:** The owner will update the employee and submit it to the system. | | **Step 7:** The system will capture the edited employee as: **Employee\_Name**, **Employee\_Surname**, **Employee\_Email**, **Employee\_Address**, **Employee\_Contact, Employee\_Password, Access\_Level, Employee\_ID\_Number.** | |
|  |  | | **Step 8:** The system will validate the employee to ensure that the details are valid employee details. | |
|  |  | | **Step 9:** The system will then update the employee for the selected **Employee\_ID** in the **Employee table**. | |
|  |  | | **Step 10:** The system will display a success message to the owner.   * Return the owner to the home page. | |
| **ALTERNATE COURSES:** | **Alt Step 4a:** The owner selects the remove employee option for the employee that they would like to remove. | | | |
|  | **Alt Step 4b:** The system prompts the user to confirm if they would like to delete the selected employee. | | | |
|  | **Alt Step 4c:** The user confirms that they would like to delete the selected employee. | | | |
|  | **Alt Step 4cc:** The User revokes the decision to remove the selected employee and selects the option to cancel confirmation.   * Return to Step 1. | | | |
|  | **Alt Step 4d:** The system removes the selected employee and its relevant details from the **Employee table.**   * Continue at step 10. | | | |
|  | **Alt Step 8:** The updated employee details are invalid, display an error message.   * Return to step 6. | | | |
| **CONCLUSION:** | The selected employee has been successfully maintained and all changes have been updated in the **Employee table** on the system and a success message has been displayed. | | | |
| **POST-CONDITION:** | Employee details have been successfully updated on the system. | | | |
| **BUSINESS RULES** | Only an employee with the required access level may maintain an employee. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | None. | | | |
| **OPEN ISSUES:** | None. | | | |

Table 4.4.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Joshua Eales**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Search Employee** | | | **USE CASE TYPE** |
| **USE CASE ID:** | 4.3 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis:◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Owner | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that an owner needs to search for an employee on the system. The user will search for aa employee by entering a keyword or phrase that matches any existing:  **Employee\_Email**, **Employee\_ID, Email\_ID\_Number** in the **Employee table**, into the provided (by the system) search bar.The system will then verify and search for any existing employee and display any product that matches the given search criteria. | | | |
| **PRE-CONDITION:** | The user must be logged into the system and have the right access levels and functionality to be able to search an employee on the system. | | | |
| **TRIGGER:** | The user needs to search for a product on the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The owner needs to search for an employee on the system. | | **Step 2**: The system prompts the owner to enter search criteria | |
|  | **Step 3:** The owner will enter any keyword or phrase that will match any existing:  **Employee\_Email**, **Employee\_ID, Email\_ID\_Number** in the **Employee table** on the system. | |  | |
|  | **Step 4:** The owner will request to search. | | **Step 5:** The system will capture the entered search keyword or phrase. | |
|  |  | | **Step 6:** The system will validate the entered search criteria so that it is a valid search keyword or phrase. | |
|  |  | | **Step 7:** The system will check the entered:  **Employee\_Email**, **Employee\_ID, Email\_ID\_Number** details for all existing employees in the **Employee table** to see if any match the captured search criteria. | |
|  |  | | **Step 8:** The system returns a list of all employee(s) matching the given search criteria. | |
| **ALTERNATE COURSES:** | **Alt Step 6:** The captured keyword or phrase is invalid, display an error message.   * Return to step 2. | | | |
|  | **Alt Step 7:** No values on the system match the given search criteria, display an error message.   * Return to step 2. | | | |
| **CONCLUSION:** | The system has successfully retrieved the required employee based off the given search criteria and displayed the results to the owner. | | | |
| **POST-CONDITION:** | The system successfully retrieves the employee(s) on the system. | | | |
| **BUSINESS RULES** | Only a user with the required access level can search an employee. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The employee searched exists on the system | | | |
| **OPEN ISSUES:** | None. | | | |

Table 4.4.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Iain Steyn**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Add New Employee Type** | | | **Use Case Type:** |
| **USE CASE ID:** | 4.4 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee would like to add a new employee type to the system. The employee will add a new employee type on the add new employee type page which will be displayed by the system when the employee wants to add a new employee type. The employee will add all the required details for the new employee type such as:  **Employee\_Type,**  which will be saved in the **Employee\_Type table.** The system will then capture and validate the entered details and check to see if the employee type - provided by the user – doesn’t already exist on the system. If it does not exist, the system will save it as a new employee type and display a success message to the employee. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to add a new employee type to the system. | | | |
| **TRIGGER:** | The employee requests to add a new employee type to the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee requests to add a new employee type to the system. | | **Step 2:** The system will display the add new employee type screen and inform the employee to enter a new employee type detail:  **Employee\_Type.** | |
|  | **Step 3:** The employee will enter a new **Employee\_Type** on the add employee type page and save it on the system. | | **Step 4:** The system will capture the entered detail:  **Employee\_Type.** | |
|  |  | | **Step 5:** The system will validate the entered type so that it is a valid employee type description. | |
|  |  | | **Step 6:** The system will check to see if the type added does not already exist in the system. | |
|  |  | | **Step 7:** The system will automatically generate a unique **Employee\_Type\_ID** and add it to the **Employee\_Type table**. | |
|  |  | | **Step 8:** The system will add the type to the **Employee\_Type table** and save it as an:  **Employee\_Type**. | |
|  |  | | **Step 9:** The system will display a success message to the employee and return the employee to the home page. | |
| **ALTERNATE COURSES:** | **Alt Step 5**:The entered **Employee\_Type** is invalid. The system will display an error message.   * Return the user to step 3. | | | |
|  | **Alt Step 6**: The entered **Employee\_Type** exists on the system, display an error message   * Return user to step 3 | | | |
| **CONCLUSION:** | The use case ends when the new employee type has been successfully added to the **Employee table** on the system and a success message is displayed to the employee. | | | |
| **POST-CONDITION:** | A new employee type has been successfully added to the system. | | | |
| **BUSINESS RULES** | Only a user with the required access level can add a new employee type. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The employee type does not yet exist on the system | | | |
| **OPEN ISSUES:** | None. | | | |

Table 4.4.4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **JessterFIT** | | | **Date: 2019/03/30** | | |
| **Authors(s): Iain Steyn**  **Janco Steenkamp (checked)** | | | **Version 1.3** | | |
|  | **USE CASE NAME:** | **Maintain Employee Type** | | | **USE CASE TYPE** |
|  | **USE CASE ID:** | 4.5 | | | Business Requirements: x |
|  | **PRIORITY:** | High | | | System Analysis: ◻ |
|  | **SOURCE:** | JessterFIT | | | System Design: ◻ |
|  | **PRIMARY BUSINESS ACTOR** | Employee | | | |
|  | **PRIMARY SYSTEM ACTOR** | None. | | | |
|  | **OTHER PARTICIPATING ACTORS:** | None. | | | |
|  | **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
|  | **DESCRIPTION:** | The Use Case describes the event that an employee would like to maintain an employee type on the system. The employee will request to maintain an employee type and the system will display the maintain employee type page. The system will then prompt the user to search for an existing employee type on the employee type page and invoke use case **4.6 Search Employee Type** and display the output on the page. The employee will then select the type they would like to maintain by selecting either the edit or delete option. All changes will be saved on the system and a success message will be displayed. | | | |
|  | **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to maintain an employee type. | | | |
|  | **TRIGGER:** | The employee selects to edit an employee type from the edit employee option. | | | |
|  | **TYPICAL COURSE** | Actor Action | | System Response | |
|  | **OF EVENTS:** | **Step 1**: The employee requests to maintain an employee type from the maintain employee type option. | | **Step 2:** The system will display the maintain employee type page and prompt the employee to search for an existing employee type on the page. | |
|  |  |  | | **Step 3:** The system will invoke use case **4.6 Search Employee Type** and display all the results to the user in a list on the maintain employee type page as well as an edit and remove option for each type in the list. | |
|  |  | **Step 4:** The employee will select the edit employee type option for the employee type that they would like to edit. | | **Step 5:** The system will retrieve all the details gathered from use case **4.6 Search Employee Type** for the selected type such as:  **Employee\_Type,**  and display it on the maintain employee type page into an editable field for the type. | |
|  |  | **Step 6:** The employee will update the employee type and save it to the system. | | **Step 7:** The system will capture the edited type as: **Employee\_Type.** | |
|  |  |  | | **Step 8:** The system will validate the type so that it is a is a valid description. | |
|  |  |  | | **Step 9:** The system will then update the type for the selected **Employee\_Type\_ID** in the **Employee\_Type** **table**. | |
|  |  |  | | **Step 10:** The system will display a success message to the employee.   * Return the employee to the home page. | |
|  | **ALTERNATE COURSES:** | **Alt Step 4a:** The employee selects the remove employee type option for the employee type that they would like to remove. | | | |
|  |  | **Alt Step 4b:** The system prompts the user to confirm if they would like to delete the selected employee type. | | | |
|  |  | **Alt Step 4c:** The user confirms that they would like to delete the selected employee type. | | | |
|  |  | **Alt Alt Step 4c:** The User revokes the decision to remove the selected employee type and selects the option to cancel confirmation.   * Return to Alt Step 4b. | | | |
|  |  | **Alt Step 4d:** The system removes the selected employee type and its relevant details from the **Employee\_Type table.**   * Continue at step 10. | | | |
|  |  | **Alt Step 8:** The updated employee type description is not a valid value.   * Return to step 6. | | | |
|  | **CONCLUSION:** | The selected employee type has been successfully maintained and all changes have been updated in the **Employee\_Type table** on the system and a success message has been displayed. | | | |
|  | **POST-CONDITION:** | Employee type details have been successfully updated on the system. | | | |
|  | **BUSINESS RULES** | Only a user with the required access level can maintain an employee type. | | | |
|  | **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
|  | **ASSUMPTIONS:** | The employee type exists on the system | | | |
|  | **OPEN ISSUES:** | None. | | | |

Table 4.4.5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Iain Steyn** | | **Version 1.1** | | |
| **USE CASE NAME:** | **Search Employee Type** | | | **USE CASE TYPE** |
| **USE CASE ID:** | 4.6 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee needs to search for an employee type on the system. The user will search for an employee type by entering a keyword or phrase that matches any existing:  **Employee\_Type\_ID,**  **Employee\_Type,**  in the **Employee\_Type table**, into the provided (by the system) search bar.The system will then verify and search for any existing employee type and display an employee type that matches the given search criteria. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to search an employee type on the system. | | | |
| **TRIGGER:** | The employee requests to search for an employee type on the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee requests to search for an employee type on the system. | | **Step 2:** The system prompts the user to enter the search criteria | |
|  | **Step 3:** The employee will enter any keyword or phrase that will match any existing:  **Employee\_Type\_ID,**  **Employee\_Type,**  in the **Employee\_Type table** on the system. | |  | |
|  | **Step 4:** The employee will request to search. | | **Step 5:** The system will capture the entered search keyword or phrase. | |
|  |  | | **Step 6:** The system will validate the entered search criteria so that it is a valid search keyword or phrase. | |
|  |  | | **Step 7:** The captured keyword or phrase is valid, and the system will check the:  **Employee\_Type\_ID,**  **Employee\_Type,**  details for all existing employee types in the **Employee\_Type table** to see if any match the captured search criteria. | |
|  |  | | **Step 8:** The system returns a list of all employee type (s) matching the given search criteria. | |
| **ALTERNATE COURSES:** | **Alt Step 7:** The captured keyword or phrase is invalid, display an error message.   * Return to step 2. | | | |
|  | **Alt Step 8:** No values on the system match the given search criteria. The system will display an error message.   * Return to step 2. | | | |
| **CONCLUSION:** | The system has successfully retrieved the required employee type based off the given search criteria and displayed the results to the employee. | | | |
| **POST-CONDITION:** | The system successfully retrieved the employee type/s from the **Employee\_Type table** on the system. | | | |
| **BUSINESS RULES** | Only a user with the required access level can maintain an employee type. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The employee exists on the system | | | |
| **OPEN ISSUES:** | None. | | | |

Table 4.4.6

5. Client Subsystem

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Janco Steenkamp** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Add New Client** | | | **Use Case Type:** |
| **USE CASE ID:** | 5.1 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Client | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that a client would like to be registered on the system. The client will register on the new client page which will be displayed by the system. The client will add all the required details to be added to the system such as:  **Client\_Name, Client\_Surname, Client\_Email, Client\_Contact\_Number, Client\_Address, Client\_Gender, Client\_Password, Client\_Instagram**  which will be saved in the **Client table.** The system will then capture and validate the entered - provided by the user. If everything is done correctly the system will display a success message to the client. | | | |
| **PRE-CONDITION:** | The client must not exist on the system to be added to the system. | | | |
| **TRIGGER:** | The client wants to be added to the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The client wants to be added to the system. | | **Step 2:** The system will display the add client screen and prompt the client to enter his/her details:  **Client\_Name, Client\_Surname, Client\_Email, Client\_Contact\_Number, Client\_Address, Client\_Gender, Client\_Password, Client\_Instagram**  **.** | |
|  | **Step 3:** The client will enter the details: **Client\_Name, Client\_Surname, Client\_Email, Client\_Contact\_Number, Client\_Address, Client\_Gender, Client\_Password, Client\_Instagram**  on the add client page and submit it to the system. | | **Step 4:** The system will capture the entered detail:  **Client\_Name, Client\_Surname, Client\_Email, Client\_Contact\_Number, Client\_Address, Client\_Gender, Client\_Instagram**  and **encrypt** the **Client\_Password** to be added | |
|  |  | | **Step 5:** The system will validate the client to confirm that it is valid client details. | |
|  |  | | **Step 6:** The system will automatically generate a unique **Client\_ID** and add it to the **Client table**. | |
|  |  | | **Step 7:** The system will add the client to the **Client table** and save it as a:  **Client**. | |
|  |  | | **Step 8:** The system will display a success message to the employee and return the employee to the home page. | |
|  |  | |  | |
| **ALTERNATE COURSES:** | **Alt Step 5:** The entered **Client details** are invalid. The system will display an error message.   * Return the user to step 3. | | | |
|  | **Alt Step 6**: The entered **Client details** exists on the system, display an error message   * Return the user to step 3 | | | |
| **CONCLUSION:** | The use case ends when the new client has been successfully added to the system and a success message is displayed for the client. | | | |
| **POST-CONDITION:** | A new client has been successfully added to the **Client table** on the system. | | | |
| **BUSINESS RULES** | Only a client which does not exist on the system will be allowed to be added to the system. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The client does not yet exist on the system | | | |
| **OPEN ISSUES:** | None. | | | |

Table 5.5.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **JessterFIT** | | | **Date: 2019/03/30** | | |
| **Authors(s): Janco Steenkamp**  **Janco Steenkamp (checked)** | | | **Version 1.1** | | |
|  | **USE CASE NAME:** | **Maintain Client** | | | **USE CASE TYPE** |
|  | **USE CASE ID:** | 5.2 | | | Business Requirements: x |
|  | **PRIORITY:** | High | | | System Analysis: ◻ |
|  | **SOURCE:** | JessterFIT | | | System Design: ◻ |
|  | **PRIMARY BUSINESS ACTOR** | Client | | | |
|  | **PRIMARY SYSTEM ACTOR** | None. | | | |
|  | **OTHER PARTICIPATING ACTORS:** | None. | | | |
|  | **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
|  | **DESCRIPTION:** | The Use Case describes the event that a client would like to maintain his/her details on the system. The client will request to maintain his/her details and the system will display the maintain client page. The client will then select the details he/she would like to maintain by selecting either the edit or delete option. All changes will be saved on the system and a success message will be displayed. | | | |
|  | **PRE-CONDITION:** | The client must be logged into the system to be able to maintain his/her client details. | | | |
|  | **TRIGGER:** | The client requests to edit his/her details from the edit client option. | | | |
|  | **TYPICAL COURSE** | Actor Action | | System Response | |
|  | **OF EVENTS:** | **Step 1**: The client requests to maintain his/her details from the maintain client option. | | **Step 2:** The system will display the maintain client page with the client detail and prompt the client to update the needed details on the page. | |
|  |  | **Step 3:** The client will update his/her details:  **Client\_Name, Client\_Surname, Client\_Email, Client\_Contact\_Number, Client\_Address, Client\_Gender, Client\_Password, Client\_Instagram**  and submit it to the system | | **Step 4:** The system will capture the edited client details. | |
|  |  |  | | **Step 5:** The system will validate the client details to confirm that it is a is valid. | |
|  |  |  | | **Step 6:** The system will then update the client details for the selected **Client\_ID** in the **Client** **table**. | |
|  |  |  | | **Step 7:** The system will display a success message to the employee.  Return the employee to the home page. | |
|  |  |  | |  | |
|  |  |  | |  | |
|  |  | **Alt Step 5:** The updated client details are not valid values, display an error message.   * Return to step 3. | | | |
|  | **CONCLUSION:** | The selected client details have been successfully updated and all changes have been added to the **Client table** on the system and a success message has been displayed. | | | |
|  | **POST-CONDITION:** | Client details have been successfully updated on the system. | | | |
|  | **BUSINESS RULES** | Only client existing on the system can maintain his/her own client details. | | | |
|  | **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
|  | **ASSUMPTIONS:** | The client enters valid details | | | |
|  | **OPEN ISSUES:** | None. | | | |

Table 5.5.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Iain Steyn**  **Janco Steenkamp (checked)** | | **Version 1.1** | | |
| **USE CASE NAME:** | **Search Client** | | | **USE CASE TYPE** |
| **USE CASE ID:** | 5.3 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | Owner | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee needs to search for a client on the system. The user will search for a client by entering a keyword or phrase that matches any existing:  **Client\_ID, Client\_Name, Client\_Surname, Client\_Email, Client\_Contact\_Number, Client\_Address, Client\_Gender, Client\_Instagram** in the **Client table**, into the provided (by the system) search bar.The system will then verify and search for any existing client and display a client that matches the given search criteria. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to search a client on the system. | | | |
| **TRIGGER:** | The employee request to search for a client on the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee requests to search for a client on the system. | | **Step 2:** The system prompts the user to enter the search criteria | |
|  | **Step 3:** The employee will enter any keyword or phrase that will match any existing:  **Client\_ID, Client\_Name, Client\_Surname, Client\_Email, Client\_Contact\_Number, Client\_Address, Client\_Gender, Client\_Instagram** in the **Client table** on the system. | |  | |
|  | **Step 4:** The employee will request to search. | | **Step 5:** The system will capture the entered search keyword or phrase. | |
|  |  | | **Step 6:** The system will validate the entered search criteria so that it is a valid search keyword or phrase. | |
|  |  | | **Step 7:** The system will check the:  **Client\_ID, Client\_Name, Client\_Surname, Client\_Email, Client\_Contact\_Number, Client\_Address, Client\_Gender, Client\_Instagram** details for all existing clients in the **Client table** to see if any match the captured search criteria. | |
|  |  | | **Step 8:** The system returns a list of all clients matching the given search criteria. | |
| **ALTERNATE COURSES:** | **Alt Step 7:** The captured keyword or phrase is invalid.   * Return to step 2. | | | |
|  | **Alt Step 8:** No values on the system match the given search criteria. The system will display an error message.   * Return to step 2. | | | |
| **CONCLUSION:** | The system has successfully retrieved the required client based off the given search criteria and displayed the results to the employee. | | | |
| **POST-CONDITION:** | The system successfully retrieves the client(s) on the system. | | | |
| **BUSINESS RULES** | Only an employee with the required access levels can search for clients. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The client exists on the system | | | |
| **OPEN ISSUES:** | None. | | | |

Table 5.5.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Janco Steenkamp**  Janco Steenkamp (checked) | | **Version 1.3** | | |
| **USE CASE NAME:** | **Sign Up for Subscription** | | | **Use Case Type:** |
| **USE CASE ID:** | 5.4 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Client | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that a client would like to sign up for a subscription on the system. The client will select to sign up for a subscription provided by the system on the subscription page. The system will update the **Client\_ID** of the client  in the **Subscription table**. If everything is done correctly the system will display a success message to the client. | | | |
| **PRE-CONDITION:** | The client must be logged into the system and must not yet be registered for a subscription to be able to sign up for a subscription | | | |
| **TRIGGER:** | The client wants to sign up for a subscription. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The client requests to sign up for a subscription | | **Step 2:** The system will display the subscription options with details regarding the options. | |
|  | **Step 3:** The client will select the option best suited. | | **Step 4:** The system will capture the entered detail. | |
|  |  | | **Step 5:** The system will validate that the client is not previously registered for the subscription | |
|  |  | | **Step 6:** The system will automatically update the **Client\_ID** in the **Subscription** **table**. | |
|  |  | | **Step 7:**Invoke **Use Case 5.17 Make Payment** | |
|  |  | | **Step 8:** The system will display a success message to the client and return the client to the home page. | |
|  |  | |  | |
| **ALTERNATE COURSES:** | **Alt Step 5:** The client is already signed up for a subscription and is invalid. The system will display an error message.   * Return the user to step 3. | | | |
| **CONCLUSION:** | The use case ends when the client has successfully signed up for a subscription and the system is updated. | | | |
| **POST-CONDITION:** | The **Client\_ID** of the **Client** has been successfully updated in the **Subscription table** on the system. | | | |
| **BUSINESS RULES** | Only a client that exists on the system can sign up for a subscription | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The client exists on the system and has not yet registered for the subscription | | | |
| **OPEN ISSUES:** | None. | | | |

Table 5.5.4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Janco Steenkamp**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Cancel Subscription** | | | **Use Case Type:** |
| **USE CASE ID:** | 5.5 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Client | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that a client would like to cancel his/her subscription. The client will navigate to the subscription page and select the **Cancel Subscription** option. The system will validate the status of the client, the system will remove the **Client\_ID** of the client from the **Subscription table**. The client will lose all subscription advantages on the system. | | | |
| **PRE-CONDITION:** | The client must exist on the system and currently be signed up for a subscription option | | | |
| **TRIGGER:** | The client requests to cancel his/her subscription. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The client requests to cancel his/her subscription. | | **Step 2:** The system will display the subscription page with available options**.** | |
|  | **Step 3:** The client will select the **Cancel Subscription** option | | **Step 4:** The system will capture the entered detail | |
|  |  | | **Step 5:** The system will check to see if the client is currently signed up for a subscription by checking for the **Client\_ID** in the **Subscription table**. | |
|  |  | | **Step 6:** The system will remove the **Client\_ID** from the **Subscription table** | |
|  |  | | **Step 7:** The system will display a success message to the client and return the employee to the home page. | |
|  |  | |  | |
|  |  | |  | |
| **ALTERNATE COURSES:** | **Alt Step 5:** The client is not signed up for any subscription thus is invalid. The system will display an error message   * Return the user to step 3. | | | |
| **CONCLUSION:** | The use case ends when the **Client\_ID** has been removed from the **Subscription table** and the client has successfully cancelled his/her subscription | | | |
| **POST-CONDITION:** | A client has been lost all subscription privileges on the system. | | | |
| **BUSINESS RULES** | Only a client already signed up for a subscription can cancel the subscription. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The client is signed up for a subscription | | | |
| **OPEN ISSUES:** | None. | | | |

Table 5.5.5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Janco Steenkamp**  Janco Steenkamp (checked) | | **Version 1.3** | | |
| **USE CASE NAME:** | **Register for a Challenge** | | | **Use Case Type:** |
| **USE CASE ID:** | 5.6 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Client | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that a client would like to register for a challenge on the system. The client will select to register for a challenge provided by the system on the challenge page. The system will update the **Challenge\_Status** of the client  in the **Client table**. If everything is done correctly the system will display a success message to the client. | | | |
| **PRE-CONDITION:** | The client must be logged into the system and must not yet be registered for a challenge to be able to register for a challenge. | | | |
| **TRIGGER:** | The client wants to sign up for a subscription. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The client wants to register for a challenge. | | **Step 2:** The system will display the challenge details and options. | |
|  | **Step 3:** The client will select the option best suited. | | **Step 4:** The system will capture the entered detail. | |
|  |  | | **Step 5:** The system will validate that the client is not previously registered for the challenge | |
|  |  | | **Step 6:** The system will automatically update the **Challenge\_Status** in the **Client** **table**. | |
|  |  | | **Step 7:**Invoke **Use Case 5.17 Make Payment** | |
|  |  | | **Step 8:** The system will display a success message to the client and return the client to the home page. | |
|  |  | |  | |
| **ALTERNATE COURSES:** | **Alt Step 5:** The client is already registered for a challenge and is thus invalid. The system will display an error message.   * Return the user to step 3. | | | |
| **CONCLUSION:** | The use case ends when the client has successfully registered for a challenge and the system is updated. | | | |
| **POST-CONDITION:** | The **Challenge\_Status** of the **Client** has been successfully updated on the system. | | | |
| **BUSINESS RULES** | Only a client that exists on the system can register for a challenge | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | None. | | | |
| **OPEN ISSUES:** | None. | | | |

Table 5.5.6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Iain Steyn** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Unregister for Challenge** | | | **Use Case Type:** |
| **USE CASE ID:** | 5.7 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Client | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that a client would like to unregister for a challenge. The client will navigate to the challenge page and select the **Cancel Challenge** option. The system will validate the status of the client, the system will update the **Challenge\_Status** of the client in the **Client table**. The client will lose all challenge advantages on the system. | | | |
| **PRE-CONDITION:** | The client must exist on the system and currently be registered for a challenge option | | | |
| **TRIGGER:** | The client wants to unregister for a challenge. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The client wants to unregister for a challenge. | | **Step 2:** The system will display the challenge page with available options**.** | |
|  | **Step 3:** The client will select the **Cancel Challenge** option | | **Step 4:** The system will capture the entered detail | |
|  |  | | **Step 5:** The system will check to see if the client is currently registered for a challenge by checking **Challenge\_Status**. | |
|  |  | | **Step 6:** The system will update the **Challenge\_Status** in the **Client table**. | |
|  |  | | **Step 7:** The system will display a success message to the client and return the client to the home page. | |
|  |  | |  | |
|  |  | |  | |
| **ALTERNATE COURSES:** | **Alt Step 5:** The client is not registered for any challenge thus is invalid. The system will display an error message   * Return the user to step 3. | | | |
| **CONCLUSION:** | The use case ends when the **Challenge\_Status** has been updated and the client has successfully unregistered for a challenge. | | | |
| **POST-CONDITION:** | A client’s status has been updated and the challenge has been unregistered. | | | |
| **BUSINESS RULES** | Only a client already registered for a challenge can unregister for a challenge. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | None. | | | |
| **OPEN ISSUES:** | None. | | | |

Table 5.5.7

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Iain Steyn** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Purchase Exercise Plan** | | | **Use Case Type:** |
| **USE CASE ID:** | 5.8 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis:◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Client | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | Employee | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | Owner | | | |
| **DESCRIPTION:** | The Use Case describes the event that a client would like to purchase an exercise plan. The client will select an exercise plan on the exercise plan page which will be displayed by the system when the user navigates to the exercise plan. The client will receive all the required details to complete the transaction such as:  **Account\_Name, Account\_No, Account\_Type, Account\_Branch, Account\_Branch\_Code,**  which will be used to make a payment to the business’ bank account. The system will then capture and validate the payment details - provided by the user. If everything is done correctly the system will display a success message to the employee. | | | |
| **PRE-CONDITION:** | The client must be logged into the system to purchase an exercise plan from the business. | | | |
| **TRIGGER:** | The client requests to purchase an exercise plan. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The client navigates to the exercise plan screen | |  | |
|  | **Step 2**: The client selects the exercise plan to be purchased | | **Step 3**: The system displays the exercise plan information on the screen along with an option to purchase the plan. | |
|  | **Step 4**: The client selects the option to purchase the selected plan | | **Step 5**: The system displays the business’ banking details:  **Account\_Name, Account\_No, Account\_Type, Account\_Branch, Account\_Branch\_Code**  on-screen along with instructions which the client needs to follow to successfully purchase the exercise plan | |
|  | **Step 6**: The client follows the instructions and sends the proof of payment to the necessary email address | | **Step 7:** Invoke **Use Case 1.9 Confirm payment** | |
|  |  | | **Step 8:** The system will add the transaction details:  **Exercise\_Plan\_ID, Sale\_ID, Client\_ID, Workout\_ID, Exercise\_Plan\_ID**  to the **Exercise\_Plan\_Sale\_Line table** | |
|  |  | | **Step 9:** The system will display a confirmation message and notify the client that we are awaiting payment confirmation. | |
|  |  | |  | |
| **ALTERNATE COURSES:** |  | | | |
| **CONCLUSION:** | The use case ends when the client successfully makes the payment to the business’ bank account. | | | |
| **POST-CONDITION:** | A new **Exercise\_Plan\_Sale\_Line** item has been added to the system | | | |
| **BUSINESS RULES** | Only a client registered on the system may purchase an exercise plan. VAT is added at 15%. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The client is registered on the system and the client has followed the payment instructions accordingly | | | |
| **OPEN ISSUES:** | None. | | | |

Table 5.5.8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Iain Steyn**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Purchase Product** | | | **Use Case Type:** |
| **USE CASE ID:** | 5.9 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Client | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | Employee | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | Owner | | | |
| **DESCRIPTION:** | The Use Case describes the event that a client would like to purchase a product. The client will select a product on the product page which will be displayed by the system when the user navigates to the product. The client will receive all the required details to complete the transaction such as:  **Account\_Name, Account\_No, Account\_Type, Account\_Branch, Account\_Branch\_Code,**  which will be used to make a payment to the business’ bank account. The system will then capture and validate the payment details - provided by the user. If everything is done correctly the system will display a success message to the employee. | | | |
| **PRE-CONDITION:** | The client must be logged into the system make a payment to purchase a product from the business. | | | |
| **TRIGGER:** | The client requests to purchase a product. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The client navigates to the product screen | |  | |
|  | **Step 2**: The client selects the product to be purchased | | **Step 3**: The system displays the product information on the screen along with an option to purchase the plan. | |
|  | **Step 4**: The client selects the option to purchase the selected plan | | **Step 5**: The system displays the business’ banking details on-screen:  **Account\_Name, Account\_No, Account\_Type, Account\_Branch, Account\_Branch\_Code,**  along with instructions which the client needs to follow to successfully purchase the product | |
|  | **Step 6**: The client follows the instructions and sends the proof of payment to the necessary email address | | **Step 7:** Invoke **Use Case 1.9 Confirm payment** | |
|  |  | | **Step 8:** The system will add the transaction details:  **Product\_ID, Sale\_ID, Client\_ID, Quantity, Amount**  to the **Product\_Sale\_Line table** | |
|  |  | | **Step 9:** The system will display a confirmation message and notify the client that we are awaiting payment confirmation. | |
|  |  | |  | |
| **ALTERNATE COURSES:** | **Alt step 4:** The client opts to not select a product to purchase   * Use case is terminated and is navigated to the product page | | | |
| **CONCLUSION:** | The use case ends when the client successfully makes the payment to the business’ bank account. | | | |
| **POST-CONDITION:** | A new **Product\_Sale\_Line** item has been added to the system | | | |
| **BUSINESS RULES** | Only a client registered on the system may purchase a product. VAT is added at 15%. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The product is on hand  The client is logged in  The client followed the payment instructions accordingly | | | |
| **OPEN ISSUES:** | None. | | | |

Table 5.5.9

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Iain Steyn** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Make New Personal Training Booking** | | | **Use Case Type:** |
| **USE CASE ID:** | 5.10 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: 🞎 |
| **SOURCE:** | JessterFIT | | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Client | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that a client would like to add a new personal training booking slot to the system. The client will add a new booking personal training booking slot on the add new personal training booking page which will be displayed by the system when the client wants to add a new personal training booking slot. The client will add all the required details for the new personal training booking slot such as:  **Personal\_Training\_Date,**  which will be saved in the **Personal Training table.** The system will then capture and validate the entered details and check to see if the personal training booking slot - provided by the user – doesn’t already exist on the system. If everything is done correctly the system will display a success message to the employee. | | | |
| **PRE-CONDITION:** | The client must be logged into the system and have the right access levels and functionality to be able to add a new personal training booking slot to the system. | | | |
| **TRIGGER:** | The client wants to add a new personal booking slot to the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The client wants to add a new personal training booking slot to the system. | | **Step 2:** The system will display the add new personal training booking slot screen and inform the employee to enter a new personal training booking slot detail:  **Personal\_Training\_Date,** | |
|  | **Step 3:** The client will enter a new **Personal\_Training\_Date,** on the add personal training booking page and save it on the system. | | **Step 4:** The system will capture the entered detail:  **Personal\_Training\_Date, Client\_ID** | |
|  |  | | **Step 5:** The system will validate the entered date so that it is a valid personal training booking date. | |
|  |  | | **Step 6:** The system will check to see if the booking date added does not already exist in the system. | |
|  |  | | **Step 7:** The system will automatically generate a unique **Training \_ID** and add it to the **Personal\_Training\_Booking table**. | |
|  |  | | **Step 8:** The system will add the date and client\_id to the **Personal\_Training\_Booking table** | |
|  |  | | **Step 9:** The system will display a success message to the employee and return the employee to the home page. | |
| **ALTERNATE COURSES:** | **Alt Step 5:** The entered **Personal\_Training\_Date,** is invalid. The system will display an error message.   * Return the user to step 3. | | | |
| **CONCLUSION:** | The use case ends when the new personal training booking slot has been successfully added to the system and a success message is displayed for the employee. | | | |
| **POST-CONDITION:** | A new personal training booking slot has been successfully added to the system. | | | |
| **BUSINESS RULES** | Only a client with an active subscription add a personal training booking slot. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | None. | | | |
| **OPEN ISSUES:** | None. | | | |

Table 5.5.10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Iain Steyn**  **Janco Steenkamp(checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Cancel personal training booking** | | | **Use Case Type:** |
| **USE CASE ID:** | 5.11 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Client | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that a client would like to cancel an existing personal training booking slot on the system. The client will cancel the booking personal training booking slot on the personal training booking page which will be displayed by the system. The client will select to cancel the relevant booking slot from the displayed booking slots previously selected which will be retrieved from the **Personal\_Training\_Booking table.** The system will then capture the request. If everything is done correctly the system will display a success message to the client. | | | |
| **PRE-CONDITION:** | The client must be logged into the system and have the right access levels and functionality to be able to cancel a personal training booking slot on the system. | | | |
| **TRIGGER:** | The client requests to cancel a personal training booking slot on the system. | | | |
| **TYPICAL COURSE**  **OF EVENTS:** | Actor Action | | System Response | |
| **Step 1**: The client requests to cancel a personal training booking slot on the system. | | **Step 2:** The system will display the personal training booking slots which the client has selected to attend on the screen. Retrieved from the **Personal\_Training\_Booking table** | |
| **Step 3:** The client will select to cancel the relevant personal training booking slots | | **Step 4:** The system will capture the entered request | |
|  | | **Step 5:** The system will validate the selected date so that it is a valid personal training booking date. | |
|  | | **Step 6:** The system will automatically remove the relevant **Training\_Booking\_ID** with its details from the **Personal\_Training\_Booking table**. | |
|  | | **Step 7:** The system will display a success message to the client, return the client to the home page and notify the employee associated with the relevant personal training booking slot | |
| **ALTERNATE COURSES:** | **Alt Step 5:** The selected **Training\_Booking\_ID** is invalid. The system will display an error message.   * Return the user to step 3. | | | |
| **CONCLUSION:** | The use case ends when an existing personal training booking slot has been successfully removed from the **Personal\_Training table** on the system and a success message is displayed to the client. | | | |
| **POST-CONDITION:** | An existing personal training booking slot has been successfully removed and the system is up to date. | | | |
| **BUSINESS RULES** | Only a client with an active subscription can remove a personal training booking slot. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The booking slot exists and previously selected by the client. | | | |
| **OPEN ISSUES:** | None. | | | |

Table 5.5.11

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Iain Steyn** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Make New Challenge Consultation Booking** | | | **Use Case Type:** |
| **USE CASE ID:** | 5.12 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: 🞎 |
| **SOURCE:** | JessterFIT | | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Client | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | Employee. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | Owner. | | | |
| **DESCRIPTION:** | The Use Case describes the event that a client would like to add a new challenge consultation booking slot to the system. The client will add a new challenge consultation booking slot on the add new challenge consultation booking page which will be displayed by the system when the client wants to add a new challenge consultation booking slot. The client will add all the required details for the new challenge consultation booking slot such as:  **Challenge\_Consultation\_Date,**  which will be saved in the **Challenge\_Consultation table.** The system will then capture and validate the entered details and check to see if the challenge consultation booking slot - provided by the user – doesn’t already exist on the system. If everything is done correctly the system will display a success message to the employee. | | | |
| **PRE-CONDITION:** | The client must be logged into the system and have the right access levels and functionality to be able to add a new challenge consultation booking slot to the system. | | | |
| **TRIGGER:** | The client wants to add a new challenge consultation booking slot to the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The client wants to add a new challenge consultation booking slot to the system. | | **Step 2:** The system will display the add new challenge consultation booking slot screen and inform the employee to enter a new challenge consultation booking slot detail:  **ChallengeConsultation\_Date,** | |
|  | **Step 3:** The client will enter a new **Challenge\_Consultation\_Date,** on the add challenge consultation booking page and save it on the system. | | **Step 4:** The system will capture the entered detail:  **Challenge\_Consultation\_Date,** | |
|  |  | | **Step 5:** The system will validate the entered date so that it is a valid challenge consultation booking date. | |
|  |  | | **Step 6:** The system will check to see if the booking date added does not already exist in the system. | |
|  |  | | **Step 7:** The system will automatically generate a unique **Challenge \_ID** and add it to the **Consultation\_Booking table**. | |
|  |  | | **Step 8:** The system will add the date to the **Consutation\_Booking table** and save it as a:  **Consultation\_Date,**. | |
|  |  | | **Step 9:** The system will display a success message to the employee and return the employee to the home page. | |
| **ALTERNATE COURSES:** | **Alt Step 5:** The entered **Consultation\_Date,** is invalid. The system will display an error message.   * Return the user to step 3. | | | |
| **CONCLUSION:** | The use case ends when the new challenge consultation booking slot has been successfully added to the system and a success message is displayed for the employee. | | | |
| **POST-CONDITION:** | A new challenge consultation booking slot has been successfully added to the system. | | | |
| **BUSINESS RULES** | Only a client with an active challenge may add a challenge consultation booking slot. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | None. | | | |
| **OPEN ISSUES:** | None. | | | |

Table 5.5.12

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Iain Steyn**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Cancel Challenge Consultation Booking** | | | **Use Case Type:** |
| **USE CASE ID:** | 5.13 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Client | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that a client would like to cancel an existing challenge consultation booking slot on the system. The client will cancel the challenge consultation booking slot on the consultation booking page which will be displayed by the system. The client will select to cancel the relevant booking slot from the displayed booking slots previously selected which will be retrieved from the **Consultation\_Booking table.** The system will then capture the request. If everything is done correctly the system will display a success message to the client. | | | |
| **PRE-CONDITION:** | The client must be logged into the system and have the required access levels and functionality to be able to cancel a challenge consultation booking slot on the system. | | | |
| **TRIGGER:** | The client requests to cancel a challenge consultation booking slot on the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The client requests to cancel a challenge consultation booking slot on the system. | | **Step 2:** The system will display the challenge consultation booking dates which the client has selected to attend in a list. Retrieved from the **Consultation\_Booking table** | |
|  | **Step 3:** The client will select to cancel the relevant challenge consultation booking slots | | **Step 4:** The system will capture the entered request | |
|  |  | | **Step 5:** The system will validate the selected slot so that it is a valid challenge consultation booking slot. | |
|  |  | | **Step 6:** The system will automatically remove the relevant **Consultation\_ID** with its details from the **Consultation\_Booking table**. | |
|  |  | | **Step 7:** The system will display a success message to the client, return the client to the home page and notify the employee associated with the relevant challenge consultation booking slot. | |
|  |  | |  | |
|  |  | |  | |
| **ALTERNATE COURSES:** | **Alt Step 5:** The selected **Consultation\_ID** is invalid. The system will display an error message.   * Return the user to step 3. | | | |
| **CONCLUSION:** | The use case ends when an existing challenge consultation booking slot has been successfully canceled on the system and a success message is displayed to the client. | | | |
| **POST-CONDITION:** | An existing challenge consultation booking slot has been successfully removed from the **Consultation\_Booking table** on the system. | | | |
| **BUSINESS RULES** | Only a client with an active challenge can remove a challenge consultation booking slot | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The challenge booking consultation slot exists and had been made by the client | | | |
| **OPEN ISSUES:** | None. | | | |

Table 5.5.13

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Janco Steenkamp**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Make Payment** | | | **Use Case Type:** |
| **USE CASE ID:** | 5.14 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Client | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | Employee | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | Owner | | | |
| **DESCRIPTION:** | The Use Case describes the event that a client would like to make a recurring payment to the business for a subscription. Given a predetermined time, the system will check the if the client’s subscription is still valid and if there are any outstanding payments. The client will receive all the required details to complete the payment such as:  **Account\_Name, Account\_No, Account\_Type, Account\_Branch, Account\_Branch\_Code, Subscription\_ID**  which will be used to make the payment to the business’ bank account. The system will then capture and validate the payment details - provided by the user. If everything is done correctly the system will display a success message to the employee. | | | |
| **PRE-CONDITION:** | The client must be signed up for a subscription on the system to activate the make payment use case | | | |
| **TRIGGER:** | A specific time of the month is at hand when subscription need to be renewed. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** |  | | **Step 1:** The system notifies the client that it is time to pay their subscription. | |
|  | **Step 2**: The client selects to make the payment | | **Step 3**: The system displays the payment information on the screen:  **Business\_Banking\_Details, Subscription\_ID, Balance, Start\_Date, End\_Date**  along with steps to complete the transaction. | |
|  | **Step 5**: The client follows the instructions and sends the proof of payment to the necessary email address | | **Step 6:** Invoke **Use Case 1.9 Confirm payment** | |
|  |  | | **Step 7:** The system will update the relevant subscription details:  **Balance, Start\_Date, End\_Date**  to the **Subscription table** | |
|  |  | | **Step 8:** The system will display a confirmation message and notify the client that we are awaiting payment confirmation. | |
|  |  | |  | |
|  |  | |  | |
| **ALTERNATE COURSES:** | **Alt step 2:** The client opts to not make the payment   * The subscription will be canceled, and the system will notify the client of the canceled subscription | | | |
| **CONCLUSION:** | The use case ends when the client makes the payment to the business’ bank account. | | | |
| **POST-CONDITION:** | An updated **Subscription** item has been added to the system | | | |
| **BUSINESS RULES** | Only a client registered for a subscription is available to make a payment regarding subscriptions. VAT is added at 15%. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | None. | | | |
| **OPEN ISSUES:** | None. | | | |

Table 5.5.14

|  |  |
| --- | --- |
| **Authors(s): Joshua Eales**  **Janco Steenkamp (checked)** | **Version 1.3** |

6. Reporting Subsystem

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | **Generate Sales Report** | | **USE CASE TYPE** |
| **USE CASE ID:** | 6.1 | | Business Requirements: ◻ |
| **PRIORITY:** | High | | System Analysis: |
| **SOURCE:** | Requirement - Owner | | System Design: |
| **PRIMARY BUSINESS ACTOR** | Owner | | |
| **PRIMARY SYSTEM ACTOR** | None | | |
| **OTHER PARTICIPATING ACTORS:** | None | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None | | |
| **DESCRIPTION:** | The use case describes the event where the owner requests to generate a sales report. System will display the report generating screen and will prompt the user to enter a start date and end date isolating sales over a certain period. Report generated will retrieve information from all sales transactions completed within the time specified by the owner. System retrieves **Product\_Name** and **Product\_Price** from **Product table** and **Exercise\_Plan\_Name** and **Exercise\_Plan\_Price** from **Exercise table**. System retrieves quantity of products and exercises plans sold will be retrieved from **Product\_Sale\_Line** and **Exercise\_Sale\_Line**. System will retrieve **Sale\_Date** and **Sale\_Amount** from **Sale table.** Once the system has retrieved the required sales information. The system will generate the report creating an onscreen display which will be split by **Products** and **Exercises** and displays it on the screen allowing the owner to see how sales are doing in each individual product offered by the business. **Product\_Name** and **Exercise\_Plane\_Name** will be included on the report as well as amounts bought and the total amount for the date. A grand total adding all sales will be at the bottom representing the total of all sales during the time specified. There will also be options for a pdf or word document of the report to be saved or printed. | | |
| **PRE-CONDITION:** | Owner must be logged into the system. | | |
| **TRIGGER:** | The owner requests to generate a sales report. | | |
| **TYPICAL COURSE** | Actor Action | System Response | |
| **OF EVENTS:** | **Step 1**: Owner requests to generate a sales report for a specific period. |  | |
|  |  | **Step 2**: System returns generate sales report screen and prompts owner to enter a **Start\_Date** and **End\_Date** for the period they would like to report on. | |
|  | **Step 3**: Owner enters **Start\_Date** and **End\_Date**. |  | |
|  |  | **Step 4**: System captures the details entered | |
|  |  | **Step 5**: System validates **Start\_Date** and **End\_Date** to ensure dates specified are not any future dates. | |
|  |  | **Step 6**: System retrieves **Product\_Name** and **Product\_Price** from **Product table** and **Exercise\_Plan\_Name** and **Exercise\_Plan\_Price** from **Exercise\_Plan table** within period specified. | |
|  |  | **Step 7**: System retrieves quantity of products and exercises plans bought both will be retrieved from **Product\_Sale\_Line** and **Exercise\_Sale\_Line**. | |
|  |  | **Step 8**: System will retrieve **Sale\_Date** and **Sale\_Amount** from **Sale table.** | |
|  |  | **Step 9**: Systems joins all details retrieved and compiles an onscreen report with the details. | |
|  |  | **Step 10**: System generates an onscreen report for the owner and makes options available for downloadable pdf and word formats of the report. | |
|  |  | **Step 11**: System displays the report to the user. | |
|  | **Step 12**: The user completes the use case by only using the on-screen report |  | |
| **ALTERNATE COURSES:** | **Alt step 4**: Start and end date entered are not valid, display an error message.   * Return to step 3. | | |
|  | **Alt Step 12a**: Owner chooses to download the pdf format of the report | | |
|  | **Alt Step 12b**: Owner chooses to download the word format of the report. | | |
| **CONCLUSION:** | The use case concludes when the sales report is successfully displayed to the owner. | | |
| **POST-CONDITION:** | Details displayed by the system are correct for the period selected. | | |
| **BUSINESS RULES** | Only the owner has access to sales report details. | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | |
| **ASSUMPTIONS:** | Business has completed sales transactions during the month. | | |
| **OPEN ISSUES:** | None | | |

Table 6.6.1

|  |  |
| --- | --- |
| JessterFIT | **Date: 2019/04/28** |
| **Authors(s): Joshua Eales**  **Janco Steenkamp (checked)** | **Version 1.3** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | **Generate Subscription Report** | | **USE CASE TYPE** |
| **USE CASE ID:** | 6.2 | | Business Requirements: ◻ |
| **PRIORITY:** | High | | System Analysis: |
| **SOURCE:** | Requirement - Owner | | System Design: |
| **PRIMARY BUSINESS ACTOR** | Owner | | |
| **PRIMARY SYSTEM ACTOR** | None | | |
| **OTHER PARTICIPATING ACTORS:** | None | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None | | |
| **DESCRIPTION:** | The use case describes the event where the owner requests to generate a subscription report. System will display the report generating screen and will prompt the user to enter a **start date** and **end date** isolating number of subscriptions over a certain period. Report generated will retrieve information from all subscription registrations transactions completed within the time specified by the owner. System retrieves **Client\_Name** and **Client\_ID** from **Client table**. System retrieves **Subscription\_Type\_Description** and **Subscription\_Type\_Price** from **Subscription\_Type table.**  System will then retrieve **Subscription\_Payment\_Date** and **Subscription\_Payment\_Amount** from **Subscription\_Payment table**. Once the system has retrieved the required subscription information. The system will generate the report creating an onscreen display allowing the owner to see how many clients have subscribed in the period specified. A grand total adding all subscriptions amounts will be at the bottom representing the total of all subscriptions during the time specified. There will also be options for a pdf or word document of the report to be saved or printed. | | |
| **PRE-CONDITION:** | Owner must be logged into the system. | | |
| **TRIGGER:** | The owner requests to generate subscription report. | | |
| **TYPICAL COURSE** | Actor Action | System Response | |
| **OF EVENTS:** | **Step 1**: Owner requests to generate a subscription report for requested period. |  | |
|  |  | **Step 2**: System will display the report generating screen and will prompt the user to enter a **Start\_Date** and **End\_Date** isolating number of subscriptions over a certain period. | |
|  | **Step 3**: Owner enters **Start\_Date** and **Start\_Date**. |  | |
|  |  | **Step 4**: System captures start and end date information. | |
|  |  | **Step 5**: System validates start and end date to ensure dates specified are not any future dates. | |
|  |  | **Step 6**: System retrieves **Client\_Name** and **Client\_ID** from **Client table.** | |
|  |  | Step 7: System retrieves **Subscription\_Type\_Description** and **Subscription\_Type\_Price** from **Subscription\_Payment table.** | |
|  |  | Step 8: System will then retrieve **Sub\_Payment\_Date** and **Sub\_Payment\_Amount** from **Subscription\_Payment table** | |
|  |  | **Step 9**: Systems joins all details retrieved. | |
|  |  | **Step 10**: System generates an onscreen report for the owner and makes options available for downloadable pdf and word formats of the report. | |
|  |  | **Step 11**: System displays the report to the owner. | |
|  | **Step 12**: Owner chooses to use onscreen report and complete the use case. |  | |
|  | **Alt step 4**: Start and end date entered are not valid, display an error message.   * Return to step 3. | | |
| **ALTERNATE COURSES:** | **Alt Step 12a**: Owner chooses to download the pdf format of the report. | | |
|  | **Alt Step 12aa**: The system generates a report in pdf format | | |
|  | **Alt Step 12b**: Owner chooses to download the word format of the report. | | |
|  | **Alt Step 12bb**: The system generates a report in a word document format. | | |
| **CONCLUSION:** | The use case concludes when the subscription report is successfully displayed to the owner. | | |
| **POST-CONDITION:** | Details displayed by the system are correct for the months subscription details. | | |
| **BUSINESS RULES** | Only the owner has access to sales report details. | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | |
| **ASSUMPTIONS:** | Business has completed subscription registrations during the month. | | |
| **OPEN ISSUES:** | None | | |

Table 6.6.2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **JessterFIT** | | | **Date: 2019/04/28** | | |
| **Authors(s): Joshua Eales**  **Janco Steenkamp (checked)** | | | **Version 1.3** | | |
| **USE CASE NAME:** | **Generate Client Progress Report** | | | **USE CASE TYPE** | |
| **USE CASE ID:** | 6.3 | | | Business Requirements: ◻ | |
| **PRIORITY:** | High | | | System Analysis: | |
| **SOURCE:** | Requirement - Owner | | | System Design: | |
| **PRIMARY BUSINESS ACTOR** | Owner | | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | Employee (personal trainer) | | | | |
| **DESCRIPTION:** | The use case describes the event where the owner wishes to generate a client progress report. The system will return the generate client progress report screen and ask the owner to enter which month they want to generate the report for. The system will invoke use case **5.3 Search Client**. The system will retrieve the required client details: **Client\_Name, Client\_Surname** from the **Client table.** The report will contain the previous months’ **client health info** retrieved from the **Client\_Bio\_Form table** and the selected month’s **client health information** for comparison. The system will then return the generated report creating an onscreen display and allowing a pdf or word document of the report to be saved. | | | | |
| **PRE-CONDITION:** | Owner must be logged into the system. | | | | |
| **TRIGGER:** | The owner requests to generate a client progress report. | | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | | |
| **OF EVENTS:** | **Step 1**: The owner requests to generate a client progress report. | |  | | |
|  |  | | **Step** **2**: The system will return the generate client progress report screen and prompt the owner to enter which month they want to generate the report for. | | |
|  | **Step 3:** Owner enters month. | |  | | |
|  |  | | **Step 4**: System captures the entered details. | | |
|  |  | | **Step 5**: System validates details to ensure dates specified are not any future dates. | | |
|  |  | | **Step 6**: Invoke use case **5.3 Search client.** | | |
|  | **Step 7:** Owner enters **Client\_ID** or **Client\_Name** into search bar. | |  | | |
|  |  | | **Step 8**: System captures search information. | | |
|  |  | | **Step 9**: System validates search criteria specified. | | |
|  |  | | **Step 10**: System will retrieve the required client details: **Client\_Name, Client\_Surname** from the **Client table** and **Client Health Info** from the **Client\_Bio\_Form table.** | | |
|  |  | | **Step** 11: System will retrieve previous months progress details. | | |
|  |  | | **Step 12**: Systems joins all details retrieved. | | |
|  |  | | **Step 13**: System generates an onscreen report for the owner and makes options available for downloadable pdf and word formats of the report. | | |
|  |  | | **Step 14**: System displays the report to the owner. | | |
|  | **Step 15**: Owner selects to use onscreen report and completes the use case. | |  | | |
| **ALTERNATE COURSES:** | **Alt Step 5**: Search criteria is invalid, display an error message   * Return to Step3. | | | | |
|  | **Alt step 9**: Search criteria is invalid, display error message   * Return to Step 7. | | | | |
|  | **Alt step 15a**: Owner chooses to download the pdf format of the report. | | | | |
|  | **Alt step 15aa**: The system generates a report in pdf format | | | | |
|  | **Alt step 15b**: Owner chooses to download the word format of the report. | | | | |
|  | **Alt step 15bb**: The system will generate the report in a word document format | | | | |
| **CONCLUSION:** | The use case concludes when the client progress report is successfully displayed to the owner. | | | | |
| **POST-CONDITION:** | Details displayed by the system are correct for the selected individual. | | | | |
| **BUSINESS RULES** | Only the owner has access to customer progress details. | | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | | |
| **ASSUMPTIONS:** | Client information exists on the system. | | | | |
| **OPEN ISSUES:** | None | | | | |

Table 6.6.3

|  |  |
| --- | --- |
| JessterFIT | **Date: 2019/04/28** |
| **Authors(s): Joshua Eales** | **Version 1.3** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | **Generate product levels Report** | | **USE CASE TYPE** |
| **USE CASE ID:** | 6.4 | | Business Requirements: ◻ |
| **PRIORITY:** | High | | System Analysis: |
| **SOURCE:** | Requirement - Owner | | System Design: |
| **PRIMARY BUSINESS ACTOR** | Employee | | |
| **PRIMARY SYSTEM ACTOR** | None | | |
| **OTHER PARTICIPATING ACTORS:** | None | | |
| **OTHER INTERESTED STAKEHOLDERS:** | Owner | | |
| **DESCRIPTION:** | The use case describes the event where the employee requests to generate a product levels report. System will display the product report generating screen and will prompt the user to enter a start date and end date isolating product levels over a certain period. The system retrieves **Product\_Name, Product\_ID** and **Product\_Quantity** from **Product table** and **Product\_Type\_Descrition** from **Product\_Type table**. The system will generate the report creating an onscreen display allowing the employee to see how much stock is on hand in the period specified. There will also be options for a pdf or word document of the report to be saved or printed. | | |
| **PRE-CONDITION:** | Employee must be logged into the system. | | |
| **TRIGGER:** | The employee requests to generate a product levels report. | | |
| **TYPICAL COURSE** | Actor Action | System Response | |
| **OF EVENTS:** | **Step 1**: Employee requests to generate a product levels report. |  | |
|  |  | **Step 2**: The system will return the generate product levels report screen and ask the owner to enter a **Start\_Date** and **End\_Date** they want to generate the report for. | |
|  | **Step 3**: Owner enters **Start\_Date**t and **End\_Date**. |  | |
|  |  | **Step 4**: System captures the entered details. | |
|  |  | **Step 5**: System validates details to ensure dates specified are not any future dates. | |
|  |  | **Step 6**: The system retrieves **Prod\_Name, Prod\_Description and Prod\_Quantity** from **Product table** and **Prod\_Type** from **Product\_Type table**. | |
|  |  | **Step 7**: Systems joins all details retrieved. | |
|  |  | **Step 8:** System generates an onscreen report for the owner and makes options available for downloadable pdf and word formats of the report. | |
|  |  | **Step 9**: System displays the report to the owner. | |
|  | **Step 10**: Owner chooses to use onscreen report and completes the use case. |  | |
|  | **Alt Step 5**: Search criteria is invalid, display error message   * Return to Step1. | | |
| **ALTERNATE COURSES:** | **Alt Step 10a**: Employee chooses to download the pdf format of the report | | |
|  | **Alt Step 10aa**: System generates the report in a pdf format. | | |
|  | **Alt Step 10b**: Employee chooses to download the word format of the pdf. | | |
|  | **Alt Step 10bb**: The system generates the report in a word document format. | | |
| **CONCLUSION:** | When the product levels report is successfully displayed to the employee. | | |
| **POST-CONDITION:** | Details displayed by the system are correct product inventory levels. | | |
| **BUSINESS RULES** | Employee must have approval from owner to order new products. | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | |
| **ASSUMPTIONS:** | Product levels are correct. | | |
| **OPEN ISSUES:** | None | | |

Table 6.6.4

|  |  |
| --- | --- |
| JessterFIT | **Date: 2019/04/28** |
| **Authors(s): Joshua Eales**  **Janco Steenkamp (checked)** | **Version 1.3** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | **Generate Challenge Booking Report** | | **USE CASE TYPE** |
| **USE CASE ID:** | 6.5 | | Business Requirements: ◻ |
| **PRIORITY:** | High | | System Analysis: |
| **SOURCE:** | Requirement - Owner | | System Design: |
| **PRIMARY BUSINESS ACTOR** | Owner | | |
| **PRIMARY SYSTEM ACTOR** | None | | |
| **OTHER PARTICIPATING ACTORS:** | None | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None | | |
| **DESCRIPTION:** | The use case describes the event where the owner requests to generate a challenge booking report. System will display the challenge booking report generating screen and will prompt the user to enter a start date and end date isolating the number of challenge bookings over a certain period. Report generated will retrieve information from all challenge booking registrations transactions completed within the period specified by the owner. System retrieves **Client\_Name** and **Client\_ID** from **Client table**. System retrieves **Challenge\_ID, Challenge\_Start\_Date** and **Challenge\_End\_Date** from **Consultation\_Booking table.**  Once the system has retrieved the required challenge booking information. The system will generate the report creating an onscreen display allowing the owner to see how many clients have booked for challenges in the period specified. A grand total adding all challenge booking amounts will be at the bottom representing the total of all challenges during the time specified. There will also be options for a pdf or word document of the report to be saved or printed. | | |
| **PRE-CONDITION:** | Owner must be logged into the system. | | |
| **TRIGGER:** | The owner requests to generate a subscription report. | | |
| **TYPICAL COURSE** | Actor Action | System Response | |
| **OF EVENTS:** | **Step 1**: Owner requests to generate a challenge booking report for requested period. |  | |
|  |  | **Step 2**: System will display the challenge booking report generating screen and will prompt the user to enter a **Start\_Date** and **End\_Date** isolating number of challenge bookings over a certain period. | |
|  | **Step 3**: Owner enters **Start\_Date** and **End\_Date**. |  | |
|  |  | **Step 4**: System captures **Start\_Date** and **Start\_Date**. | |
|  |  | **Step 5:** System validates entered details to ensure dates specified are not any future dates. | |
|  |  | **Step 6**: System retrieves **Client\_Name** and **Client\_ID** from **Client table**. | |
|  |  | **Step 7**: System retrieves **Challenge\_ID, Challenge\_Start\_Date** and **Challenge\_End\_Date** from **Consultation\_Booking table.** | |
|  |  | **Step 8**: Systems joins all details retrieved. | |
|  |  | **Step 9**: System generates an onscreen report for the owner and makes options available for downloadable pdf and word formats of the report. | |
|  |  | **Step** 10: System displays the report to the owner. | |
|  | **Step 11**: Owner selects to use onscreen report and complete the use case. |  | |
|  | **Alt** **step 5**: **Start\_Date** and **End\_Date** entered are not valid, display error meassage.   * Return to step 3. | | |
| **ALTERNATE COURSES:** | **Alt Step 11a**: Owner chooses to download the pdf format of the report. | | |
|  | **Alt step 11aa**: The system generates a report in pdf format. | | |
|  | **Alt Step 11b**: Owner chooses to download the word format of the pdf. | | |
|  | **Alt Step 11bb**: The system generates the report in a Microsoft word document format. | | |
| **CONCLUSION:** | When the challenge booking report is successfully displayed to the owner. | | |
| **POST-CONDITION:** | Details displayed by the system are correct for the selected period. | | |
| **BUSINESS RULES** | Only the owner has access to all report details. | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | |
| **ASSUMPTIONS:** | There is a challenge active in the selected period | | |
| **OPEN ISSUES:** | None | | |

Table 6.6.5

7. Sale Subsystem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **JessterFIT** | | | **Date: 2019/04/28** | | |
| **Authors(s): Joshua Eales**  **Janco Steenkamp (checked)** | | | **Version 1.3** | | |
| **USE CASE NAME:** | **Search Sale** | | | **USE CASE TYPE** | |
| **USE CASE ID:** | 7.1 | | | Business Requirements: ◻ | |
| **PRIORITY:** | High | | | System Analysis: | |
| **SOURCE:** | JessterFIT | | | System Design: | |
| **PRIMARY BUSINESS ACTOR** | Employee | | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | Owner | | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee needs to search for an access level on the system. The user will search for an access level by entering a keyword or phrase that matches any existing:  **Sale\_ID, Client\_ID, Sale\_Date** in the **Sale table**, into the provided (by the system) search bar.The system will then verify and search for any existing access level and display its relevant details that matches the given search criteria. | | | | |
| **PRE-CONDITION:** | Employee must be logged into the system and have the required access level | | | | |
| **TRIGGER:** | Employee request to search a sale on the system. | | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | | |
| **OF EVENTS:** | **Step 1**: Employee requests to search a sale that has been completed. | |  | | |
|  |  | | **Step 2**: System will return the search sale screen and prompt the employee to enter a **Sale\_ID, Client\_ID, Sale\_Date**. | | |
|  | **Step 3**: The employee will enter the **Sale\_ID, Client\_ID, Sale\_Date** needed | |  | | |
|  |  | | **Step 4**: The system will capture the **Sale\_ID, Client\_ID, Sale\_Date**. | | |
|  |  | | **Step 5**: The system will then validate if the **Sale\_ID, Client\_ID, Sale\_Date** is valid | | |
|  |  | | **Step 6**: The system will then retrieve the sale details: **Client\_ID, Sale\_Amount** and **Sale\_Date** from the **Sale table.** | | |
|  |  | | **Step 7**: System displayed results to the employee in the form of a list. | | |
| ALTERNATE COURSES: | **Alt Step 5**: The employee did not enter a valid **Sale\_ID, Client\_ID, Sale\_Date**, display an error message  Return to step 3. | | | | |
|  | **Alt Step 6**: No results matched the **Sale\_ID, Client\_ID, Sale\_Date** entered, display an error message.   * Return to step 3. | | | | |
| **CONCLUSION:** | The use case concludes when the requested sale details have been retrieved from the **Sale table** on the system. | | | | |
| **POST-CONDITION:** | Sale details have been retrieved form the system. | | | | |
| **BUSINESS RULES** | Only employees with the required access level may search sales. | | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | | |
| **ASSUMPTIONS:** | The sale exists on the system | | | | |
| **OPEN ISSUES:** | None | | | | |

Table 7.7.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **JessterFIT** | | | **Date: 2019/04/28** | | |
| **Authors(s): Joshua Eales**  **Janco Steenkamp (checked)** | | | **Version 1.3** | | |
| **USE CASE NAME:** | **Return Sale** | | | **USE CASE TYPE** | |
| **USE CASE ID:** | 7.2 | | | Business Requirements: ◻ | |
| **PRIORITY:** | High | | | System Analysis: | |
| **SOURCE:** | JessterFIT | | | System Design: | |
| **PRIMARY BUSINESS ACTOR** | Client | | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | | |
| **OTHER PARTICIPATING ACTORS:** | Employee | | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | Owner | | | | |
| **DESCRIPTION:** | The use case describes the event where a client requests to return a sale. The system will open the returns screen and prompt the client to enter all information relating to the sale wishing to be returned. The employee will invoke **Use Case 7.1 Search Sale**. The employee will select the appropriate sale. The employee will select to reverse/ remove the sale from the **Sale table**. The system will then reimburse the money to the clients account once the return is submitted and in the case of a product money will be reimbursed once an employee has made sure the product has not been tampered with. Once a return is successful a confirmation message will be displayed to the client. | | | | |
| **PRE-CONDITION:** | Client must be logged into the system. | | | | |
| **TRIGGER:** | Client requests to return a sale. | | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | | |
| **OF EVENTS:** | **Step 1**: Client requests to return a sale. | |  | | |
|  |  | | **Step 2**: The system will open the returns screen and prompt the client to enter details:  **Sale\_ID**  relating to the sale requested to be returned | | |
|  | **Step 3**: The client will enter the requested return details such as: **Sale\_ID** | |  | | |
|  |  | | **Step 4**: The system will capture the return details: **Sale\_ID**. | | |
|  |  | | **Step 5**: The system will then validate if the **Sale\_ID** and **Client\_ID** details are valid. | | |
|  |  | | **Step 6**: The system will remove the sale details from the **Sale table**. | | |
|  |  | | **Step 7**: The system will then reimburse the money to the clients account once the return is submitted. In the case of a product, money will be reimbursed once an employee has made sure the product has not been tampered with. | | |
|  |  | | **Step 8**: The system will display a success message once the return is completed and return the client to the home page | | |
| **ALTERNATE COURSES:** | **Alt Step 5**: **Sale\_ID** and **Client\_ID** could not be validated, display an error message.   * Return to step 3. | | | | |
|  | **Alt Step 7**: Product returned has been tampered with. Terminate use case and inform client that return has been denied. | | | | |
| **CONCLUSION:** | This use case concludes when the amount of the sale has been successfully reimbursed to the clients’ account. | | | | |
| **POST-CONDITION:** | Details entered are correct to the sale wishing to be returned. | | | | |
| **BUSINESS RULES** | Products cannot be tampered with or damaged in any way when returned. | | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | | |
| **ASSUMPTIONS:** | The product has not been tampered with | | | | |
| **OPEN ISSUES:** | None | | | | |

Table 7.7.2

8. Exercise Subsystem

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Iain Steyn**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Add New Exercise Plan Type** | | | **Use Case Type:** |
| **USE CASE ID:** | 8.1 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | Owner. | | | |
| **DESCRIPTION:** | The use case describes the event that an employee requests to add a new exercise plan type to the system. The employee will add a new exercise plan type on the add new exercise plan type page which will be displayed by the system when the employee wants to add a new exercise plan type. The employee will add all the required details for the new exercise plan type such as:  **Exercise\_Plan\_Type\_Description,**  which will be saved in the **Exercise\_Plan\_Type table.** The system will then capture and validate the entered details and check to see if the exercise plan type - provided by the user – doesn’t already exist on the system. If everything is done correctly the system will display a success message to the employee. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to add a new exercise plan type to the system. | | | |
| **TRIGGER:** | The employee requests to add a new exercise plan type to the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee requests to add a new exercise plan type to the system. | | **Step 2:** The system will display the add new exercise plan type screen and inform the employee to enter a new exercise plan type detail:  **Exercise\_Plan\_Type\_Description.** | |
|  | **Step 3:** The employee will enter a new **Exercise\_Plan\_Type\_Description** on the add exercise plan type page and save it on the system. | | **Step 4:** The system will capture the entered detail:  **Exercise\_Plan\_Type\_Description.** | |
|  |  | | **Step 5:** The system will validate the entered type so that it is a valid exercise plan type description. | |
|  |  | | **Step 6:** The system will check to see if the type added does not already exist in the system. | |
|  |  | | **Step 7:** The system will automatically generate a unique **Exercise\_Plan\_Type\_ID** and add it to the **Exercise\_Plan\_Type table**. | |
|  |  | | **Step 8:** The system will add the type to the **Exercise\_Plan\_Type table** and save it as an:  **Exercise\_Plan\_Type\_Description**. | |
|  |  | | **Step 9:** The system will display a success message to the employee and return the employee to the home page. | |
| **ALTERNATE COURSES:** | **Alt Step 5:** The entered **Exercise\_Plan\_Type\_Description** is invalid, display an error message.   * Return the user to step 3. | | | |
| **CONCLUSION:** | The use case concludes when the new exercise plan type has been successfully added to the system and a success message is displayed to the employee. | | | |
| **POST-CONDITION:** | A new exercise plan type has been successfully added to the **Exercise\_Plan\_Type table** on the system. | | | |
| **BUSINESS RULES** | Only an employee with the required access level can add an exercise plan type. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The Exercise Plan Type does not yet exist on the system. | | | |
| **OPEN ISSUES:** | None. | | | |

Table 8.8.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Iain Steyn**  **Janco Steenkamp (checked)** | | **Version 1.5** | | |
| **USE CASE NAME:** | **Maintain Exercise Plan Type** | | | **USE CASE TYPE** |
| **USE CASE ID:** | 8.2 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee requests to maintain an exercise plan type on the system. The employee will request to maintain an exercise plan type and the system will display the maintain exercise plan type page. The system will then prompt the user to search for an existing exercise plan type on the exercise plan type page and invoke use case **1.3 Search Exercise Plan Type** and display the output on the page. The employee will then select the type they would like to maintain by selecting either the edit or delete option. All changes will be added to the system and a success message will be displayed. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to maintain an exercise plan type. | | | |
| **TRIGGER:** | The employee selects to edit an exercise plan type from the edit exercise plan option. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee requests to maintain an exercise plan type from the maintain exercise plan option. | | **Step 2:** The system will display the maintain exercise plan type page and display all existing exercise plan types as well as an edit and remove option for each existing type. | |
|  | **Step 3:** The employee will select the edit exercise plan type option for the exercise plan type that they would like to edit. | | **Step 4:** The system will retrieve all the details for the selected **Exercise\_Plan\_Type\_ID.** | |
|  |  | | **Step 5:** The system will display the selected exercise details to edit for the user. | |
|  | **Step 6:** The employee will update the exercise plan type and save it to the system. | | **Step 7:** The system will capture the edited type as: **Exercise\_Plan\_Type\_Description.** | |
|  |  | | **Step 8:** The system will validate the type so that it is a is a valid description. | |
|  |  | | **Step 9:** The system will then update the type for the selected **Exercise\_Plan\_Type\_ID** in the **Exercise\_Plan\_Type** **table**. | |
|  |  | | **Step 10:** The system will display a success message to the employee.   * Return the employee to the home page. | |
|  | **Alt Step 3:** The user does not see the type they would like to edit or remove.   * **Invoke 1.3 Use Case Search Exercise Plan Type.** | | | |
| **ALTERNATE COURSES:** | **Alt Step 3a:** The employee selects the remove exercise plan type option for the exercise plan type that they would like to remove. | | | |
|  | **Alt Step 3b:** The system prompts the user to confirm if they would like to delete the selected exercise plan type. | | | |
|  | **Alt Step 3c:** The user confirms that they would like to delete the selected exercise plan type. | | | |
|  | **Alt Step 3cc:** The User revokes the decision to remove the selected exercise plan type and selects the option to cancel confirmation.   * Return to Step 2. | | | |
|  | **Alt Step 3d:** The system removes the selected exercise plan type and its relevant details from the **Exercise\_Plan\_Type table.**   * Continue at step 9. | | | |
|  | **Alt Step 8:** The updated exercise plan type description is not a valid value, display an error message.   * Return to step 6. | | | |
| **CONCLUSION:** | The selected exercise plan type has been successfully maintained and all changes have been updated on the system; a success message has been displayed. | | | |
| **POST-CONDITION:** | **Exercise\_Plan\_Type table** is up to date. | | | |
| **BUSINESS RULES** | Only an employee with the required access level can maintain an exercise plan type. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The Exercise Plan Type exists on the system | | | |
| **OPEN ISSUES:** | None. | | | |

Table 8.8.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Iain Steyn**  **Janco Steenkamp (checked)** | | **Version 1.1** | | |
| **USE CASE NAME:** | **Search Exercise Plan Type** | | | **USE CASE TYPE** |
| **USE CASE ID:** | 8.3 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee requests to search for an exercise plan type on the system. The user will search for an exercise plan type by entering a keyword or phrase that matches any existing:  **Exercise\_Plan\_Type\_ID,**  **Exercise\_Plan\_Type\_Description,**  in the **Exercise\_Plan\_Type table**, into the provided (by the system) search bar.The system will then verify and search for any existing exercise plan type and display an exercise plan type that matches the given search criteria. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to search an exercise plan type on the system. | | | |
| **TRIGGER:** | The employee needs to search for an exercise plan type on the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee needs to search for an exercise plan type on the system. | | **Step 2**: The system prompts the employee to enter search criteria | |
|  | **Step 3:** The employee will enter any keyword or phrase that will match any existing:  **Exercise\_Plan\_Type\_ID,**  **Exercise\_Plan\_Type\_Description,**  in the **Exercise\_Plan\_Type table** on the system. | |  | |
|  | **Step 4:** The employee will request to search. | | **Step 5:** The system will capture the entered search keyword or phrase. | |
|  |  | | **Step 6:** The system will validate the entered search criteria so that it is a valid search keyword or phrase. | |
|  |  | | **Step 7:** The system will check the:  **Exercise\_Plan\_Type\_ID,**  **Exercise\_Plan\_Type\_Description,**  details for all existing exercise plan types in the **Exercise\_Plan\_Type table** to see if any match the captured search criteria. | |
|  |  | | **Step 8:** The system returns a list of all exercise plan type (s) matching the given search criteria. | |
| **ALTERNATE COURSES:** | **Alt Step 6:** The captured keyword or phrase is invalid, display an error message.   * Return to step 2. | | | |
|  | **Alt Step 7:** No values on the system match the given search criteria, the system will display an error message.   * Return to step 2. | | | |
| **CONCLUSION:** | The system has successfully retrieved the required exercise plan type based off the given search criteria and displayed the results to the employee. | | | |
| **POST-CONDITION:** | The system successfully retrieved the exercise plan type/s from the **Exercise\_Plan\_Type table** on the system. | | | |
| **BUSINESS RULES** | Only an employee with the required access levels can maintain an exercise plan type. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The Exercise Plan Type exists on the system. | | | |
| **OPEN ISSUES:** | None. | | | |

Table 8.8.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/04/12** | | |
| **Authors: Iain Steyn**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Add new exercise type** | | | **Use Case Type:** |
| **USE CASE ID:** | 8.4 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: **◻** |
| **SOURCE:** | JessterFit | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | Owner | | | |
| **DESCRIPTION:** | The use case describes the event that an employee requests to add a new exercise type to the system. The employee will add a new exercise type on the add new exercise type page which will be displayed by the system when the employee wants to add a new exercise type. The employee will add all the required details for the new exercise type such as:  **Exercise\_Type\_Description,**  which will be saved in the **Exercise\_Type table.** The system will then capture and validate the entered details and check to see if the exercise type - provided by the user – doesn’t already exist on the system. If everything is done correctly the system will display a success message to the employee. | | | |
| **PRE-CONDITION:** | The exercise type must be approved by the owner.  The user must have logged in with the required access level | | | |
| **TRIGGER:** | The employee selects to add a new exercise type to the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1:** An employee requests to add a new exercise type onto the system. | |  | |
|  |  | | **Step 2**: The system requests the employee to enter the information required | |
|  | **Step3:** The employee enters the **Exercise\_Type\_Description** as requested by the system | |  | |
|  |  | | **Step 4**:  The system captures the data for a new exercise type entered by the employee | |
|  |  | | **Step 5**: The system validates the format of the captured detail to check if it is in the correct format as follows. | |
|  |  | | **Step 6**: The system checks if the exercise type does not already exist on the system by checking the **Exercise\_Type** table for a similar **Exercise\_Type\_Description** | |
|  |  | | **Step 7:** The system displays a confirmation screen with the information entered by the employee for them to confirm correctness of the information | |
|  | **Step 8**: The employee confirms the details displayed. | | **Step 9:** The system automatically generates a unique **Exercise\_Type\_ID** and adds it to the **Exercise\_Type table.** | |
|  |  | | **Step 10:** The system adds the **Exercise\_Type\_Description** to the **Exercise\_Type\_ID** in the **Exercise\_Type table.** | |
|  |  | | **Step 11**: The system displays a success message to the employee and returns him to Exercise type page. | |
| **ALTERNATE COURSES:** | **Alt Step 5**: The information entered fails the validation test, the system displays an error message.   * Returns to step 3 | | | |
|  | **Alt Step 6**: The exercise entered already exists, display an error message   * Return to step 3 | | | |
|  | **Alt Step 8**: The employee does not confirm the information displayed and cancels the transaction.   * Return to step 3 | | | |
| **CONCLUSION:** | The use case concludes when a new challenge has been successfully added to the system and a success message is displayed to the employee. | | | |
| **POST-CONDITION:** | A new exercise type has been successfully added to **Exercise\_Type table** on the system. | | | |
| **BUSINESS RULES** | Only an employee with the required access level can add an exercise type. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None | | | |
| **ASSUMPTIONS:** | The exercise type does not exist on the system | | | |
| **OPEN ISSUES:** | None | | | |

Table 8.8.4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/04/12** | | |
| **Authors: Iain Steyn**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Maintain exercise type** | | | **Use Case Type:** |
| **USE CASE ID:** | 8.5 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis:◻ |
| **SOURCE:** | JessterFit | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | Owner | | | |
| **DESCRIPTION:** | The use case describes the event that an employee would like to maintain an exercise type on the system. The employee will request to maintain an exercise type and the system will display the maintain exercise type page. The system will then prompt the user to search for an existing employee on the employee page and invoke use case **8.6 Search Exercise Type** and display the output on the page. The employee will then select the **Exercise Type** they would like to maintain by selecting either the edit or delete option. All changes will be updated on the system and a success message will be displayed | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to maintain an employee. | | | |
| **TRIGGER:** | The employee selects to edit an employee from the edit option. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee requests to maintain an **Exercise Type**. | | **Step 2:** The system will display the maintain exercise type page and prompt the owner to search for an existing exercise on the page. | |
|  |  | | **Step 3:** The system will invoke use case **8.6 Search Exercise Type** and display the results in a list | |
|  | **Step 4:** The employee will select the edit exercise type option for the exercise type that they would like to edit. | | **Step 5:** The system will retrieve all the details gathered from use case **8.6 Search Exercise Type** for the selected exercise type such as: **Exercise\_Type\_Description** and display it on the maintain employee page into an editable field for the employee. | |
|  | **Step 6:** The employee will update the exercise type and submit it to the system. | | **Step 7:** The system will capture the edited exercise type as: **Exercise\_Type\_Description** | |
|  |  | | **Step 8:** The system will validate the employee to ensure that the details are valid employee details. | |
|  |  | | **Step 9:** The system will then update the **Exercise\_Type\_Description** for the selected **Exercise\_Type\_ID** in the **Exercise\_Type table**. | |
|  |  | | **Step 10:** The system will display a success message to the employee.   * Return the employee to the home page. | |
| **ALTERNATE COURSES:** | **Alt Step 4a:** The user selects the remove exercise type option for the exercise type that they would like to remove. | | | |
|  | **Alt Step 4b:** The system prompts the user to confirm if they would like to delete the selected exercise type. | | | |
|  | **Alt Step 4c:** The user confirms that they would like to delete the selected exercise type. | | | |
|  | **Alt Step 4cc:** The user revokes the decision to remove the selected exercise type and selects the option to cancel confirmation.   * Return to Step 1. | | | |
|  | **Alt Step 4d:** The system removes the selected exercise type and its relevant details from the **Exercise\_Type table.**   * Continue at step 10. | | | |
|  | **Alt Step 8:** The updated exercise type details are invalid, display an error message.   * Return to step 6. | | | |
| **CONCLUSION:** | The system has successfully retrieved the required exercise type based off the given search criteria and displayed the results to the employee. | | | |
| **POST-CONDITION:** | The system successfully retrieved the exercise type/s from the **Exercise\_Type table** on the system. | | | |
| **BUSINESS RULES** | Only an employee with the required access levels can maintain an exercise plan type. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The Exercise Plan Type exists on the system. | | | |
| **OPEN ISSUES:** | None. | | | |

Table 8.8.5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/04/12** | | |
| **Authors: Iain Steyn**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Search Exercise Type** | | | **Use Case Type:** |
| **USE CASE ID:** | 8.6 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis:◻ |
| **SOURCE:** | JessterFit | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None | | | |
| **DESCRIPTION:** | The use case describes the event that an employee requests to search for an exercise type on the system. The user will search for an exercise type by entering a keyword or phrase that matches any existing:  **Exercise\_Type\_ID,**  **Exercise\_Type\_Description,**  in the **Exercise\_Type table**, into the provided (by the system) search bar.The system will then verify and search for any existing exercise type and display an exercise type that matches the given search criteria. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to search an exercise type on the system. | | | |
| **TRIGGER:** | The employee requests to search for an exercise type on the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee needs to search for an exercise type on the system. | | **Step 2**: The system prompts the employee to enter search criteria | |
|  | **Step 3:** The employee will enter any keyword or phrase that will match any existing:  **Exercise\_Type\_ID,**  **Exercise\_Type\_Description,**  in the **Exercise\_Type table** on the system. | |  | |
|  | **Step 4:** The employee will request to search. | | **Step 5:** The system will capture the entered search keyword or phrase. | |
|  |  | | **Step 6:** The system will validate the entered search criteria so that it is a valid search keyword or phrase. | |
|  |  | | **Step 7:** The system will check the:  **Exercise\_Type\_ID,**  **Exercise\_Type\_Description,**  details for all existing exercise types in the **Exercise\_Type table** to see if any match the captured search criteria. | |
|  |  | | **Step 8:** The system returns a list of all exercise type (s) matching the given search criteria. | |
| **ALTERNATE COURSES:** | **Alt Step 6:** The captured keyword or phrase is invalid, display an error message.   * Return to step 2. | | | |
|  | **Alt Step 7:** No values on the system match the given search criteria, the system will display an error message.   * Return to step 2. | | | |
| **CONCLUSION:** | The system has successfully retrieved the required exercise type based off the given search criteria and displayed the results to the employee. | | | |
| **POST-CONDITION:** | The system successfully retrieved the exercise type/s from the **Exercise\_Type table** on the system. | | | |
| **BUSINESS RULES** | Only an employee with the required access levels can maintain an exercise type. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The Exercise Type exists on the system. | | | |
| **OPEN ISSUES:** | None. | | | |

Table 8.8.6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Janco Steenkamp**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Add New sets** | | | **Use Case Type:** |
| **USE CASE ID:** | 8.7 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee would like to add a new exercise set to the system. The employee will add the amount of sets of a needed workout to an exercise on the add new set page which will be displayed by the system. The employee will add all the required details for the new set such as:  **Set\_Description,**  which will be saved in the **Set table.** The system will then capture and validate the entered details. If everything is done correctly the system will display a success message to the employee. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to add a new set to the system. | | | |
| **TRIGGER:** | The employee requests to add a new exercise set to the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee wants to add a new set to the system. | | **Step 2:** The system will display the add new set screen and inform the employee to enter a new set detail:  **Set\_Description.** | |
|  | **Step 3:** The employee will enter a new **Set\_Description** on the add set page and save it on the system. | | **Step 4:** The system will capture the entered detail:  **Set\_Description.** | |
|  |  | | **Step 5:** The system will validate the entered type so that it is a valid set description. | |
|  |  | | **Step 6:** The system will automatically generate a unique **Set\_ID** and add it to the **Set table**. | |
|  |  | | **Step 7:** The system will add the set to the **Set table** and save it as an:  **Set\_Description** | |
|  |  | | **Step 8:** The system will display a success message to the employee and return the employee to the home page.  . | |
| **ALTERNATE COURSES:** | **Alt Step 5:** The entered **Set\_Description** is invalid. The system will display an error message.   * Return the user to step 3. | | | |
| **CONCLUSION:** | The use case concludes when the new set has been successfully added to the system and a success message is displayed to the employee. | | | |
| **POST-CONDITION:** | A new set has been successfully added to the system. | | | |
| **BUSINESS RULES** | Only an employee with the required access level can add a set. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | None. | | | |
| **OPEN ISSUES:** | None. | | | |

Table 8.8.7

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **JessterFIT** | | | **Date: 2019/03/30** | | |
| **Authors(s): Janco Steenkamp** | | | **Version 1.1** | | |
|  | **USE CASE NAME:** | **Maintain Sets** | | | **USE CASE TYPE** |
|  | **USE CASE ID:** | 8.8 | | | Business Requirements: x |
|  | **PRIORITY:** | High | | | System Analysis: ◻ |
|  | **SOURCE:** | JessterFIT | | | System Design: ◻ |
|  | **PRIMARY BUSINESS ACTOR** | Employee | | | |
|  | **PRIMARY SYSTEM ACTOR** | None. | | | |
|  | **OTHER PARTICIPATING ACTORS:** | None. | | | |
|  | **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
|  | **DESCRIPTION:** | The Use Case describes the event that an employee would like to maintain a set on the system. The employee will request to maintain a set and the system will display the maintain set page. The system will then prompt the user to search for an existing exercise on the set page and invoke use case **8.11 Search Exercise** and display the output on the page. The employee will then select the set they would like to maintain by selecting either the edit or delete option. All changes will be saved on the system and a success message will be displayed. | | | |
|  | **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to maintain a set. | | | |
|  | **TRIGGER:** | The employee requests to edit set from the edit set option. | | | |
|  | **TYPICAL COURSE** | Actor Action | | System Response | |
|  | **OF EVENTS:** | **Step 1**: The employee requests to maintain a set from the maintain set option. | | **Step 2:** The system will display the maintain set page and prompt the employee to search for an existing exercise on the page. | |
|  |  |  | | **Step 3:** The system will invoke use case **8.11 Search Exercise** and display all the results to the user in a list on the maintain set page as well as an edit and remove option for each set in the list. | |
|  |  | **Step 4:** The employee will select the edit set option for the set that they would like to edit. | | **Step 5:** The system will retrieve all the details gathered from use case **8.11 Search Exercise** for the selected set such as:  **Set\_Description**  and display it on the maintain set page into an editable field for the type. | |
|  |  | **Step 6:** The employee will update the set and save it to the system. | | **Step 7:** The system will capture the edited type as: **Set\_Description.** | |
|  |  |  | | **Step 8:** The system will validate the set so that it is a is a valid description. | |
|  |  |  | | **Step 9:** The system will then update the **Set\_Description** for the selected **Set\_ID** in the **Set** **table**. | |
|  |  |  | | **Step 10:** The system will display a success message to the employee.   * Return the employee to the home page. | |
|  | **ALTERNATE COURSES:** | **Alt Step 4a:** The employee selects the remove option for the set that they would like to remove. | | | |
|  |  | **Alt Step 4b:** The system prompts the user to confirm if they would like to delete the selected set. | | | |
|  |  | **Alt Step 4c:** The user confirms that they would like to delete the selected set. | | | |
|  |  | **Alt Step 4cc:** The employee revokes the decision to remove the selected set and selects the option to cancel confirmation.   * Return to Alt Step 4b. | | | |
|  |  | **Alt Step 4d:** The system removes the selected set and its relevant details from the **Set table.**   * Continue at step 10. | | | |
|  |  | **Alt Step 8:** The updated set description is not a valid value.   * Return to step 6. | | | |
|  | **CONCLUSION:** | The selected set has been successfully been maintained, all changes have been updated to the system and a success message has been displayed. | | | |
|  | **POST-CONDITION:** | Set details have been successfully updated to the **Set tables**. | | | |
|  | **BUSINESS RULES** | Only an employee with the required access level can maintain a set. | | | |
|  | **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
|  | **ASSUMPTIONS:** | The Set exists on the system | | | |
|  | **OPEN ISSUES:** | None. | | | |

Table 8.8.8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/04/12** | | |
| **Authors: Doreen Maloka**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Add New Exercise** | | | **Use Case Type:** |
| **USE CASE ID:** | 8.9 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None | | | |
| **DESCRIPTION:** | The use case describes the event that an employee would like to add a new product type to the system. The employee will add a new product type on the add new product type page which will be displayed by the system when the employee wants to add a new product type. The employee will add all the required details for the new product type such as: **Exercise\_Name, Exercise\_Image, Exercise\_Description, Exercise\_Type\_ID**  which will be saved in the **Exercise table.** If everything is done correctly the system will display a success message to the employee. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to add a new exercise to the system. | | | |
| **TRIGGER:** | The employee requests to add a new exercise to the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee requests to add a new exercise to the system | | **Step 2:** The system will display the add new exercise screen and inform the employee to enter a new exercise detail:  **Exercise\_Name,**  **Exercise\_Image, Exercise\_Description,**  **Exercise\_Type\_ID** | |
|  | **Step 3:** The employee will enter the add exercise details such as:  **Exercise\_Name,**  **Exercise\_Image, Exercise\_Description, ,**  **Exercise\_Type\_ID**  on the system. | | **Step 4:** The system will capture the entered details | |
|  |  | | **Step 5:** The system will validate the entered **Exercise\_Name, Exercise\_Image, Exercise\_Description** so that it is valid. | |
|  |  | | **Step 6:**  The system will validate the **Exercise\_Type\_ID**  from the database so that it is valid. | |
|  |  | | **Step 7:** The system will automatically generate a unique **Exercise\_ID** and add it to the **Exercise table** | |
|  |  | | **Step 8:** The system will save details to the database. | |
|  |  | | **Step 9:** The system will display a success message to the employee and return the employee to the home page. | |
|  |  | |  | |
| **ALTERNATE COURSES:** | **Alt Step 5:** The entered **Exercise\_Name, Exercise\_Image, Exercise\_Description** is not valid. The system will display an error message.   * Return the user to step 3. | | | |
|  | **Alt Step 6:** The entered **Exercise\_Type\_ID** is invalid. The system will display an error message.   * Return the user to step 3 | | | |
| **CONCLUSION:** | The use case concludes when the new exercise has been successfully added to the system and a success message is displayed for the employee. | | | |
| **POST-CONDITION:** | A new exercise has been successfully added to the system. | | | |
| **BUSINESS RULES** | Only a manager or owner can edit an exercise. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None | | | |
| **ASSUMPTIONS:** | The exercise does not exist on the system | | | |
| **OPEN ISSUES:** | None | | | |

Table 8.8.9

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/04/12** | | |
| **Authors: Doreen Maloka** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Maintain exercise** | | | **Use Case Type:** |
| **USE CASE ID:** | 8.10 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: 🞎 |
| **SOURCE:** | JessterFIT | | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Manager | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None | | | |
| **DESCRIPTION:** | The use case describes the process where a manager wants to update or delete an exercise from the system. The system invokes use case 2.3 search exercise and retrieves the information from the database. The manager then updates the relevant information and saves it. The new information is updated on the system and displays a success message. The manager could alternatively delete the information from the system. The information will then be removed from the database and a success message will be displayed | | | |
| **PRE-CONDITION:** | The Manager must be logged into the system and have the right access levels and functionality to be able to add a new exercise. There are exercises saved on the system. | | | |
| **TRIGGER:** | The manager wants to update or delete exercise information from the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The manager selects to update exercise information on the system | | **Step 2:** The system displays a page where the employee can add search criteria for an exercise. | |
|  | **Step 2:** The manager enters criteria for exercise they want to update | |  | |
|  |  | | **Step 3:** The system invokes **use case 1.9 search exercise** and displays information retrieved | |
|  | **Step 4:** The manager enters the information that requires changing and save them on the system. | |  | |
|  |  | | **Step 5:** The system will validate the entered exercise details so that they are valid details. | |
|  |  | | **Step 6:** The entered details are valid. | |
|  |  | | **Step 7:** The system will save all the details to the **Exercise table** such as: | |
|  |  | | **Step 8:** The system will display a success message to the manager. | |
|  |  | |  | |
| **ALTERNATE COURSES:** | **Alt Step 1:** The manager selects to remove the exercise details. | | | |
|  | **Alt step 3:** The system invokes use case Add exercise | | | |
|  | **Alt Step 6:** The entered details are not valid   * Return the user to step 3. | | | |
| **CONCLUSION:** | The use case ends when exercise details are updated on the system or when they are successfully removed from the system. | | | |
| **POST-CONDITION:** | Updated exercise details are saved on the system. | | | |
| **BUSINESS RULES** | Only a manager can update exercise details. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None | | | |
| **ASSUMPTIONS:** | None | | | |
| **OPEN ISSUES:** | None | | | |

Table 8.8.10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Joshua Eales**  **Janco Steenkamp (checked)** | | **Version 1.1** | | |
| **USE CASE NAME:** | **Search Exercise** | | | **USE CASE TYPE** |
| **USE CASE ID:** | 8.11 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis:◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee needs to search for an exercise on the system. The user will search for an exercise by entering a keyword or phrase that matches any existing:  **Exercise\_ID,**  **Exercise\_Name,**  in the **Exercise table.** The system will then verify and search for any existing exercise and display an exercise that matches the given search criteria. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to search an exercise on the system. | | | |
| **TRIGGER:** | The employee needs to search for an exercise on the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee needs to search for an exercise on the system. | | **Step 2**: The system displays the search criteria and prompts the employee to enter a keyword or phrase | |
|  | **Step 3:** The employee will enter any keyword or phrase that will match any existing:  **ExerciseID,**  **Exercise\_Name**  in the **Exercise table** on the system. | |  | |
|  | **Step 4:** The employee will request to search. | | **Step 5:** The system will capture the entered search keyword or phrase. | |
|  |  | | **Step 6:** The system will validate the entered search criteria so that it is a valid search keyword or phrase. | |
|  |  | | **Step 7:** The system will check the:  **Exercise\_ID,**  **Exercise\_Name,**  details for all existing exercise plans in the **Exercise table** to see if any match the captured search criteria. | |
|  |  | | **Step 8:** The system returns a list of all exercise(s) matching the given search criteria. | |
| **ALTERNATE COURSES:** | **Alt Step 6:** The captured keyword or phrase is invalid, display an error message.   * Return to step 2. | | | |
|  | **Alt Step 7:** No values on the system match the given search criteria, display an error message.   * Return to step 2. | | | |
| **CONCLUSION:** | The system has successfully searched for the required exercise based off the given search criteria and displayed the results to the employee. | | | |
| **POST-CONDITION:** | The system successfully finds the exercise(s) on the system. | | | |
| **BUSINESS RULES** | Only a manager or an owner can maintain an exercise. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The exercise exists on the system. | | | |
| **OPEN ISSUES:** | None. | | | |

Table 8.8.11

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Joshua Eales**  Janco Steenkamp (checked) | | **Version 1.3** | | |
| **USE CASE NAME:** | **Add New Exercise Plan** | | | **Use Case Type:** |
| **USE CASE ID:** | 8.12 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis:◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee would like to add a new exercise plan to the system. The employee will add a new exercise plan on the add new exercise plan page which will be displayed by the system when the employee wants to add a new exercise plan. The employee will add all the required details for the new exercise plan type such as: **ExercisePlan\_Name**, **ExercisePlan\_Description** and **ExercisePlan\_Price** which will be saved in the **Exercise\_Plan** table**.**  The system will then capture and validate the entered details and check to see if the exercise plan - provided by the user – doesn’t already exist on the system. If everything is done correctly the system will display a success message to the employee. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be allowed to add a new exercise plan to the system. | | | |
| **TRIGGER:** | The employee requests to add a new exercise plan to the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee requests to add a new exercise plan to the system. | | **Step 2:** The system will display the add new exercise plan screen and inform the employee to enter a new exercise plans detail’s: **ExercisePlan\_Name**, **ExercisePlan\_Description** and **ExercisePlan\_Price.** | |
|  | **Step 3:** The employee will enter a new **ExercisePlan\_Name**, **ExercisePlan\_Description** and **ExercisePlan\_Price** on the add exercise plan page and save it on the system. | | **Step 4:** The system will capture the entered exercise plan details. | |
|  |  | | **Step 5:** The system will validate the entered plan to ensure all details are valid exercise plan details. | |
|  |  | | **Step 6:** The system will check to see if the plan added does not already exist in the system. | |
|  |  | | **Step 7:** The system will automatically generate a unique **Exercise\_Plan\_ID** and add it to the **Exercise\_Plan table**. | |
|  |  | | **Step 8:** The system will add the plan to the **Exercise\_Plan table** and save it as a new exercise plan. | |
|  |  | | **Step 9:** The system will display a success message to the employee and return the employee to exercise plan page. | |
| **ALTERNATE COURSES:** | **Alt Step 5:** The entered **ExercisePlan\_Name**, **ExercisePlan\_Description** and **ExercisePlan\_Price** are invalid. The system will display an error message.   * Return the user to step 3. | | | |
| **CONCLUSION:** | The use case ends when the new exercise plan has been successfully added to the system and a success message is displayed for the employee. | | | |
| **POST-CONDITION:** | A new exercise plan has been successfully added to the system. | | | |
| **BUSINESS RULES** | Only employees with approval from the owner can add a new exercise plan to the system. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The exercise plan does not exist on the system | | | |
| **OPEN ISSUES:** | None. | | | |

Table 8.8.12

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Joshua Eales**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Maintain Exercise Plan** | | | **USE CASE TYPE** |
| **USE CASE ID:** | 8.13 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis:◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee would like to maintain an exercise plan on the system. The employee will request to maintain an exercise plan and the system will display the maintain exercise plan page. The system will then prompt the user to search for an existing exercise by invoking use case **8.14 Search Exercise Plan** and display the output on the page. The employee will then select the plan they would like to maintain by selecting either the edit or delete option. All changes will be saved on the system and a success message will be displayed. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to maintain an exercise plan. | | | |
| **TRIGGER:** | The employee selects to edit an exercise plan from the edit exercise plan option. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee requests to maintain an exercise plan from the maintain exercise plan option. | | **Step 2:** The system will display the maintain exercise plan page and prompt the employee to search for an existing exercise plan on the page. | |
|  |  | | **Step 3:** The system will invoke use case **8.14 Search Exercise Plan** and display all the results to the user in a list on the maintain exercise plan page as well as an edit and remove option for each type in the list. | |
|  | **Step 4:** The employee will select the edit exercise plan option for the exercise plan that they would like to edit. | | **Step 5:** The system will retrieve all the details gathered from use case **8.14 Search Exercise Plan** for the selected type such as: **ExercisePlan\_Name**, **ExercisePlan\_Description** and **ExercisePlan\_Price,** and display it on the maintain exercise plan page into an editable field for the plan. | |
|  | **Step 6:** The employee will update the exercise plan and save it to the system. | | **Step 7:** The system will capture the edited plan as: **ExercisePlan\_Name**, **ExercisePlan\_Description** and **ExercisePlan\_Price.** | |
|  |  | | **Step 8:** The system will validate the plan to ensure all entered details are valid. | |
|  |  | | **Step 9:** The system will then update the type for the selected **Exercise\_Plan\_ID** in the **Exercise\_Plan** **table**. | |
|  |  | | **Step 10:** The system will display a success message to the employee.   * Return the employee to the home page. | |
| **ALTERNATE COURSES:** | **Alt Step 4a:** The employee selects the remove exercise plan option for the exercise plan that they would like to remove. | | | |
|  | **Alt Step 4b:** The system prompts the user to confirm if they would like to delete the selected exercise plan. | | | |
|  | **Alt Step 4c:** The user confirms that they would like to delete the selected exercise plan. | | | |
|  | **Alt Step 4cc:** The User revokes the decision to remove the selected exercise plan and selects the option to cancel confirmation.   * Return to Alt Step 4b. | | | |
|  | **Alt Step 4d:** The system removes the selected exercise plan and its relevant details from the **Exercise\_Plan table.**   * Continue at step 10. | | | |
|  | **Alt Step 8:** The updated exercise plan description is not a valid value.   * Return to step 6. | | | |
| **CONCLUSION:** | The selected exercise plan has been successfully maintained and all changes have been saved on the system and a success message has been displayed. | | | |
| **POST-CONDITION:** | Exercise plan details have been successfully maintained. | | | |
| **BUSINESS RULES** | Only a manager or an owner can maintain an exercise plan. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The exercise plan exists on the system | | | |
| **OPEN ISSUES:** | None. | | | |

Table 8.8.13

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Joshua Eales**  **Janco Steenkamp (checked)** | | **Version 1.1** | | |
| **USE CASE NAME:** | **Search Exercise Plan** | | | **USE CASE TYPE** |
| **USE CASE ID:** | 8.14 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis:◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee needs to search for an exercise plan on the system. The user will search for an exercise plan by entering a keyword or phrase that matches any existing:  **Exercise\_Plan\_ID,**  **Exercise\_Plan\_Name,**  in the **Exercise\_Plan table.** The system will then verify and search for any existing exercise plan and display an exercise plan that matches the given search criteria. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to search an exercise plan on the system. | | | |
| **TRIGGER:** | The employee needs to search for an exercise plan on the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee needs to search for an exercise plan on the system. | | **Step 2**: The system displays the search criteria and prompts the employee to enter a keyword or phrase | |
|  | **Step 3:** The employee will enter any keyword or phrase that will match any existing:  **Exercise\_Plan\_ID,**  **Exercise\_Plan\_Name**  in the **Exercise\_Plan table** on the system. | |  | |
|  | **Step 4:** The employee will request to search. | | **Step 5:** The system will capture the entered search keyword or phrase. | |
|  |  | | **Step 6:** The system will validate the entered search criteria so that it is a valid search keyword or phrase. | |
|  |  | | **Step 7:** The system will check the:  **Exercise\_Plan\_ID,**  **Exercise\_Plan\_Name,**  details for all existing exercise plans in the **Exercise\_Plan table** to see if any match the captured search criteria. | |
|  |  | | **Step 8:** The system returns a list of all exercise plan(s) matching the given search criteria. | |
| **ALTERNATE COURSES:** | **Alt Step 6:** The captured keyword or phrase is invalid, display an error message.   * Return to step 2. | | | |
|  | **Alt Step 7:** No values on the system match the given search criteria, display an error message.   * Return to step 2. | | | |
| **CONCLUSION:** | The system has successfully searched for the required exercise plan based off the given search criteria and displayed the results to the employee. | | | |
| **POST-CONDITION:** | The system successfully finds the exercise plan(s) on the system. | | | |
| **BUSINESS RULES** | Only a manager or an owner can maintain an exercise plan. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The exercise plan exists on the system. | | | |
| **OPEN ISSUES:** | None. | | | |

Table 8.8.14

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/04/12** | | |
| **Authors: Doreen Maloka**  Janco Steenkamp (checked) | | **Version 1.3** | | |
| **USE CASE NAME:** | **Add Exercise Body Type** | | | **Use Case Type:** |
| **USE CASE ID:** | 8.15 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS** | None | | | |
| **DESCRIPTION:** | The use case describes the event that an employee would like to add a new exercise body type to the system. The employee will add a new exercise body type on the add new exercise body type page which will be displayed by the system when the employee wants to add a new exercise body type. The employee will add all the required details for the new exercise body type such as:  **Exercise\_Body\_Type\_Description,**  which will be saved in the **Exercise\_Body\_Type table.** If everything is done correctly the system will display a success message to the employee. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to add a new exercise body type to the system. | | | |
| **TRIGGER:** | The employee requests to add a new exercise body type to the system | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee wants to add a new exercise body type to the system. | | **Step 2:** The system will display the add new exercise body type screen and inform the employee to enter a new exercise body type detail:  **Exercise\_Body\_Type\_Description.** | |
|  | **Step 3:** The employee will enter a new **Exercise\_Body\_Type\_Description** on the add exercise body type page and save it on the system. | | **Step 4:** The system will capture the entered detail:  **Exercise\_Body\_Type\_Description.** | |
|  |  | | **Step 5:** The system will validate the entered type so that it is a valid exercise body type description. | |
|  |  | | **Step 6:** The system checks for duplicates in the database. | |
|  |  | | **Step 7:** The system will automatically generate a unique **Exercise\_Body\_Type\_ID** and add it to the **Exercise\_Body\_Type table**. | |
|  |  | | **Step 8:** The system will add the type to the **Exercise\_Body\_Type table** and save it as an:  **Exercise\_Body\_Type\_Description**. | |
|  |  | | **Step9:** The system will display a success message to the employee and return the employee to the home page. | |
| **ALTERNATE COURSES:** | **Alt Step 5:** The entered **Exercise\_Body\_Type\_Description** is invalid. The system will display an error message.   * Return the user to step 3. | | | |
|  | **Alt Step 6:** The entered **Exercise\_Body\_Type\_Description** is a duplicate. The system will display an error message.   * Return the user to step 3. | | | |
| **CONCLUSION:** | The use case ends when the new exercise body type has been successfully added to the system and a success message is displayed for the employee. | | | |
| **POST-CONDITION:** | A new exercise body type has been successfully added to the system. | | | |
| **BUSINESS RULES** | Only an employee with the required access level can edit an exercise body type | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None | | | |
| **ASSUMPTIONS:** | The exercise body type does not exist on the system | | | |
| **OPEN ISSUES:** | None | | | |

Table 8.8.15

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Joshua Eales**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Maintain Exercise Body type** | | | **USE CASE TYPE** |
| **USE CASE ID:** | 8.16 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis:◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee would like to maintain an exercise body type on the system. The employee will request to maintain an exercise body type and the system will display the maintain exercise body type page. The system will then prompt the user to search for an existing exercise body type by invoking use case **8.17 Search Exercise body type** and display the output on the page. The employee will then select the body type they would like to maintain by selecting either the edit or delete option. All changes will be saved on the system and a success message will be displayed. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to maintain an exercise body type. | | | |
| **TRIGGER:** | The employee selects to edit an exercise body type from the edit exercise body type option. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee requests to maintain an exercise body type from the maintain exercise body option. | | **Step 2:** The system will display the maintain exercise body type page and prompt the employee to search for an existing exercise body type on the page. | |
|  |  | | **Step 3:** The system will invoke use case **8.17 Search Exercise Body Type** and display all the results to the user in a list on the maintain exercise body type page as well as an edit and remove option for each type in the list. | |
|  | **Step 4:** The employee will select the edit exercise body type option for the exercise plan that they would like to edit. | | **Step 5:** The system will retrieve all the details gathered from use case **8.17 Search Exercise Body Type** for the selected type such as: **Exercise\_Body\_Type\_Description** and display it on the maintain exercise body type page into an editable field for the type. | |
|  | **Step 6:** The employee will update the exercise plan and save it to the system. | | **Step 7:** The system will capture the edited plan as: **Exercise\_Body\_Type\_Description.** | |
|  |  | | **Step 8:** The system will validate the plan to ensure all entered details are valid. | |
|  |  | | **Step 9:** The system will then update the type for the selected **Exercise\_Body\_Type\_ID** in the **Exercise Body Type** **table**. | |
|  |  | | **Step 10:** The system will display a success message to the employee.   * Return the employee to the home page. | |
| **ALTERNATE COURSES:** | **Alt Step 4a:** The employee selects the remove exercise body type option for the exercise body type that they would like to remove. | | | |
|  | **Alt Step 4b:** The system prompts the user to confirm if they would like to delete the selected exercise body type. | | | |
|  | **Alt Step 4c:** The user confirms that they would like to delete the selected exercise body type. | | | |
|  | **Alt Step 4cc:** The User revokes the decision to remove the selected exercise body type and selects the option to cancel confirmation.   * Return to Alt Step 4b. | | | |
|  | **Alt Step 4d:** The system removes the selected exercise body type and its relevant details from the **Exercise Body Type table.**   * Continue at step 10. | | | |
|  | **Alt Step 8:** The updated exercise body type description is not a valid value.   * Return to step 6. | | | |
| **CONCLUSION:** | The selected exercise body type has been successfully maintained and all changes have been saved on the system and a success message has been displayed. | | | |
| **POST-CONDITION:** | Exercise body type details have been successfully maintained. | | | |
| **BUSINESS RULES** | Only a manager or an owner can maintain an exercise body type. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The exercise body type exists on the system | | | |
| **OPEN ISSUES:** | None. | | | |

Table 8.8.16

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/04/12** | | |
| **Authors: Iain Steyn**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Search Exercise body type** | | | **Use Case Type:** |
| **USE CASE ID:** | 8.17 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis:◻ |
| **SOURCE:** | JessterFit | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None | | | |
| **DESCRIPTION:** | The use case describes the event that an employee requests to search for an exercise body type on the system. The user will search for an exercise body type by entering a keyword or phrase that matches any existing:  **Exercise\_Body\_Type\_ID,**  **Exercise\_Body\_Type\_Description,**  in the **Exercise\_Body\_Type table**, into the provided (by the system) search bar.The system will then verify and search for any existing exercise body type and display an exercise body type that matches the given search criteria. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to search an exercise body type on the system. | | | |
| **TRIGGER:** | The employee requests to search for an exercise body type on the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee needs to search for an exercise body type on the system. | | **Step 2**: The system prompts the employee to enter search criteria | |
|  | **Step 3:** The employee will enter any keyword or phrase that will match any existing:  **Exercise\_Body\_Type\_ID,**  **Exercise\_Body\_Type\_Description,**  in the **Exercise\_Body\_Type table** on the system. | |  | |
|  | **Step 4:** The employee will request to search. | | **Step 5:** The system will capture the entered search keyword or phrase. | |
|  |  | | **Step 6:** The system will validate the entered search criteria so that it is a valid search keyword or phrase. | |
|  |  | | **Step 7:** The system will check the:  **Exercise\_Body\_Type\_ID,**  **Exercise\_Body\_Type\_Description,**  details for all existing exercise types in the **Exercise\_Body\_Type table** to see if any match the captured search criteria. | |
|  |  | | **Step 8:** The system returns a list of all exercise body type (s) matching the given search criteria. | |
| **ALTERNATE COURSES:** | **Alt Step 6:** The captured keyword or phrase is invalid, display an error message.   * Return to step 2. | | | |
|  | **Alt Step 7:** No values on the system match the given search criteria, the system will display an error message.   * Return to step 2. | | | |
| **CONCLUSION:** | The system has successfully retrieved the required exercise body type based off the given search criteria and displayed the results to the employee. | | | |
| **POST-CONDITION:** | The system successfully retrieved the exercise type/s from the **Exercise\_Body\_Type table** on the system. | | | |
| **BUSINESS RULES** | Only an employee with the required access levels can maintain an exercise body type. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The Exercise body type exists on the system. | | | |
| **OPEN ISSUES:** | None. | | | |

Table 8.8.17

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/04/12** | | |
| **Authors: Doreen Maloka** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Add Workout** | | | **Use Case Type:** |
| **USE CASE ID:** | 8.18 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: 🞎 |
| **SOURCE:** | JessterFIT | | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Manager | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee would like to add a new workout to the system. The employee will add a new workout on the add new workout page which will be displayed by the system when the employee wants to add a new workout. The employee will add all the required details for the new workout such as:  **Workout\_Description,**  which will be saved in the **Workout table.** If everything is done correctly the system will display a success message to the employee. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to add a new workout to the system. | | | |
| **TRIGGER:** | The employee wants to add a new workout to the system | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee wants to add a new workout to the system. | | **Step 2:** The system will display the add new workout page and inform the employee to enter a new workout detail:  **Workout\_Description.** | |
|  | **Step 3:** The employee will enter a new **Workout\_Description** on the add workout page and save it on the system. | | **Step 4:** The system will capture the entered detail:  **Workout\_Description.** | |
|  |  | | **Step 5:** The system will validate the entered type so that it is a valid workout description. | |
|  |  | | **Step 6:** The system will automatically generate a unique **Workout \_ID** and add it to the **Workout table**. | |
|  |  | | **Step 7:** The system will add the type to the **Workout table** and save it as an:  **Workout\_Description**. | |
|  |  | | **Step 8:** The system will display a success message to the employee and return the employee to the home page. | |
|  |  | |  | |
| **ALTERNATE COURSES:** | **Alt Step 5:** The entered **Workout\_Description** is invalid. The system will display an error message.  Return the user to step 3. | | | |
| **CONCLUSION:** | The use case ends when the new workout has been successfully added to the system and a success message is displayed for the employee. | | | |
| **POST-CONDITION:** | A new workout has been successfully added to the system. | | | |
| **BUSINESS RULES** | Only a manager or owner can edit a workout | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None | | | |
| **ASSUMPTIONS:** | None | | | |
| **OPEN ISSUES:** | None | | | |

Table 8.8.18

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Joshua Eales**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Maintain Workout** | | | **USE CASE TYPE** |
| **USE CASE ID:** | 8.19 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis:◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee would like to maintain a workout on the system. The employee will request to maintain a workout and the system will display the maintain workout page. The system will then prompt the user to search for an existing workout by invoking use case **8.20 Search Workout** and display the output on the page. The employee will then select the workout they would like to maintain by selecting either the edit or delete option. All changes will be saved on the system and a success message will be displayed. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to maintain a workout. | | | |
| **TRIGGER:** | The employee selects to edit a workout from the edit workout option. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee requests to maintain a workout from the maintain workout option. | | **Step 2:** The system will display the maintain workout page and prompt the employee to search for an existing workout on the page. | |
|  |  | | **Step 3:** The system will invoke use case **8.20 Search Workout** and display all the results to the user in a list on the maintain workout page as well as an edit and remove option for each workout in the list. | |
|  | **Step 4:** The employee will select the edit workout option for the workout that they would like to edit. | | **Step 5:** The system will retrieve all the details gathered from use case **8.20 Search Workout** for the selected workout such as: **Workout\_Description** and display it on the maintain workout page into an editable field for the workout. | |
|  | **Step 6:** The employee will update the workout and save it to the system. | | **Step 7:** The system will capture the edited plan as: **Exercise\_Body\_Type\_Description.** | |
|  |  | | **Step 8:** The system will validate the workout to ensure all entered details are valid. | |
|  |  | | **Step 9:** The system will then update the workout for the selected **Workout\_ID** in the **Workout** **table**. | |
|  |  | | **Step 10:** The system will display a success message to the employee.   * Return the employee to the home page. | |
| **ALTERNATE COURSES:** | **Alt Step 4a:** The employee selects the remove exercise workout option for the exercise body type that they would like to remove. | | | |
|  | **Alt Step 4b:** The system prompts the user to confirm if they would like to delete the selected work out. | | | |
|  | **Alt Step 4c:** The user confirms that they would like to delete the selected workout. | | | |
|  | **Alt Step 4cc:** The User revokes the decision to remove the selected work out and selects the option to cancel confirmation.   * Return to Alt Step 4b. | | | |
|  | **Alt Step 4d:** The system removes the selected workout and its relevant details from the **Workout table.**   * Continue at step 10. | | | |
|  | **Alt Step 8:** The updated exercise body type description is not a valid value.   * Return to step 6. | | | |
| **CONCLUSION:** | The selected workout has been successfully maintained and all changes have been saved on the system and a success message has been displayed. | | | |
| **POST-CONDITION:** | Workout details have been successfully maintained. | | | |
| **BUSINESS RULES** | Only a manager or an owner can maintain a workout. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The workout exists on the system | | | |
| **OPEN ISSUES:** | None. | | | |

Table 8.8.19

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/04/12** | | |
| **Authors: Doreen Maloka** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Search Workout** | | | **Use Case Type:** |
| **USE CASE ID:** | 8.20 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: 🞎 |
| **SOURCE:** | JessterFIT | | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Manager | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee needs to search for a workout on the system. The user will search for a workout by entering a keyword or phrase that matches any existing:  **Workout \_ID,**  **Workout\_Description,**  in the **Workout table.** The system will then verify and search for any existing workout and display a workout that matches the given search criteria. | | | |
| **PRE-CONDITION:** | The Employee must be logged into the system and have the right access levels and functionality to be able to add a new workout. There are workout saved on the system. | | | |
| **TRIGGER:** | The employee wants to search for a workoutinformation from the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee selects to search for a workout from the system | | **Step 2:** The system will prompt the employee to enter any keywords that match any existing:  **Workout\_ID,**  **Workout\_Description.** | |
|  | **Step 3:** The employee will enter any keyword or phrase that will match any existing:  **Workout\_ID,**  **Workout\_Description,**  on the system. | |  | |
|  | **Step 4:** The employee will request to search | | **Step 5:** The system will capture the entered search keyword or phrase | |
|  |  | | **Step 6:** The system will validate the entered search criteria so that it is a valid search keyword or phrase. | |
|  |  | | **Step 7:** The captured keyword or phrase is valid, and the system will check the:  **Workout\_ID,**  **Workout\_Description,**  details for all existing workout in the **Workout table** to see if any match the captured search criteria. | |
|  |  | | **Step 8:** The system returns a list of all workout(s) matching the given search criteria. | |
| **ALTERNATE COURSES:** | **Alt Step 6:** The captured keyword or phrase is invalid.  Return to step 4. | | | |
|  | **Alt Step 8:** No values on the system match the given search criteria. The system will display an error message.  Return to step 4. | | | |
| **CONCLUSION:** | The system has successfully searched for the required workout based off of the given search criteria and displayed the results to the employee. | | | |
| **POST-CONDITION:** | The system successfully finds the workout on the system. | | | |
| **BUSINESS RULES** |  | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None | | | |
| **ASSUMPTIONS:** | None | | | |
| **OPEN ISSUES:** | None | | | |

Table 8.8.20

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/04/12** | | |
| **Authors: Doreen Maloka** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Add exercise image** | | | **Use Case Type:** |
| **USE CASE ID:** | 8.21 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: 🞎 |
| **SOURCE:** | JessterFIT | | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Manager | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee would like to add a new exercise image to the system. The employee will add a new exercise image on the add new exercise image page which will be displayed by the system when the employee wants to add an exercise image. The employee will add all the required details for the new workout such as:  **Image\_URL,**  which will be saved in the **Exercise\_Image\_Table.** If everything is done correctly the system will display a success message to the employee. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to add a new exercise image to the system. | | | |
| **TRIGGER:** | The employee wants to add a new exercise image to the system | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee wants to add a new exercise image to the system. | | **Step 2:** The system will display the add new exercise image page and inform the employee to enter a new workout detail:  Exercise\_Image\_URL | |
|  | **Step 3:** The employee will enter a new Exercise\_Image\_URL on the add exercise page and save it on the system. | | **Step 4:** The system will capture the entered detail:  Exercise\_Image\_URL | |
|  |  | | **Step 5:** The system will validate the entered exercise image so that it is a valid exercise image. | |
|  |  | | **Step 6:** The system will automatically generate a unique Exercise\_Image\_ID and add it to the **Exercise image table**. | |
|  |  | | **Step 7:** The system will add the type to the **Exercise image table** and save it as an:  Exercise\_Image\_URL | |
|  |  | | **Step 8:** The system will display a success message to the employee and return the employee to the home page. | |
|  |  | |  | |
| **ALTERNATE COURSES:** | **Alt Step 5:** The entered Exercise\_Image\_URL is invalid. The system will display an error message.  Return the user to step 3. | | | |
| **CONCLUSION:** | The use case ends when the new exercise image has been successfully added to the system and a success message is displayed for the employee. | | | |
| **POST-CONDITION:** | A new exercise image has been successfully added to the system. | | | |
| **BUSINESS RULES** | Only a manager or owner can edit a exercise image | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None | | | |
| **ASSUMPTIONS:** | None | | | |
| **OPEN ISSUES:** | None | | | |

Table 8.8.21

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Joshua Eales**  **Janco Steenkamp (checked)** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Remove Exercise Image** | | | **USE CASE TYPE** |
| **USE CASE ID:** | 8.22 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis:◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee would like to remove an exercise image on the system. The employee will request to remove an exercise image and the system will display the remove exercise image page. The system will then prompt the user to search for an existing workout by invoking use case **8.23 Search Exercise Image** and display the output on the page. The employee will then select the workout they would like to maintain by selecting either the edit or delete option. All changes will be saved on the system and a success message will be displayed. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to maintain a workout. | | | |
| **TRIGGER:** | The employee selects to edit a workout from the edit workout option. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee requests to remove an exercise image from the maintain workout option. | | **Step 2:** The system will display the remove exercise image page and prompt the employee to search for an existing exercise image on the page. | |
|  |  | | **Step 3:** The system will invoke use case **8.23 Search Exercise Image** and display all the results to the user in a list on the remove exercise image page. | |
|  | **Step 4:** The employee will select to remove the exercise image option for the exercise image that they would like to remove. | | **Step 5:** The system will retrieve all the details gathered from use case **8.23 Search Exercise image** for the selected exercise image such as: **Exercise\_Image\_URL** and display it on the remove exercise image page into an editable field for the workout. | |
|  | **Step 6:** The employee will select to remove the workout and save changes on the system. | | **Step 7:** The system will then remove the exercise image for the selected **Exercise\_Image\_ID** in the **Exercise Image** **table**. | |
|  |  | | **Step 8:** The system will display a success message to the employee.  Return the employee to the home page. | |
| **CONCLUSION:** | The selected exercise image has been successfully remmoved and all changes have been saved on the system and a success message has been displayed. | | | |
| **POST-CONDITION:** | Exercise image details have been successfully removed. | | | |
| **BUSINESS RULES** | Only a manager or an owner can remove an exercise image. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The exercise exists on the system | | | |
| **OPEN ISSUES:** | None. | | | |

Table 8.8.22

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/04/12** | | |
| **Authors: Doreen Maloka** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Search exercise image** | | | **Use Case Type:** |
| **USE CASE ID:** | 8.23 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: 🞎 |
| **SOURCE:** | JessterFIT | | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None | | | |
| **OTHER PARTICIPATING ACTORS:** | None | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee needs to search for an exercise image on the system. The user will search for a exercise image by entering a keyword or phrase that matches any existing:  **Exercise\_Image\_ID,**  **Exercise\_Image\_URL,**  in the **Exercise image table.** The system will then verify and search for any existing exercise image and display the images that match the given search criteria. | | | |
| **PRE-CONDITION:** | The Employee must be logged into the system and have the right access levels and functionality to be able to search an exercise image | | | |
| **TRIGGER:** | The employee wants to search for an exercise images’ information from the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee selects to search for a exercise image from the system | | **Step 2:** The system will prompt the employee to enter any keywords that match any existing:  **Exercise\_Image\_ID,**  **Exercise\_Image\_URL.** | |
|  | **Step 3:** The employee will enter any keyword or phrase that will match any existing:  **Exercise\_Image\_ID,**  **Exercise\_Image\_URL.**  on the system. | |  | |
|  | **Step 4:** The employee will request to search | | **Step 5:** The system will capture the entered search keyword or phrase | |
|  |  | | **Step 6:** The system will validate the entered search criteria so that it is a valid search keyword or phrase. | |
|  |  | | **Step 7:** The captured keyword or phrase is valid, and the system will check the:  **Exercise\_Image\_ID,**  **Exercise\_Image\_URL.**  details for all existing exercise images in the Exercise image **table** to see if any match the captured search criteria. | |
|  |  | | **Step 8:** The system returns a list of all image(s) matching the given search criteria. | |
| **ALTERNATE COURSES:** | **Alt Step 6:** The captured keyword or phrase is invalid.  Return to step 4. | | | |
|  | **Alt Step 8:** No values on the system match the given search criteria. The system will display an error message.  Return to step 4. | | | |
| **CONCLUSION:** | The system has successfully searched for the required exercise image based off of the given search criteria and displayed the results to the employee. | | | |
| **POST-CONDITION:** | The system successfully finds the exercise image on the system. | | | |
| **BUSINESS RULES** | None | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None | | | |
| **ASSUMPTIONS:** | None | | | |

Table 8.8.24

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Iain Steyn**  Janco Steenkamp (checked) | | **Version 1.3** | | |
| **USE CASE NAME:** | **Create Exercise Plan** | | | **Use Case Type:** |
| **USE CASE ID:** | 8.24 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The use case describes the event that an employee would like to create an exercise plan on the system. The employee will create an exercise plan on the create exercise plan page which will be displayed by the system when the employee wants to create an exercise plan. The employee will select multiple workouts to create an exercise plan  which will be saved in the **Exercise\_Plan table.** The system will then capture and validate the entered details and check to see if the exercise plan - created by the user – doesn’t already exist on the system. If everything is done correctly the system will display a success message to the employee. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to create an exercise plan on the system. | | | |
| **TRIGGER:** | The employee requests to create an exercise plan on the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee requests to create an exercise plan on to the system. | | **Step 2:** The system will display the create exercise plan screen and inform the employee to create the exercise plan. | |
|  | **Step 3:** The employee will select the relevant **Workout\_IDs** to be added to the new exercise plan | | **Step 4:** The system will capture the selected details:  **Workout\_IDs.** | |
|  |  | | **Step 5:** The system will validate the selected details so that it is a valid exercise plan. | |
|  |  | | **Step 6:** The system will automatically generate a unique **Exercise\_Plan\_ID** and add it to the **Exercise\_Plan table**. | |
|  |  | | **Step 7:** The system will add the exercise plan details to the **Exercise\_Plan table**. | |
|  |  | | **Step 8:** The system will display a success message to the employee and return the employee to the home page. | |
|  | **Alt Step 3:** The employee selects to create a workout   * **Invoke Use Case 8.26 Create Workout** | | | |
| **ALTERNATE COURSES:** | **Alt Step 5:** The entered exercise plan details are invalid. The system will display an error message.   * Return the user to step 3. | | | |
| **CONCLUSION:** | The use case ends when the exercise plan has been created, successfully added to the system and a success message is displayed to the employee. | | | |
| **POST-CONDITION:** | A new exercise plan has been successfully created and added to the system. | | | |
| **BUSINESS RULES** | Only an employee with the required access levels can create a workout. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The workouts selected exists on the system | | | |
| **OPEN ISSUES:** | None. | | | |

Table 8.8.24

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **JessterFIT** | | **Date: 2019/03/30** | | |
| **Authors(s): Iain Steyn** | | **Version 1.3** | | |
| **USE CASE NAME:** | **Create Workout** | | | **Use Case Type:** |
| **USE CASE ID:** | 8.25 | | | Business Requirements: x |
| **PRIORITY:** | High | | | System Analysis: ◻ |
| **SOURCE:** | JessterFIT | | | System Design: ◻ |
| **PRIMARY BUSINESS ACTOR** | Employee | | | |
| **PRIMARY SYSTEM ACTOR** | None. | | | |
| **OTHER PARTICIPATING ACTORS:** | None. | | | |
| **OTHER INTERESTED STAKEHOLDERS:** | None. | | | |
| **DESCRIPTION:** | The Use Case describes the event that an employee would like to create a workout on the system. The employee will create a workout on the create workout page which will be displayed by the system when the employee wants to create a workout. The employee will add all the required details for the new workout such as:  **Exercise\_ID, Set\_ID**  which will be saved in the **Workout table.** The system will then capture and validate the entered details and check to see if the workout - created by the user – doesn’t already exist on the system. If everything is done correctly the system will display a success message to the employee. | | | |
| **PRE-CONDITION:** | The employee must be logged into the system and have the right access levels and functionality to be able to create a workout on the system. | | | |
| **TRIGGER:** | The employee wants to create a workout on the system. | | | |
| **TYPICAL COURSE** | Actor Action | | System Response | |
| **OF EVENTS:** | **Step 1**: The employee wants to create a workout on to the system. | | **Step 2:** The system will display the create workout screen and inform the employee to create the workout. | |
|  | **Step 3:** The employee will select the relevant **Exercise\_ID, Set\_ID** to be added to the new workout | | **Step 4:** The system will capture the selected details:  **Exercise\_ID, Set\_ID.** | |
|  |  | | **Step 5:** The system will validate the selected details so that it is a valid workout. | |
|  |  | | **Step 6:** The system will automatically generate a unique **Workout\_ID** and add it to the **Workout table**. | |
|  |  | | **Step 7:** The system will add the workout details to the **Workout table**. | |
|  |  | | **Step 8:** The system will display a success message to the employee and return the employee to the home page. | |
| **ALTERNATE COURSES:** | **Alt Step 5:** The entered work out details are invalid. The system will display an error message.   * Return the user to step 3. | | | |
| **CONCLUSION:** | The use case ends when the workout has been created, successfully added to the system and a success message is displayed to the employee. | | | |
| **POST-CONDITION:** | A new workout has been successfully created and added to the system. | | | |
| **BUSINESS RULES** | Only an employee with the required access levels can create a workout. | | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | None. | | | |
| **ASSUMPTIONS:** | The workout does not exist on the system. | | | |
| **OPEN ISSUES:** | None. | | | |

Table 8.8.25

Conclusion

This section serves an imperative aspect for the Functional Specification document as it gives us an outline of how the various actors interact with the system and how each requirement will be met. The Use Case narratives will be a guide throughout this document as it serves as one of the foundations, allowing stakeholders to understand any business rules or details about any use cases that may seem ambiguous.

Process Models of Proposed System

Introduction

This section is composed of a complete context diagram, functional decomposition diagram, data flow diagrams (high, middle and primitive level) as well as a complete Data Dictionary for the data flow diagrams. These diagrams are laid out to support the logical view of the system.

3.1 Context Diagram

3.2 Functional Decomposition Diagrams

Data Flow Diagrams

High Level Data Flow Diagrams

* 1. Admin Subsystem



Continued:

##### 2. Product Subsystem

3. User Subsystem

4. Employee Subsystem

5. Client Subsystem

6. Reporting Subsystem



7. Sale Subsystem

8. Exercise Subsystem





Mid-Level Data Flow Diagrams

1. Admin Subsystem









2. Product Subsystem

3. User Subsystem

4. Employee Subsystem





5. Client Subsystem

6. Reporting Subsystem











7. Sale Subsystem

8. Exercise Subsystem

Mid Level Add Exercise Plan TyPe





Add Sets



Add Exercise



Add exercise plan



Search Exercise Plan



Add Exercise Body Type



Search Exercise Body Type



Add Workout



Search Workout



Add Exercise Image



Remove Exercise Image



Search Exercise Image



Create Exercise Plan

Primitive Data Flow Diagrams

1. Admin Subsystem

1.1 Add New Subscription Type

1.2 Maintain Subscription Type

1.3 Search Subscription Type



1.4 Add New Challenge



1.5 Maintain Challenge



1.6 Search challenge



1.7 Add Personal training books slots



1.8 Add Challenge consultation booking slots



1.9 Confirm payment



1.10 Search Subscription



1.11 Search personal training booking



1.12 Search challenge consultation booking



1.13 Maintain Company



2. Product Subsystem

2.1 Add new product type

2.2 Maintain product type



2.3 Search product type



2.4 Add new product



2.5 Maintain product



2.6 Search product



2.7 Add new supplier

2.8 Maintain supplier



2.9 Search supplier



2.10 Add New Product Order

2.11 Pay Supplier

2.12 Confirm Supplier Order

3. User Subsystem

3.1 Login



3.2 Update password

3.3 Forgotten password



3.4 Log out



3.5 Add Access level

3.6 Maintain Access level



3.7 Search Access Level

3.8 Add Functionality

3.9 Maintain Functionality

3.10 Search Functionality

**4. Employee Subsystem**

4.1 Add new employee

4.2 Maintain employee



4.3 Search employee



4.4 Add New Employee Type



4.5 Maintain Employee Type



4.6 Search Employee Type



5. Client Subsystem

5.1 Add New Client



5.2 Maintain Client



5.3 Search Client



5.4 Signup for a Subscription

5.5 Cancel Subscription

5.6 Register for a Challenge



5.7 Unregister for a Challenge



5.8 Purchase Exercise Plan



5.9 Purchase Product/s



5.10 Make new Personal training Booking

5.11 Cancel Personal Training Booking

5.12 Make new Challenge Consultation Booking

5.13 Cancel Challenge Consultation Booking

5.14 Make payment

6. Reporting Subsystem

6.1 Generate Sales Report



6.2 Generate Subscription Report

6.3 Generate Client Progress Report



6.4 Generate Product Levels Report



6.5 Generate Challenge Booking Report

**7. Sale Subsystem**

7.1 Search Sale

7.2 Return sale

**8. Exercise Subsystem**

4. Data Model

Introduction:

This section contains the data model, showing all data that will be stored in the information system and the structure that the data will be placed into. The logical Entity Relationship Diagram illustrates the attributes used to describe each entity and the cardinality of the relationships between entities.

##### Please refer to the attached Appendix: Logical Data Model 3rd Normal Form ERD

Conclusion

This section contained the data model, modeled using the logical Entity Relationship Diagram, representing the structure of the data in Third Normal Form.

5. Interfaces and other Inputs

Introduction

This section outlines the detailed interfaces and input depicted in the context diagram (**See Section 3.1**). This section will go into further detail showing the data used in processes from start to finish and the description, purpose of input data, when the data will be used, the entities and attributes associated with entered data as well as the logical layout.

| Use Case Number | Flow Line | Description | Purpose | When it will be used | Entities and attributes | Logical Layout |
| --- | --- | --- | --- | --- | --- | --- |
| 1. Admin Subsystem | | | | | | |
| 1.1 Add new subscription type | 1.1. | The employee requests to add a new subscription type to the system. | Triggering input to start the adding new subscription type process | When new subscription type details need to be added to the system. | Employee | Requested by employee |
| 1.1 | The employee will enter all the required subscription type details | Details of new subscription type details requesting to be added to the system. | When new subscription type details need to be added to the system. | Employee  **Subscription\_Type\_Desc**  **Subscription\_Type\_Price** | Entered by employee |
| 1.2 Maintain subscription type | 1.2. | The employee requests to maintain a subscription type from the maintain subscription option | Triggering input to start the maintenance of a subscription type process. | When a subscription type details need to be maintained on the system. | Employee | Requested by employee |
| 1.2 | The employee will select the edit subscription type option for the subscription type that they would like to edit. | Input needed to continue maintenance on the subscription type process. | When a subscription type details need to be maintained on the system. | Employee | Selected by employee |
| 1.2 | The employee will update the subscription type and save it to the system. | Details of maintenance for subscription type process | When a subscription type details need to be maintained and saved on the system. | Employee  **Subscription­\_Type\_Desc**  **Subscription\_Type\_Price** | Entered by employee |
| Alt 1.2. | The employee selects the remove subscription type option | Triggering input to start the removal of a subscription type process. | When a subscription type details need to be removed from the system. | Employee | Selected by employee |
| ALT 1.2. | The user confirms that they would like to delete the selected subscription type | Input needed to complete the removal of a subscription type process. | When a subscription type details need to be removed from the system. | Employee | Selected by employee |
| 1.3 Search subscription type | 1.3 | The employee requests to search for a subscription type on the system. | Triggering input to start the search subscription type | When a subscription type details need to be searched on the system | Employee | Requested by employee |
| 1.3 | The employee will enter any keyword or phrase that will match any existing subscription type. | Details for search subscription type process. | When a subscription type details need to be searched on the system | Employee  **Subscription\_Type\_ID,**  **Subscription\_Type\_Description,**  **Subscription\_Type\_Price,** | Entered by employee |
| 1.3 | The employee will request to search. | Triggering input to complete the search subscription process. | When an entered subscription type details need to be searched on the system | Employee | Selected by employee |
| 1.4 Add new challenge | 1.4. | The employee requests to add a new challenge to the system. | Triggering input to start the adding new challenge process. | When new challenge details need to be added to the system. | Employee | Requested by employee |
| 1.4. | Employee enters details requested by the system. | Details of new challenge details requesting to be added to the system. | When new challenge details need to be added to the system. |  | Entered by the employee |
| 1.4 | Employee confirms the correctness of the information entered. | Details of new challenge need to be confirmed. | When the system displays the new challenge, data entered by the employee. | Employee | Confirmed by employee. |
| 1.5 Maintain challenge | 1.5. | Employee requests to maintain a challenge on the system. | Triggering input to start the maintenance of a challenge process. | When challenge details need to be maintained on the system. | Employee | Requested by the employee |
| 1.5. | The employee will select the edit challenge option for the challenge they would like to edit. | Input needed to select specific challenge needing maintenance | When challenge details need to be maintained on the system. | Employee | Selected by employee |
| 1.5. | The employee will update the challenge and save it to the system. | Details of maintenance for challenge process. | When a challenge details need to be maintained and saved on the system. | Employee  **Challenge\_ID, Challenge\_Start\_Date and Challenge\_End\_Date**. | Entered by employee |
| Alt 1.5. | The employee selects the remove challenge option | Triggering input to start the removal of a challenge process. | When a challenge details need to be removed from the system. | Employee | Selected by employee |
| ALT 1.5. | The user confirms that they would like to delete the selected challenge | Input needed to complete the removal of a challenge process. | When a challenge details need to be removed from the system. | Employee | Selected by employee |
| Alt 1.5. | The user revokes the decision to remove the selected challenge and selects the option to cancel confirmation. | To cancel removal of challenge details process. | When an employee wants to cancel a removal of challenge details. | Employee | Selected by employee |
| 1.6 Search challenge | 1.6. | The employee requests to search for a challenge on the system. | Triggering input to start the search challenge process. | When a challenge details need to be searched on the system | Employee | Requested by employee |
| 1.6. | The employee will enter any keyword or phrase that will match any existing challenge. | Details for search challenge process. | When a challenge details need to be searched on the system | Employee  **Challenge\_ID** | Entered by employee |
| 1.6. | The employee will request to search. | Triggering input to complete the search challenge process. | When an entered challenge details need to be searched on the system | Employee | Selected by employee |
| 1.7 Add personal training booking slots | 1.7. | The employee requests to add personal training booking slots to the system. | Triggering input to start the add new personal training booking slots process. | When an employee needs to add new personal training booking slots to the system. | Employee | Requested by employee |
| 1.7. | Employee selects the day on which they choose to add the new personal training booking slots. | To select a day on which personal training booking slots will be added. | When the system prompts the user to select a day for personal training booking slots to be added to. | Employee | Selected by employee |
| 1.7. | Employee enters personal trainers’ details. For specified hours they are available for on the day. | Details for personal trainers who can be booked on the day the new personal training slots were added. | When the system prompts the employee to enter personal trainers’ details. | Employee | Entered by employee |
| 1.7. | Employee proceeds with added details. | Details needing to be accepted. | System prompts employee to decide whether he wants to go back and add to missing slots. | Employee | Selected by employee |
| 1.8 Add challenge consultation booking slots | 1.8. | The employee requests to add challenge booking slots to the system. | Triggering input to start the add new challenge consultation booking slots process. | When an employee needs to add new challenge consultation booking slots to the system. | Employee | Requested by employee |
| 1.8. | Employee selects the day on which they choose to add the challenge consultation booking slots. | To select a day on which challenge consultation booking slots will be added. | When the system prompts the user to select a day for challenge consultation booking slots to be added to. | Employee | Selected by employee |
| 1.8. | Employee enters personal trainers’ details. For specified hours they are available for on the day. | Details for personal trainers who can be booked on the day the new challenge consultation slots were added. | When the system prompts the employee to enter personal trainers’ details. | Employee | Entered by employee |
| 1.8 | Employee proceeds with added consultation details. | Details needing to be accepted. | System prompts employee to decide whether he wants to go back and add to missing slots. | Employee | Selected by employee |
| 1.9 Confirm payment | 1.9 | The customer made a payment to the business. | Initial request to start the confirm payment process | When an employee |  | Requested by client |
| 1.9 | The employee confirms the payment details. | Input needed to complete the confirm payment process | When an employee wants to confirm a payment details on the system | Employee  **Proof of Payment**  **Sale\_ID** | Confirmed by employee |
| 1.10 Search subscription | 1.10 | The employee requests to search for a subscription on the system. | Triggering input to start the search subscription process. | When a subscription details need to be searched on the system | Employee | Requested by employee |
| 1.10 | The employee will enter any keyword or phrase that will match any existing subscription. | Details for search subscription process. | When a subscription details need to be searched on the system | Employee  **Subscription\_ID, Subscription\_Type** | Entered by employee |
| 1.10 | The employee will request to search. | Triggering input to complete the search subscription process. | When an entered subscription details need to be searched on the system | Employee | Selected by employee |
| 1.11 Search personal training booking | 1.11 | The employee requests to search for a personal training booking on the system. | Triggering input to start the search personal training booking process. | When a personal training booking details need to be searched on the system | Employee | Requested by employee |
| 1.11 | The employee will enter any keyword or phrase that will match any existing personal training booking. | Details for search personal training booking process. | When a personal training booking details need to be searched on the system | Employee  **Client\_ID, Employee\_ID, Training\_Date\_Booked, Training\_Venue** | Entered by employee |
| 1.11 | The employee will request to search. | Triggering input to complete the search personal training booking process. | When an entered personal training booking details need to be searched on the system | Employee | Selected by employee |
| 1.12 Search challenge consultation booking | 1.1 | The employee requests to search for a challenge consultation booking on the system. | Triggering input to start the search challenge consultation booking process. | When a challenge consultation booking details need to be searched on the system | Employee | Requested by employee |
| 1.12 | The employee will enter any keyword or phrase that will match any existing challenge consultation booking. | Details for search challenge consultation booking process. | When a challenge consultation booking details need to be searched on the system | Employee  **Client\_ID, Employee\_ID, Training\_Date\_Booked, Training\_Venue,** | Entered by employee |
| 1.12 | The employee will request to search. | Triggering input to complete the search challenge consultation booking process. | When an entered challenge consultation booking details need to be searched on the system | Employee | Selected by employee |
| 1.13 Maintain company | 1.13 | The employee requests to maintain the company from the maintain company option. | Initial request to start the maintain company process. | When an employee wants to maintain the company details | Employee | Requested by employee |
|  | 1.13 | The employee will select the edit detail option for the details they would like to edit. | Input needed to complete the maintain company process | When an employee wants to maintain the company details | Employee  **Company\_Name, Company\_Address, Company\_RegisterNo, Company\_Email, Company\_Contact\_No, Account\_Name, Account\_No, Account\_Type, Account\_Branch, Account\_Branch\_Code**. | Entered by employee |
|  | 1.13 | The employee will update the company and save it to the system. | Trigger input recorded to save the data | When an employee wants to submit maintained company details | Employee | Submitted by employee |
| 1. Product Subsystem | | | | | | |
| 2.1 Add new product type | 2.1 | Employee requests to add a new product type to the system. | Too add a new product type to the system. | When an employee needs to add a new product type to the system. | Employee | Requested by employee |
|  | 2.1. | Employee enters new product type details. | Input to complete the new product type details. | When an employee needs to add a new product to the system. | Employee  **Prod\_Type\_ID**  **Prod\_Type\_Description**, | Entered by employee |
| 2.2 Maintain product type | 2.2 | Employee requests to maintain a product type on the system. | Too maintain a products types details on the system. | When an employee needs to maintain a product types details on the system. | Employee | Requested by employee |
|  | 2.2. | Employee will enter search criteria. | To search for the specific product types needing maintenance. | When system requests search criteria to search by. | Employee  **Product\_Type\_ID, Prod\_Type\_Description** | Entered by employee |
|  | 2.2 | The employee will select the edit product type option for the product type that they would like to edit. | To tell the system you choosing to edit a product type. | When an employee needs to maintain a product types details on the system. | Employee | Selected by employee |
|  | 2.2 | The employee will update the product type and save it to the system. | To change details of the selected product type. | When an employee needs to maintain a product types details on the system. | Employee  **Prod\_Type\_Description** | Updated by employee |
|  | Alt 2.2 | The employee selects the remove product type option. | To remove details of the product type selected. | When an employee needs to delete a product types details from the system. | Employee | Selected by employee |
|  | Alt 2.2 | The user confirms that they would like to delete the selected product type. | To remove details of the product type selected. | When an employee needs to delete a product types details from the system. | Employee | Selected by employee |
|  | Alt 2.2 | The user revokes the decision to remove the selected product type and selects the option to cancel confirmation. | To remove details of the product type selected. | When an employee needs to delete a product types details from the system. | Employee | Selected by employee |
| 2.3 Search product type | 2.3 | The employee requests search for a product type on the system. | To search for a product type on the system. | When an employee needs to search for a product type on the system | Employee | Selected by employee |
|  | 2.3 | The employee will enter any keyword or phrase that will match any existing product type on the system. | Inputs to search specifically for a certain product type on the system. | When an employee needs to search for a product type on the system | Employee  **Prod\_Type\_Description** | Entered by employee |
|  | 2.3 | The employee will proceed with the search. | To search for a product type on the system. | When an employee needs to search for a product type on the system | Employee | Selected by employee. |
| 2.4 Add new product | 2.4. | Employee requests to add a new product to the system. | Too add a new product to the system. | When an employee needs to add a new product to the system. | Employee | Requested by employee |
| 2.4. | Employee enters new product details. | Input to complete the new product details. | When an employee needs to add a new product to the system. | Employee  **Prod\_Name**, **Prod\_Description**, **Prod\_Price** and **Prod\_Quantity**. | Entered by employee |
| 2.5 Maintain product | 2.5. | Employee requests to maintain a product on the system. | Too maintain a products details on the system. | When an employee needs to maintain a products details on the system. | Employee | Requested by employee |
| 2.5. | Employee will enter search criteria. | To search for the specific product needing maintenance. | When system requests search criteria to search by. | Employee  **Product\_ID, Prod\_Name** | Entered by employee |
| 2.5. | The employee will select the edit product option for the product that they would like to edit. | To tell the system you choosing to edit a product. | When an employee needs to maintain a products details on the system. | Employee | Selected by employee |
| 2.5. | The employee will update the product and save it to the system. | To change details of the selected product. | When an employee needs to maintain a products details on the system. | Employee  **Prod\_Name**, **Prod\_Description**, **Prod\_Price** and **Prod\_Quantity** | Updated by employee |
| Alt 2.5 | The employee selects the remove product option. | To remove details of the product selected. | When an employee needs to delete a products details from the system. | Employee | Selected by employee |
| Alt 2.5 | The user confirms that they would like to delete the selected product. | To remove details of the product selected. | When an employee needs to delete a products details from the system. | Employee | Selected by employee |
| Alt 2.5 | The user revokes the decision to remove the selected product and selects the option to cancel confirmation. | To remove details of the product selected. | When an employee needs to delete a products details from the system. | Employee | Selected by employee |
| 2.6 Search product | 2.6 | The employee requests search for a product on the system. | To search for a product on the system. | When an employee needs to search for a product on the system | Employee | Selected by employee |
| 2.6 | The employee will enter any keyword or phrase that will match any existing product on the system. | Inputs to search specifically for a certain product on the system. | When an employee needs to search for a product on the system | Employee  **Prod\_Name**, **Prod\_Description**, **Prod\_Price** and **Prod\_Quantity** | Entered by employee |
| 2.6 | The employee will proceed with the search. | To search for a product on the system. | When an employee needs to search for a product on the system | Employee | Selected by employee. |
| 2.7 Add new supplier | 2.7 | Employee requests to add a new supplier to the system. | Triggering input to start the add new supplier process | When an employee needs to add a new supplier to the system. | Employee | Requested by employee |
| 2.7 | Employee enters new supplier details. | Input details needed to complete the add new supplier process. | When an employee needs to add a new supplier to the system. | Employee  **Supplier\_Description,**  **Supplier\_Email,**  **Supplier\_Phone,**  **Supplier\_Address,**  **Supplier\_Contact,** | Entered by employee |
| 2.8 Maintain supplier | 2.8 | Employee requests to maintain a supplier on the system. | Too maintain a suppliers’ details on the system. | When an employee needs to maintain a suppliers’ details on the system. | Employee | Requested by employee |
|  | 2.8. | Employee will enter search criteria. | To search for the specific supplier needing maintenance. | When system requests search criteria to search by. | Employee  **Supplier\_Description,**  **Supplier\_Email,**  **Supplier\_Phone,**  **Supplier\_Address,**  **Supplier\_Contact,** | Entered by employee |
|  | 2.8 | The employee will select the edit supplier option for the supplier that they would like to edit. | To tell the system you choosing to edit a supplier. | When an employee needs to maintain a suppliers’ details on the system. | Employee | Selected by employee |
|  | 2.8 | The employee will update the supplier and save it to the system. | To change details of the selected supplier. | When an employee needs to maintain a suppliers’ details on the system. | Employee  **Supplier\_Description,**  **Supplier\_Email,**  **Supplier\_Phone,**  **Supplier\_Address,**  **Supplier\_Contact,** | Updated by employee |
|  | Alt 2.8 | The employee selects the remove supplier option. | To remove details of the supplier selected. | When an employee needs to delete a suppliers’ details from te system. | Employee  **Supplier\_Description,**  **Supplier\_Email,**  **Supplier\_Phone,**  **Supplier\_Address,**  **Supplier\_Contact,** | Selected by employee |
|  | Alt 2.8 | The user confirms that they would like to delete the selected supplier. | To remove details of the supplier selected. | When an employee needs to delete a suppliers’ details from the system. | Employee | Selected by employee |
| 2.9 Search | 2.9 | The employee will enter any keyword or phrase that will match any existing supplier on the system. | Inputs to search specifically for a certain supplier on the system. | When an employee needs to search for a product on the system | Employee  **Supplier\_ID,**  **Supplier\_Description,**  **Supplier\_Email,** | Entered by employee |
| 2.9 | The employee will proceed with the search. | Input to complete the search supplier process. | When an employee needs to search for a supplier on the system | Employee | Selected by employee. |
| 2.10 Add new product order | 2.10 | The employee will request to create a new supplier order. | Input to request to add a new product order. | When an employee needs to add a new product order to the system. | Employee | Selected by employee |
| 2.10 | The employee will enter the needed product order details. | Input to add a new product orders details to the system. | When an employee would like to add new product order details to the system. | Employee  **Supplier\_Order\_ID, Product\_ID, Quantity, Amount** | Entered by employee |
| 2.10 | The employee will confirm the product order details and submit the product order. | When an employee wants to submit new product order details. | When an employee would like to add new product order details to the system. | Employee | Selected by employee |
| 2.11 Pay supplier | 2.11 | The employee will search the supplier order details | Input needed to pay a supplier for an order received. | When an employee wants to pay a supplier on the system. | Employee  **Supplier\_Order\_ID** | Entered by employee |
| 2.11 | The employee will use the supplier details to make the product order payment. | Input needed to pay a supplier for an order received. | When an employee wants to pay a supplier on the system. | Employee | Selected by employee |
| 2.12 Confirm supplier order | 2.12 | The employee will search the supplier order. | Input needed to confirm a supplier order. | When an employee wants to confirm a supplier order | Employee  **Supplier\_Order\_ID** | Entered by employee |
| 2.12 | The employee will enter the supplier order details received to the system. | Input needed to confirm a supplier order. | When an employee wants to confirm a supplier order | Employee | Entered by employee |
| 1. User Subsystem | | | | | | |
| 3.1 Login | 3.1. | The user requests to access their account, by selecting to log in. | Input for requesting to log into the system. | When a user wishes to log into the system. | User | Selected by user |
| 3.1 | The user provides current password as requested by the system | Input for login details | When a user wishes to log into the system. | User  **Username**  **Password** | Entered by user |
| 3.2 Update password | 3.2 | The employee requests to update their password. | Initial request to update password | When a user wishes to update a password | User | Selected by user |
| 3.2 | The employee provides current password as requested by the system | Input for old password details | When a user wishes to update a password | User  **Password**  **Username** | Entered by user |
| 3.2 | The employee inputs the new password details | Input for new password details | When a user wishes to update a password | User  **Password**  **Username** | Entered by user |
| 3.3 Forgotten password | 3.3 | The user requests for new password since they have forgotten their password | Initial request to reset a forgotten password | When a user needs to reset a forgotten password | User | Selected by user |
| 3.3 | The user enters their email address | Input needed to reset a forgotten password. | When a user needs to reset a forgotten password | User  **Employee\_Email\_Address**  **Client\_Email\_Address** | Entered by User |
| 3.4 Logout | 3.4 | The user request to log out of the system by selecting logout. | Input request to log out of the system. | When an employee needs to log out of the system. | User | Selected by user |
| 3.4 | The user selects to confirm logout. | Request to confirm log out. | When a user needs to log out of the system. | User | Selected by user |
| 3.5 Add access level | 3.5 | An Employee wants to add a new Access level onto the system. | Initial request to add a new access level to the system. | When an employee needs to add a new access level to the system. | Employee | Selected by employee |
| 3.5 | The Employee enters the information as requested by the system | Input details needed to add an access level. | When an employee needs to add a new access level to the system. | Employee | Entered by employee |
| 3.5 | The employee confirms the correctness of the information entered. | Request to confirm details are correct. | When an employee needs to add a new access level to the system. | Employee | Selected by employee |
| 3.6 Maintain access level | 3.6 | An employee requests to maintain an access Level on the system. | Initial input request to maintain an access level. | When an employee needs to add a new access level to the system. | Employee | Selected by employee |
| 3.6 | The Employee Selects the Access Level which they want to maintain | Request to select access level needing maintenance | When an employee needs to add a new access level to the system. | Employee | Selected by employee |
| 3.6 | The Employee edits the details which needed to be updated and select to update | Input for details needing to maintain. | When an employee needs to add a new access level to the system. | Employee | Entered by employee |
| Alt Step 3.6 | The Employee selects to remove the exercise type  The use case ends | Request to remove access level from system. | When an employee needs to remove an access levels details from the system. | Employee | Selected by employee |
| 3.7 Search access level | 3.7 | Employee requests to make a search for an access level. | Initial request to search for an access level’s details. | When an employee needs to search for an access level’s details. | Employee | Selected by employee |
| 3.7 | The Employee enters the details required in the search criteria | Input needed to search for a specific access level | When an employee needs to search for an access level’s details | Employee  **Access\_Level\_ID** | Entered by employee |
| 3.7 | The employee selects the specific access level they are looking for. | Request to see searched access level details. | When an employee needs to search for an access level’s details | Employee | Selected by employee |
| 3.8 Add functionality | 3.8 | An Employee requests to add new functionality onto the system. | Initial request to add functionality to the system | When an employee needs to add functionality to the system. | Employee | Selected by employee |
| 3.8 | The Employee enters the information as requested by the system. | Input needed to add new functionality to the system | When an employee needs to add functionality to the system. | Employee | Entered by employee |
| 3.8 | The employee confirms the correctness of the information entered. | Selected to confirm the correctness of the functionality details entered. | When an employee needs to add functionality to the system. | Employee | Selected by employee |
| 3.9 Maintain Functionality | 3.9 | An employee requests to maintain functionality on the system. | Initial input request to maintain a functionality. | When an employee needs to add a new access level to the system. | Employee | Selected by employee |
|  | 3.9 | The Employee Selects the functionality which they want to maintain | Request to select functionality needing maintenance | When an employee needs to add a new functionality to the system. | Employee | Selected by employee |
|  | 3.9 | The Employee edits the details which needed to be updated and selects to update | Input for details needing to maintain. | When an employee needs to add new functionality to the system. | Employee | Entered by employee |
|  | Alt Step 3.9 | The Employee selects to remove functionality  The use case ends | Request to remove access level from system. | When an employee needs to remove an access levels details from the system. | Employee | Selected by employee |
| 3.10 Search functionality | 3.10 | Employee requests to make a search for a specific functionality. | Initial request to search a specific functionality on the system | When an employee needs to search for a specific functionality on the system. | Employee | Selected by employee |
| 3.10 | The Employee enters the details required in the search criteria | Input for search criteria to find specific functionality | When an employee needs to search for a specific functionality on the system. | Employee  **Functionality\_ID** | Entered by employee |
| 3.10 | The employee selects the specific functionality they are looking for. | Request to select specific functionality searched. | When an employee needs to search for a specific functionality on the system. | Employee | Entered by employee |
| 1. Employee Subsystem | | | | | | |
| 4.1 Add new employee | 4.1 | The owner requests to add a new employee to the system. | Initial request to start the add new employee process | When an owner wants to add a new employee’s details to the system. | Owner | Selected by owner. |
| 4.1 | Owner enters new employee details. | Input for the add new employee process | When an owner wants to add a new employee’s details to the system. | Owner  **Employee\_Name**, **Employee\_Surname**, **Employee\_Email**, **Employee\_Address**, **Employee\_Contact** and **Employee\_IDnumber** | Entered by owner |
| 4.2 Maintain employee | 4.2 | The owner requests to maintain an employee’s details. | Request to start the maintain employee process | When an owner needs to maintain an employee’s details on the system. | Owner | Selected by owner |
| 4.2 | The owner will select the edit employee option for the employee that they would like to edit. |  |  |  |  |
| 4.3 Search employee | 4.3 | The owner requests to search for an employee on the system. | Initial request to start the search employee process. | When the owner needs to search an employee’s details on the system. | Owner | Selected by owner |
| 4.3 | The owner will enter any keyword or phrase that will match any existing employee. | Input to search employee process. | When the owner needs to search an employee’s details on the system. | Owner  **Employee\_ID**  **Employee\_Name** | Entered by owner |
| 4.4 Add new employee type | 4.4 | The owner requests to add a new employee type to the system | Initial request to start the add new employee type process. | When an employee wants to add new a new employee type to the system. | Owner | Selected by owner |
| 4.4 | The owner will enter new employee type details | Input for the add new employee type process. | When an owner wants to add new a new employee type to the system. | Owner  **Employee\_Type\_Desc** | Entered by owner |
| 4.5 Maintain employee type | 4.5 | The owner requests to maintain an employee type. | Initial request to start the maintain employee type process | When an owner wants to maintain an employee type on the system. | Owner | Selected by owner |
| 4.5 | The owner will select the edit employee type option for the employee type that they would like to edit. | Request to edit employee type details. | When an owner wants to maintain an employee type on the system. | Owner | Selected by owner |
| 4.5 | The owner will update the employee type and save it to the system. | Input for maintain exercise type process. | When an owner wants to maintain an employee type on the system. | Owner  **Employee\_Type\_Desc** | Entered by owner |
| Alt 4.5 | The owner selects the remove employee type option. | Request to remove the employee type details from the system. | When an owner wants to remove an employee type from the system. | Owner | Selected by owner |
| Alt 4.5 | The user confirms that they would like to delete the selected employee type. | Confirm removal of employee type details | When an owner wants to remove an employee type from the system. | Owner | Selected by owner |
| 4.6 Search employee type | 4.6 | The owner requests to search for an employee type on the system. | Initial request to start the search employee type process. | When an owner wants to search an employee type. | Owner | Selected by owner |
| 4.6 | The owner will enter any keyword or phrase that will match any existing employee type | Input required to search for the employee type process | When an owner wants to search an employee type. | Owner  **Employee\_Type\_ID**  **Employee\_Type\_Desc** | Entered by owner |
| 4.6 | The owner will request to search. | Request to complete search process | When an employee wants to search an employee type. | Owner | Selected by owner |
| 1. Client Subsystem | | | | | | |
| 5.1 Add new client | 5.1 | The client requests to be added to the system. | Initial request to start the add new client process | When a client wants to add a new client to the system. | Client | Requested by client |
| 5.1 | The client will enter the details for the new client. | Input to the add new client process | When a client wants to add a new client to the system. | Client  **Client\_Name**  **Client\_Surname**  **Client\_Email**  **Client\_Phone\_Number**  **Client\_Address**  **Client\_Gender** | Entered by client |
| 5.2 Maintain client | 5.2 | The client requests to maintain his/her details. | Initial request to start the maintain client process | When a client wants to maintain their details on the system. | Client | Requested by client |
| 5.2 | The client will update his/her details and save it to the system | Input needed to complete the maintain client process. | When a client wants to maintain their details on the system. | Client  **Client\_Name**  **Client\_Surname**  **Client\_Email**  **Client\_Phone\_Number**  **Client\_Address**  **Client\_Gender** | Entered by client |
| 5.3 Search client | 5.3 | The employee requests to search for a client on the system. | Initial request to start the search client process | When an employee needs to search a client’s details on the system | Employee | Requested by employee |
| 5.3 | The employee will enter any keyword or phrase that will match any existing client | Input needed to complete the search process | When an employee needs to search a client’s details on the system | Employee  **Client\_Name**  **Client\_Surname**  **Client\_Email**  **Client\_Phone\_Number**  **Client\_Address**  **Client\_Gender** | Entered by employee |
| 5.4 Signup for a subscription | 5.4 | The client requests to sign up for a subscription | Initial request to the signup for subscription process | When a client wants to sign up for a subscription on the system | Client | Requested by client |
| 5.4 | The client will select the option best suited. | Needed to finish the signup for subscription process | When a client wants to sign up for a subscription on the system | Client | Selected by client |
| 5.5 Cancel subscription | 5.5 | The client requests to cancel his/her subscription. | Needed to finish the cancel subscription process. | When a client wants to sign up for a subscription on the system | Client | Selected by client |
| 5.5 | The client will select the **Cancel Subscription** option. | Needed to finish the cancel subscription use | When a client wants to sign up for a subscription on the system | Client | Selected by client |
| 5.6 Register for a challenge | 5.6 | The client wants to register for a challenge | Request to start the register for challenge process. | When a client wants to register for a challenge | Client | Requested by client |
| 5.6 | The client will select the challenge best suited. | Needed to finish the register for a challenge process | When a client wants to register for a challenge | Client | Selected by client |
| 5.7 Unregister for a challenge | 5.7 | The client requests to unregister for a challenge. | Request to start the unregister for challenge process. | When the client wants to unregister for a challenge on the system | Client | Requested by client |
| 5.7 | The client will select the **Cancel Challenge** option | Needed to finish the unregister for challenge process | When the client wants to unregister for a challenge on the system | Client | Selected by client |
| 5.8 Purchase an exercise plan | 5.8 | The client navigates to the exercise plan screen | Needed to start the purchase an exercise plan process | When the client wants to purchase an exercise plan | Client | Requested by client |
| 5.8 | The client selects the exercise plan to be purchased | Needed to continue with purchase exercise plan process | When the client wants to purchase an exercise plan | Client | Selected by client |
| 5.8 | The client selects the option to purchase the selected plan | Needed to continue with purchase exercise plan process | When the client wants to purchase an exercise plan | Client | Selected by client |
| 5.8 | The client follows the instructions and sends the proof of payment to the necessary email address | Input for purchase exercise plan process | When the client wants to purchase an exercise plan | Client | Sent by client |
| 5.9 Purchase product | 5.9 | The client navigates to the product screen | Needed to start the purchase product process | When the client needs to purchase a product. | Client | Requested by client |
| 5.9 | The client selects the product to be purchased | Needed to continue with purchase product process | When the client needs to purchase a product | Client | Selected by client |
| 5.9 | The client selects the option to purchase the selected plan | Needed to continue with purchase product process | When the client needs to purchase a product | Client | Selected by client |
| 5.9 | The client follows the instructions and sends the proof of payment to the necessary email address | Input for purchase product |  |  |  |
| 5.10 Make new personal training booking | 5.10 | The client requests to add a new personal training booking to the system | Request to start the make new personal training booking  process | When the client needs to make a personal training booking | Client | Requested by client |
| 5.10 | The client will enter a new personal training bookingdetail | Input for make new personal training booking  process | When the client needs to make a personal training booking | Client  **Personal\_training\_date** | Entered by client |
| 5.11 Cancel personal training booking | 5.11 | The client requests to cancel a personal training booking slot on the system. | Request to start cancel personal training booking process | When the client needs to cancel a personal training booking. | Client | Requested by the client |
| 5.11 | The client will select to cancel the relevant personal training booking slots | Needed to be selected to continue with the cancel personal training process | When the client needs to cancel a personal training booking | Client | Selected by client |
| 5.12 Make new challenge consultation booking | 5.12 | The client requests to add a new challenge consultation booking slot to the system. | Request to start make new challenge consultation booking | When the client needs to make a new challenge consultation booking. | Client | Request by client |
| 5.12 | The client will enter a new**,** on the add challenge consultation booking page and save it on the system. | Input needed to complete the make new challenge consultation booking | When the client needs to make a new challenge consultation booking. | Client  **Challenge\_Consultation\_Date** | Entered by client |
| 5.13 Cancel challenge consultation booking | 5.13 | The client requests to cancel a challenge consultation booking slot on the system. | Request to starts cancel consultation booking process | When the client needs to make a new challenge consultation booking. | Client | Requested by client |
| 5.13 | The client will select to cancel the relevant challenge consultation booking | Needed to be selected to complete the cancel consultation booking process | When the client needs to make a new challenge consultation booking. | Client | Selected by client |
| 5.14 Make payment | 5.14 | The client requests to make the payment | Request to start make payment process | When the client needs to make a payment on the system | Client | Requested by client |
| 5.14 | The client follows the instructions and sends the proof of payment to the necessary email address | Needed to confirm payment | When the client needs to make a payment on the system | Client |  |
| 1. Reporting Subsystem | | | | | | |
| 6.1 Generate sales report | 6.1 | Owner requests to generate a sales report for a specific period. | Trigger to start the generate sales report process | When the owner wishes to generate a sales report. | Owner | Requested by owner |
| 6.1 | Owner enters start and end date. | Input to determine the reporting period on which the generate sales report process must report on. | When the owner wishes to generate a sales report. | Owner  **Start\_Date**  **End\_Date** | Entered by owner |
| 6.1 | Owner chooses to use onscreen report | Request to use onscreen report generated by the generate sales report process. | When an owner wishes to observer a report | Owner | Chosen by owner |
| Alt 6.1 | Owner chooses to download the pdf format of the report | Request to download pdf format report generated by the generate sales report process. | When an owner wishes to observer a report | Owner | Chosen by owner |
| Alt 6.1 | Owner chooses to download the word format of the report | Request to download word format report generated by the generate sales report process. | When an owner wishes to observer a report | Owner | Chosen by owner |
| 6.2 Generate Subscription report | 6.2 | Owner requests to generate a subscription report for a specific period. | Trigger to start the generate subscription report process | When the owner wishes to generate a subscription report. | Owner | Requested by owner |
| 6.2 | Owner enters start and end date. | Input to determine the reporting period on which the generate subscription report process must report on. | When the owner wishes to generate a subscription report. | Owner  **Start\_Date**  **End\_Date** | Entered by owner |
| 6.2 | Owner chooses to use onscreen report | Request to use onscreen report generated by the generate subscription report process. | When an owner wishes to observe a report | Owner | Chosen by owner |
| Alt 6.2 | Owner chooses to download the pdf format of the report | Request to download pdf format report generated by the generate subscription report process. | When an owner wishes to observe a report | Owner | Chosen by owner |
| Alt 6.2 | Owner chooses to download the word format of the report | Request to download word format report generated by the generate subscription report process. | When an owner wishes to observe a report | Owner | Chosen by owner |
| 6.3 Generate client progress report | 6.3 | Owner requests to generate a client progress report for a specific period. | Trigger to start the generate client progress report process | When the owner wishes to generate a client progress report. | Owner | Requested by owner |
| 6.3 | Owner enters start and end date. | Input to determine the reporting period on which the generate client progress report process must report on. | When the owner wishes to generate a client progress report. | Owner  **Start\_Date**  **End\_Date** | Entered by owner |
| 6.3 | Owner enters valid client details to search. | Input needed to find client owner is wanting to generate client progress report on. | When the owner wishes to generate a client progress report. | Owner  **Client\_ID**  **Client\_Name** | Entered by owner |
| 6.3 | Owner chooses to use onscreen report | Request to use onscreen report generated by the generate client progress report process. | When an owner wishes to observe a report | Owner | Chosen by owner |
| Alt 6.3 | Owner chooses to download the pdf format of the report | Request to download pdf format report generated by the generate client progress report process. | When an owner wishes to observe a report | Owner | Chosen by owner |
| Alt 6.3 | Owner chooses to download the word format of the report | Request to download word format report generated by the generate client progress report process. | When an owner wishes to observe a report | Owner | Chosen by owner |
| 6.4 Generate product levels | 6.4 | Employee requests to generate a sales report for a specific period. | Trigger to start the generate product levels report process | When the employee wishes to generate a product levels report. | Employee | Requested by employee |
| 6.4 | Employee enters start and end date. | Input to determine the reporting period on which the generate product levels report process must report on. | When the employee wishes to generate a product levels report. | Employee  **Start\_Date**  **End\_Date** | Entered by employee |
| 6.4 | Employee chooses to use onscreen report | Request to use onscreen report generated by the generate product levels report process. | When an employee wishes to observe a report | Employee | Chosen by owner |
| Alt 6.4 | Employee chooses to download the pdf format of the report | Request to download pdf format report generated by the generate product report process. | When an employee wishes to observe a report | Employee | Chosen by owner |
| Alt 6.4 | Employee chooses to download the word format of the report | Request to download word format report generated by the generate product levels report process. | When an employee wishes to observe a report | Employee | Chosen by owner |
| 6.5 Generate challenge booking report | 6.5 | Owner requests to generate a challenge booking report for a specific period. | Trigger to start the generate challenge booking report process | When the owner wishes to generate a challenge booking report. | Owner | Requested by owner |
| 6.5 | Owner enters start and end date. | Input to determine the reporting period on which the generate challenge booking report process must report on. | When the owner wishes to generate a challenge booking report. | Owner  **Start\_Date**  **End\_Date** | Entered by owner |
| 6.5 | Owner chooses to use onscreen report | Request to use onscreen report generated by the generate challenge booking report process. | When an owner wishes to observe a report | Owner | Chosen by owner |
| Alt 6.5 | Owner chooses to download the pdf format of the report | Request to download pdf format report generated by the generate challenge booking report process. | When an owner wishes to observe a report | Owner | Chosen by owner |
| Alt 6.5 | Owner chooses to download the word format of the report | Request to download word format report generated by the generate challenge booking report process. | When an owner wishes to observe a report | Owner | Chosen by owner |
| 1. Sale Subsystem | | | | | | |
| 7.1 Search Sale | 7.1 | Employee requests to search a sale that has been made. | Trigger to start the search sale process. | When an employee wishes to search a sale made on the system. | Employee | Requested by employee |
| 7.1 | The employee will enter valid details to search for the sale made. | Input for the search sale process. | When an employee wishes to search for a sale made on the system. | Employee  **Sale\_ID** | Entered by employee |
| 7.2 Return sale | 7.2 | Client requests to return a sale that has been made. | Trigger to start the return sale process. | When a client wishes to return a sale made on the system. | Client | Requested by client |
| 7.2 | The client will enter all return details. | Input for the return sale process. | When a client wishes to return a sale made on the system. | Client  **Sale\_ID**  **Sale\_Amount**  **Sale\_Date** | Entered by employee |
| 1. Exercise Subsystem | | | | | | |
| 8.1 Add new exercise plan type | 8.1 | The employee requests to add a new exercise plan type to the system | Initial request to start the add new exercise plan type process. | When an employee wants to add a new exercise plan type to the system. | Employee | Selected by employee |
| 8.1 | The employee will enter new exercise plan type details | Input for the add new exercise plan type process. | When an employee wants to add a new exercise plan type to the system. | Employee  **Exercise\_Plan\_Type\_Desc** | Entered by employee |
| 8.2 Maintain exercise plan type | 8.2 | The employee requests to maintain an exercise plan type. | Trigger to start the maintain exercise plan type process. | When an employee wants to maintain an exercise plan details on the system | Employee | Requested by employee |
| 8.2 | The employee will select the edit exercise plan type option for the exercise plan type that they would like to edit. | Selected to proceed with the maintain exercise plan type process | When an employee wants to maintain an exercise plan details on the system | Employee | Selected by employee |
| 8.2 | The employee will update the exercise plan type and save it to the system. | Input to update details in the maintain exercise plan type process. | When an employee wants to maintain an exercise plan details on the system | Employee  **Exercise\_Plan\_Type\_Desc** | Entered by employee |
| Alt 8.2 | The employee selects the remove exercise plan type option | Trigger request to start removal of exercise plan type process. | When an employee wants to remove an exercise plan type details from the system. | Employee | Selected by employee |
| Alt 8.2 | The employee confirms that they would like to delete the selected exercise plan type. | Confirmation needed to confirm the removal of an exercise plan type. | When an employee wants to remove an exercise plan type details from the system. | Employee | Confirmed by employee |
| Alt 8.2 | The employee revokes the decision to remove the exercise plan type details from the system. | Confirmation needed to revoke the removal of the exercise plan type. | When an employee wishes to revoke removing an exercise plan type from the system. | Employee | Revoked by employee |
| 8.3 Search exercise plan type | 8.3 | The employee needs to search for an exercise plan type on the system. | Trigger to begin the search exercise plan type process. | When an employee wishes to search for a specific exercise plan type’s details on the system. | Employee | Requested by employee |
| 8.3 | Employee will enter any keyword or phrase that will match any exercise plan type on the system. | Input for search exercise plan type process. | When an employee wishes to search for a specific exercise plan type’s details on the system. | Employee  **Exercise\_Plan\_Type\_ID,**  **Exercise\_Plan\_Type\_Desc** | Entered by employee |
| 8.4 Add new exercise type | 8.4 | An employee requests to add a new exercise type onto the system | Trigger to start the add new exercise type process | When an employee wishes to add a new exercise type to the system | Employee | Requested by employee |
| 8.4 | Employee enters all information as requested by the system. | Input for add new exercise type process. | When an employee wishes to add a new exercise type to the system | Employee | Entered by employee |
| 8.4 | Employee confirms the correctness of the information entered. | Confirmation needed to complete the add new exercise type process. | When an employee wishes to add a new exercise type to the system | Employee | Confirmed by employee |
| Alt 8.4 | Employee does not confirm the information entered. | Revoke confirmation of correctness of details to allow for correction. | When an employee wishes to add a new exercise type to the system | Employee | Revoked by employee |
| 8.5 Maintain exercise type | 8.5 | The employee requests to maintain an exercise type. | Trigger to start the maintain exercise type process. | When an employee wants to maintain an exercise type details on the system | Employee | Requested by employee |
|  | 8.5 | The employee will select the edit exercise type option for the exercise type that they would like to edit. | Selected to proceed with the maintain exercise type process | When an employee wants to maintain an exercise type details on the system | Employee | Selected by employee |
|  | 8.5 | The employee will update the exercise type and save it to the system. | Input to update details in the maintain exercise type process. | When an employee wants to maintain an exercise type details on the system | Employee  **Exercise\_Type\_Desc** | Entered by employee |
|  | Alt 8.5 | The employee selects the remove exercise type option | Trigger request to start removal of exercise type process. | When an employee wants to remove an exercise type details from the system. | Employee | Selected by employee |
|  | Alt 8.5 | The employee confirms that they would like to delete the selected exercise type. | Confirmation needed to confirm the removal of an exercise type. | When an employee wants to remove an exercise type details from the system. | Employee | Confirmed by employee |
|  | Alt 8.5 | The employee revokes the decision to remove the exercise type details from the system. | Confirmation needed to revoke the removal of the exercise type. | When an employee wishes to revoke removing an exercise type from the system. | Employee | Revoked by employee |
| 8.6 Search exercise type | 8.6 | Employee requests to make a search for an exercise type. | Trigger to start the search exercise type process | When an employee wishes to search an exercise type on the system. | Employee | Requested by employee |
| 8.6 | Employee enters the details required in the search criteria | Input for the search exercise type process | When an employee wishes to search an exercise type on the system. | Employee | Entered by employee |
| 8.6 | Employee selects the specific exercise type they are looking for. | Needed to complete the search exercise type process. | When an employee wishes to search an exercise type on the system. | Employee | Selected by employee |
| 8.7 Add new sets | 8.7 | Employee wants to add a new set to the system | Trigger to start the add new set process | When an employee wishes to add new sets to the system. | Employee | Requested by employee |
| 8.7 | Employee will enter new set details on the system and save. | Input for the add new set process. | When an employee wishes to add new sets to the system. | Employee  **Set\_Description** | Entered by employee |
| 8.8 Maintain sets | 8.8 | The employee requests to maintain a set. | Trigger to start the maintain set process. | When an employee wants to maintain a set’s details on the system | Employee | Requested by employee |
| 8.8 | The employee will select the edit set option for the set that they would like to edit. | Selected to proceed with the maintain set process | When an employee wants to maintain a sets details on the system | Employee | Selected by employee |
| 8.8 | The employee will update the set and save it to the system. | Input to update details in the maintain set process. | When an employee wants to maintain a sets details on the system | Employee  **Set\_Description** | Entered by employee |
| Alt 8.8 | The employee selects the remove set option | Trigger request to start removal of set process. | When an employee wants to remove a sets details from the system. | Employee | Selected by employee |
| Alt 8.8 | The employee confirms that they would like to delete the selected set. | Confirmation needed to confirm the removal of a set. | When an employee wants to remove a sets details from the system. | Employee | Confirmed by employee |
| 8.9 Add new exercise | 8.9 | The employee requests to add a new exercise to the system | Initial request to start the add new exercise process. | When an employee wants to add a new exercise to the system. | Employee | Selected by employee |
| 8.9 | The employee will enter new exercise details | Input for the add new exercise process. | When an employee wants to add a new exercise to the system. | Employee  **Exercise\_Name,**  **Exercise\_Image, Exercise\_Description, Exercise\_Video,**  **Exercise\_Type\_ID** | Entered by employee |
| 8.10 Maintain exercise | 8.10 | The employee requests to maintain an exercise | Trigger to start the maintain exercise type process. | When an employee wants to maintain an exercise details on the system | Employee | Requested by employee |
|  | 8.10 | The employee will select the edit exercise type option for the exercise type that they would like to edit. | Selected to proceed with the maintain exercise type process | When an employee wants to maintain an exercise type details on the system | Employee | Selected by employee |
|  | 8.10 | The employee will update the exercise type and save it to the system. | Input to update details in the maintain exercise type process. | When an employee wants to maintain an exercise type details on the system | Employee  **Exercise\_Type\_Desc** | Entered by employee |
|  | Alt 8.10 | The employee selects the remove exercise type option | Trigger request to start removal of exercise type process. | When an employee wants to remove an exercise type details from the system. | Employee | Selected by employee |
|  | Alt 8.10 | The employee confirms that they would like to delete the selected exercise type. | Confirmation needed to confirm the removal of an exercise type. | When an employee wants to remove an exercise type details from the system. | Employee | Confirmed by employee |
|  | Alt 8.10 | The employee revokes the decision to remove the exercise type details from the system. | Confirmation needed to revoke the removal of the exercise type. | When an employee wishes to revoke removing an exercise type from the system. | Employee | Revoked by employee |
| 8.11 Search exercise | 8.11 | Employee requests to make a search for an exercise | Trigger to start the search exercise process | When an employee wishes to search an exercise on the system. | Employee | Requested by employee |
|  | 8.11 | Employee enters the details required in the search criteria | Input for the search exercise process | When an employee wishes to search an exercise on the system. | Employee  Exercise\_ID  Exercise\_Name | Entered by employee |
|  | 8.11 | Employee selects the specific exercise they are looking for. | Needed to complete the search exercise process. | When an employee wishes to search an exercise on the system. | Employee | Selected by employee |
| 8.12 Add new exercise plan | 8.12 | The employee requests to add a new exercise plan to the system | Initial request to start the add new exercise plan process. | When an employee wants to add a new exercise plan to the system. | Employee | Selected by employee |
| 8.12 | The employee will enter new exercise plan details | Input for the add new exercise plan process. | When an employee wants to add a new exercise plan to the system. | Employee  **ExercisePlan\_Name**, **ExercisePlan\_Description** and **ExercisePlan\_Price** | Entered by employee |
| 8.13 Maintain exercise plan | 8.13 | The employee requests to maintain an exercise plan. | Trigger to start the maintain exercise plan process. | When an employee wants to maintain an exercise plan details on the system | Employee | Requested by employee |
| 8.13 | The employee will select the edit exercise plan option for the exercise plan that they would like to edit. | Selected to proceed with the maintain exercise plan process | When an employee wants to maintain an exercise plan details on the system | Employee | Selected by employee |
| 8.13 | The employee will update the exercise plan and save it to the system. | Input to update details in the maintain exercise plan process. | When an employee wants to maintain an exercise plan details on the system | Employee  **ExercisePlan\_Name**, **ExercisePlan\_Description** and **ExercisePlan\_Price** | Entered by employee |
| Alt 8.13 | The employee selects the remove exercise plan option | Trigger request to start removal of exercise plan process. | When an employee wants to remove an exercise, plans details from the system. | Employee | Selected by employee |
| Alt 8.13 | The employee confirms that they would like to delete the selected exercise plan. | Confirmation needed to confirm the removal of an exercise plan. | When an employee wants to remove an exercise, plans details from the system. | Employee | Confirmed by employee |
| Alt 8.13 | The employee revokes the decision to remove the exercise plan details from the system. | Confirmation needed to revoke the removal of the exercise plan. | When an employee wishes to revoke removing an exercise plan from the system. | Employee | Revoked by employee |
| 8.14 Search exercise plan | 8.14 | The employee needs to search for an exercise plan on the system. | Trigger to begin the search exercise plan process. | When an employee wishes to search for a specific exercise plans details on the system. | Employee | Requested by employee |
| 8.14 | Employee will enter any keyword or phrase that will match any exercise plan on the system. | Input for search exercise plan process. | When an employee wishes to search for a specific exercise plans details on the system. | Employee  **Exercise\_Plan\_ID,**  **Exercise\_Plan\_Name** | Entered by employee |
| 8.14 | Employee will request to search. | Request to complete the search for the specific exercise plan. | When an employee wishes to search for a specific exercise plans details on the system. | Employee | Requested by employee |
| 8.15 Add exercise body type | 8.15 | The employee requests to add a new exercise body type to the system | Initial request to start the add new exercise body type process. | When an employee wants to add a new exercise body type to the system. | Employee | Selected by employee |
| 8.15 | The employee will enter new exercise body type details | Input for the add new exercise body type process. | When an employee wants to add a new exercise body type to the system. | Employee  **Exercise\_Body\_Type\_Desc** | Entered by employee |
| 8.16 Maintain exercise body type | 8.16 | The employee requests to maintain an exercise body type. | Trigger to start the maintain exercise body type process. | When an employee wants to maintain an exercise body type details on the system | Employee | Requested by employee |
|  | 8.16 | The employee will select the edit exercise body type option for the exercise body type that they would like to edit. | Selected to proceed with the maintain exercise body type process | When an employee wants to maintain an exercise body type details on the system | Employee | Selected by employee |
|  | 8.16 | The employee will update the exercise body type and save it to the system. | Input to update details in the maintain exercise body type process. | When an employee wants to maintain an exercise body type details on the system | Employee  **Exercise\_Body\_Type\_Desc** | Entered by employee |
|  | Alt 8.16 | The employee selects the remove exercise body type option | Trigger request to start removal of exercise body type process. | When an employee wants to remove an exercise body type details from the system. | Employee | Selected by employee |
|  | Alt 8.16 | The employee confirms that they would like to delete the selected exercise body type. | Confirmation needed to confirm the removal of an exercise body type. | When an employee wants to remove an exercise body type details from the system. | Employee | Confirmed by employee |
|  | Alt 8.16 | The employee revokes the decision to remove the exercise body type details from the system. | Confirmation needed to revoke the removal of the exercise type. | When an employee wishes to revoke removing an exercise type from the system. | Employee | Revoked by employee |
| 8.17 Search exercise body type | 8.17 | The employee needs to search for an exercise body type on the system. | Trigger to begin the search exercise body type process. | When an employee wishes to search for a specific exercise body types details on the system. | Employee | Requested by employee |
|  | 8.17 | Employee will enter any keyword or phrase that will match any exercise body type on the system. | Input for search exercise body type process. | When an employee wishes to search for a specific exercise body types details on the system. | Employee  **Exercise\_Body\_Type\_Desc**  Exercise\_Body\_Type\_ID | Entered by employee |
|  | 8.17 | Employee will request to search. | Request to complete the search for the specific exercise body type. | When an employee wishes to search for a specific exercise body types details on the system. | Employee | Requested by employee |
| 8.18 Add workout | 8.18 | The employee requests to add a new workout to the system | Initial request to start the add new workout process. | When an employee wants to add a new workout to the system. | Employee | Selected by employee |
| 8.18 | The employee will enter new work out details | Input for the add new work out process. | When an employee wants to add a new workout to the system. | Employee  **Workout\_Description** | Entered by employee |
| 8.19 Maintain workout | 8.19 | The employee requests to maintain a workout | Trigger to start the maintain a workout process. | When an employee wants to maintain a workout details on the system | Employee | Requested by employee |
| 8.19 | The employee will select the edit workout option for the workout that they would like to edit. | Selected to proceed with the maintain workout process | When an employee wants to maintain a workout, details on the system | Employee | Selected by employee |
| 8.19 | The employee will update the workout and save it to the system. | Input to update details in the maintain workout process. | When an employee wants to maintain a workout details on the system | Employee  **Workout\_Desc** | Entered by employee |
| Alt 8.19 | The employee selects the remove workout option | Trigger request to start removal of workout process. | When an employee wants to remove a workout details from the system. | Employee | Selected by employee |
| Alt 8.19 | The employee confirms that they would like to delete the selected workout. | Confirmation needed to confirm the removal of a workout. | When an employee wants to remove a workout details from the system. | Employee | Confirmed by employee |
| Alt 8.19 | The employee revokes the decision to remove the workout details from the system. | Confirmation needed to revoke the removal of the workout | When an employee wishes to revoke removing a workout from the system. | Employee | Revoked by employee |
| 8.20 Search workout | 8.20 | The employee needs to search for a workout on the system. | Trigger to begin the search workout process. | When an employee wishes to search for a specific work out details on the system. | Employee | Requested by employee |
| 8.20 | Employee will enter any keyword or phrase that will match any work out on the system. | Input for search work out process. | When an employee wishes to search for a specific work out details on the system. | Employee  **Workout\_ID,**  **Workout\_Description.** | Entered by employee |
| 8.20 | Employee will request to search. | Request to complete the search for the specific work out. | When an employee wishes to search for a specific work outs details on the system. | Employee | Requested by employee |
| 8.21 Add new exercise image | 8.21 | The employee requests to add a new exercise image to the system | Initial request to start the add new exercise image process. | When an employee wants to add a new exercise image to the system. | Employee | Selected by employee |
| 8.21 | The employee will enter exercise image details | Input for the add new exercise image process. | When an employee wants to add a new exercise image to the system. | Employee  Exercise\_Image\_URL | Entered by employee |
| 8.22 Remove exercise image | 8.22 | The employee requests to remove an exercise image from the maintain workout option. | Initial request to start remove exercise image process | When an employee wants to remove an exercise image from the system | Employee | Requested by employee |
| 8.22 | The employee will select to remove the exercise image option for the exercise image that they would like to remove. | Input selected for remove exercise plan process. | When an employee wants to remove an exercise image from the system | Employee  **Exercise\_Image\_URL** | Selected by employee |
| 8.22 | The employee will select to remove the workout and save changes on the system. | Finial selection to complete remove exercise type process | When an employee wants to remove an exercise image from the system | Employee | Selected by employee |
| 8.23 Search exercise image | 8.23 | The employee needs to search for an exercise image on the system. | Trigger to begin the search exercise image process. | When an employee wishes to search for a specific exercise image details on the system. | Employee | Requested by employee |
|  | 8.23 | Employee will enter any keyword or phrase that will match any exercise image on the system. | Input for search exercise image process. | When an employee wishes to search for a specific exercise images details on the system. | Employee  **Exercise\_Image\_ID,**  **Exercise\_Image\_URL.** | Entered by employee |
|  | 8.23 | Employee will request to search. | Request to complete the search for the specific exercise image. | When an employee wishes to search for a specific exercise images details on the system. | Employee | Requested by employee |
| 8.24 Create exercise plan | 8.24 | The employee requests to create a new exercise plan on the system | Initial request to start the create new exercise plan process. | When an employee wants to create a new exercise plan on the system. | Employee | Selected by employee |
| 8.24 | The employee will select the relevant details to be added to the new exercise plan | Input for the create new exercise plan process. | When an employee wants to create a new exercise on the system. | Employee  **Workout\_IDs.** | Entered by employee |
| 8.25 Create workout | 8.25 | The employee requests to create a new workout on the system | Initial request to start the create new workout process. | When an employee wants to create a new workout on the system. | Employee | Selected by employee |
|  | 8.25 | The employee will select the relevant details to be added to the new workout | Input for the create new exercise plan process. | When an employee wants to create a new exercise on the system. | Employee  **Exercise\_ID,**  **Set\_ID.** | Entered by employee |

Conclusion:

This section outlines the system interfaces and input data depicted in the context diagram (**See section 3.1**) for the implementation of the Jesster FIT system. The section went into detail discussing the data used in processes, description of data, purpose of input data, when the data will be used, the entities and attributes associated with entered data and the logical layout.

7. Reports and other Outputs

Introduction

The following section constructs every single output generated by the Jesster FIT system which is previously depicted in the context diagram (**see section 3.1**). A table is created which describes the purpose of every output, when outputs will be used, entities and attributes involved and the logical layout of the output. Outputs to external applications are also included in this section.

Table 7.1 (Reports and Outputs)

| Use Case Number | Flow Line | Description | Purpose | When it will be produced | Entities and attributes | Logical layout |
| --- | --- | --- | --- | --- | --- | --- |
| 1. Admin Subsystem | | | | | | |
| 1.1 Add new subscription type | 1.1 | The system will display the add new subscription type screen and inform the employee to enter all the required subscription type details such as. | Informs the employee what subscription type details are required. | When employee requests to add a new subscription type to the system. | **Subscription\_Type\_Description**  **Subscription\_Type\_Price.** | Information displayed to the user on the screen. |
| 1.1 | The system will display a success message. | To make the employee aware of the success of the add new subscription type process. | When an employee submits required subscription type details. | None | Notification message displayed to user about successful subscription type addition. |
| Alt 1.1 | The entered subscription type details are invalid. | To make the user aware that subscription type details entered are not correct | When an employee enters invalid subscription type details. | None | Error message displayed to user stating that subscription type details entered are invalid. |
| Alt 1.1 | Subscription type details already exist on the system. | To make user aware that the entered subscription type details already exist on the system | When an employee enters details that are already on the system. | None | Error message displayed to user stating that subscription type details entered already exist on the system. |
| 1.2 Maintain subscription type | 1.2 | The system will display the maintain subscription type page and prompt the employee to search for an existing subscription type on the system. | To take the user to the maintain subscription type page. | When an employee is using the system to maintain a subscription type. | None | Information displayed to the user on the screen. |
| 1.2 | The system will invoke use case 1.3 Search subscription type and display all results to the user. | To return a list of found subscription type matching the search criteria and display it to user. | When an employee is using the system to maintain a subscription type’s details. | **Subscription\_Type\_Description**  **Dubscription\_Type\_ID** | Information displayed to the user on the screen. |
| 1.2 | The system will display a success message. | To make the employee aware of the success of the maintain subscription type process. | When an employee submits the maintained subscription type details. | None | Notification message displayed to user about successful subscription type maintenance. |
| Alt 1.2 | The system prompts the user to confirm if they would like to delete the selected subscription type. | To confirm if the user wants to remove the subscription types’ details from the system | When an employee selects to remove a subscription types’ details | None | Notification message displayed requesting if they want to continue with removing the subscription type from the system. |
| 1.3 Search subscription type | 1.3 | The system prompts the owner to search criteria for the subscription type. | To make the owner aware of the details needed to be used to search for a subscription type | When owner requests to search for a specific subscription type on the system. | **Subscription\_Type\_ID**  **Subscription\_Type\_Description** | Information displayed to the user on the screen. |
| 1.3 | The system displays the details of the selected subscription ty pe | To show the employee the results of the searched details. | When an employee searches by criteria |  | Information displayed to the user on the screen. |
| Alt 1.3 | The system displays a message saying no results found matching the entered criteria | To inform the employee that no results were found on the system. | When an employee searches by criteria |  | Error message notifying employee no details matched searched criteria. |
| Alt 1.3 | The captured keyword or phrase is invalid | To inform the employee that the entered search criteria details are invalid. | When an employee searches by criteria |  | Error message notifying employee that entered search details are invalid. |
| 1.4 Add new challenge | 1.4 | System prompts the employee to enter the required details. | To ensure the employee knows to enter all details. | When an employee is using the system to add a new challenge. | Employee  **Challenge\_Start\_Date** and **Challenge\_End\_Date** | Information displayed to the user on the screen. |
| 1.4 | The system will display a success message. | To make the employee aware of the success of the add new challenge process. | When an employee submits required challenge details. | None | Notification message displayed to user about successful challenge addition. |
| Alt 1.4 | The entered challenge details are invalid. | To make the user aware that challenge details entered are not correct | When an employee enters invalid challenge details. | None | Error message displayed to user stating that challenge details entered are invalid. |
| Alt 1.4 | Challenge details already exist on the system. | To make user aware that the entered challenge details already exist on the system | When an employee enters details that are already on the system. | None | Error message displayed to user stating that challenge details entered already exist on the system. |
| 1.5 Maintain challenge | 1.5 | The system will display the maintain challenge page and prompt the employee to search for an existing challenge on the system. | To take the user to the maintain challenge page. | When an employee is using the system to maintain a challenge | None | Information displayed to the user on the screen. |
| 1.5 | The system will invoke use case 1.6 Search Challenge and display all results to the user. | To return a list of found challenges matching the search criteria and display it to user. | When an employee is using the system to maintain a challenge | **Challenge\_ID, Challenge\_Start\_Date Challenge\_End\_Date** | Information displayed to the user on the screen. |
| 1.5 | The system will display a success message. | To make the employee aware of the success of the maintain challenge process. | When an employee submits the maintained challenge details. | None | Notification message displayed to user about successful challenge maintenance. |
| Alt 1.5 | The system prompts the user to confirm if they would like to delete the selected challenge. | To confirm if the user wants to remove the challenge details from the system | When an employee selects to remove a challenges’ details | None | Notification message displayed requesting if the want to continue with removing the challenge from the system. |
| 1.6 Search challenge | 1.6 | The system returns a list of the challenge(s) matching the given search criteria. | To display the returned search results to the user. | When an employee uses the system to search a challenge. | **Challenge\_ID, Challenge\_Start\_Date Challenge\_End\_Date** | Information displayed to the user on the screen. |
| Alt 1.6 | The captured keyword or phrase is invalid. | To make the user aware that challenge details searched are invalid. | When an employee searches an invalid challenge detail. | None | Error message displayed to user saying search criteria entered is invalid. |
| Alt 1.6 | No values on the system match the given search criteria. The system will display an error message. | To make the user aware that challenge details searched do not exist on the system | When an employee searches for a challenge that does not exist on the system. | None | Error message displayed to user saying search criteria didn’t match any record in the database. |
| 1.7 Add personal training booking slots | 1.7 | System prompts the user to select a date. | Make user aware that a date needs to be selected to continue with the add personal training booking slots process. | When a user is adding a new Personal training booking slot to the system | None | Information displayed to the user on the screen |
| 1.7 | The system prompts the employee to enter personal trainer details. | Make user aware that personal trainer details need to be added to continue with the add personal training booking slots process | When a user is adding a new Personal training booking slot to the system | None | Information displayed to the user on the screen |
| 1.7 | System makes the user aware of available dates not allocated and prompts the user if he wishes to add more booking slots or complete the use case. | Makes user aware of available dates not allocated. | When a user is adding a new Personal training booking slot to the system | None | Message displayed making user aware of available slots still available for booking. |
| 1.7 | Success message is displayed to employee and the employee is returned to the calendar page allowing for further personal training booking slots to be added if needed | Makes user aware that all personal training booking slots have been added successfully to the system | When a user saves new Personal training booking slot to the system | None | Message notification making user aware that personal training booking slots added have been successfully saved. |
| Alt 1.7 | Personal trainer information entered was invalid or incomplete | Makes user aware that personal training booking slots entered are invalid. | When a user saves new Personal training booking slot to the system | None | Error message making user aware that entered personal training booking slot details are invalid or incomplete. |
| 1.8 Add challenge consultation booking slots | 1.8 | System prompts the user to select a date. | Make user aware that a date needs to be selected to continue with the add challenge booking slots process. | When a user is adding a new challenge consultation booking slot to the system | None | Information displayed to the user on the screen |
| 1.8 | The system prompts the employee to enter personal trainer details. | Make user aware that personal trainer details need to be added to continue with the add challenge consultation booking slots process | When a user is adding a new challenge consultation booking slot to the system | None | Information displayed to the user on the screen |
| 1.8 | System makes the user aware of available dates not allocated and prompts the user if he wishes to add more booking slots or complete the use case. | Makes user aware of available dates not allocated. | When a user is adding a new challenge consultation booking slot to the system | None | Message displayed making user aware of available slots still available for booking. |
| 1.8 | Success message is displayed to employee and the employee is returned to the calendar page allowing for further challenge consultation booking slots to be added if needed | Makes user aware that all challenge consultation booking slots have been added successfully to the system | When a user saves new challenge consultation booking slot to the system | None | Message notification making user aware that challenge consultation booking slots added have been successfully saved. |
| Alt 1.8 | Personal trainer information entered was invalid or incomplete | Makes user aware that challenge consultation booking slots entered are invalid. | When a user saves new challenge consultation booking slot to the system | None | Error message making user aware that entered challenge consultation booking slot details are invalid or incomplete. |
| 1.9 Confirm payment | 1.9 | The system will display a confirmation message on the screen and notify the customer/ client via email. | Make user aware that confirm payment process was completed successfully | When a payment needs confirmation | None | Message notifying user that payment was confirmed successfully. |
| Alt 1.9 | The payment is invalid. The system will display an error message. | Make user aware that confirmation of payment was invalid. | When a payment needs to be confirmed. | None | Error message invalid payment confirmation. |
| Alt 1.9 | The system is unable to confirm the payment | Make employee aware that the system is unable to confirm a payment. | When a payment needs to be confirmed | None | Error message notifying employee that the system was unable to confirm the payment. |
| 1.10 Search subscription | 1.10 | System prompts the user to enter a phrase or keyword. | Make user aware of what details needs to be added to search. | When an employee needs to search for a subscription on the system. | **Subscription\_ID**  **Subscription\_Type\_ID** | Information displayed to the user on the screen |
| 1.10 | The system returns a list of the subscription(s) matching the given search criteria. | Show user results that have been returned from entered search criteria. | After an employee has requested to search entered details. |  | Information displayed to the user on the screen |
| ALT1.10 | The captured keyword or phrase is invalid | Make user aware that search criteria entered is invalid. | After an employee has submitted search request. | None | Error message informing user of invalid search criteria. |
| Alt 1.10 | No values on the system match the given search criteria. The system will display an error message. | To make the user aware that subscription searched do not exist on the system | When an employee searches for a subscription that does not exist on the system. | None | Error message displayed to user saying search criteria didn’t match any record in the database. |
| 1.11 Search personal training booking | 1.11 | System prompts the user to enter a phrase or keyword. | Make user aware of what details needs to be added to search. | When an employee needs to search for a personal training booking on the system. | **Client\_ID, Employee\_ID, Training\_Date\_Booked, Training\_Venue,** | Information displayed to the user on the screen |
| 1.11 | The system returns a list of the personal training booking(s) matching the given search criteria. | Show user results that have been returned from entered search criteria. | After an employee has requested to search entered details. | **None** | Information displayed to the user on the screen |
| ALT1.11 | The captured keyword or phrase is invalid | Make user aware that search criteria entered is invalid. | After an employee has submitted search request. | None | Error message informing user of invalid search criteria. |
| Alt 1.11 | No values on the system match the given search criteria. The system will display an error message. | To make the user aware that personal training booking searched does not exist on the system | When an employee searches for a personal training booking that does not exist on the system. | None | Error message displayed to user saying search criteria didn’t match any record in the database. |
| 1.12 Search challenge consultation booking | 1.12 | System prompts the user to enter a phrase or keyword. | Make user aware of what details needs to be added to search. | When an employee needs to search for a challenge consultation booking on the system. | **Client\_ID, Employee\_ID, Training\_Date\_Booked, Training\_Venue, Consultation\_ID** | Information displayed to the user on the screen |
| 1.12 | The system returns a list of the challenge consultation booking(s) matching the given search criteria. | Show user results that have been returned from entered search criteria. | After an employee has requested to search entered details. | **None** | Information displayed to the user on the screen |
| ALT1.12 | The captured keyword or phrase is invalid | Make user aware that search criteria entered is invalid. | After an employee has submitted search request. | None | Error message informing user of invalid search criteria. |
| Alt 1.12 | No values on the system match the given search criteria. The system will display an error message. | To make the user aware that challenge consultation booking searched does not exist on the system | When an employee searches for a challenge consultation booking that does not exist on the system. | None | Error message displayed to user saying search criteria didn’t match any record in the database. |
| 1.13 Maintain company | 1.13 | The system will display the maintain company page and prompt the employee to select a detail to update. | To inform the employee on what details need to be requested to maintain the company. | When an employee requests to maintain the company’s details. | **Company\_Name, Company\_Address, Company\_RegisterNo, Company\_Email, Company\_Contact\_No, Account\_Name, Account\_No, Account\_Type, Account\_Branch, Account\_Branch\_Code**. | Information displayed to the user on the screen. |
|  | 1.13 | The system will display a success message. | To make the employee aware of the success of the maintain company process. | When an employee submits the maintained company details. | None | Notification message displayed to user about successful company maintenance. |
|  | Alt 1.5 | The system prompts the user to confirm if they would like to delete the selected challenge. | To confirm if the user wants to remove the challenge details from the system | When an employee selects to remove a challenges’ details | None | Notification message displayed requesting if the want to continue with removing the challenge from the system. |
| 1. Product Subsystem | | | | | | |
| 2.1 Add new product type | 2.1 | The system will display the add new product type screen and prompt the employee to enter the new product type details | Make employee aware of what details need to be added. | When an employee requests to add a new product type to the system. | **Product\_Type\_Description** | Information displayed to the user on the screen |
| 2.1 | The system will display a success message. | To make the employee aware of the success of the add new product type process. | When an employee submits required product type details. | None | Notification message displayed to user about successful product type addition. |
| Alt 2.1 | The entered product type details are invalid. | To make the user aware that product type details entered are not correct | When an employee enters invalid product typedetails. | None | Error message displayed to user stating that product type details entered are invalid. |
| Alt 2.1 | Product type details already exist on the system. | To make user aware that the entered product type details already exist on the system | When an employee enters details that are already on the system. | None | Error message displayed to user stating that product type details entered already exist on the system. |
| 2.2 Maintain product type | 2.2 | The system will display the maintain product type page and prompt the employee to search for an existing product type on the system. | To take the user to the maintain product type page. | When an employee is using the system to maintain a product type | None | Information displayed to the user on the screen. |
| 2.2 | The system will invoke use case 2.3 Search Product type and display all results to the user. | To return a list of found product types matching the search criteria and display it to user. | When an employee is using the system to maintain a product type | **Product\_Type\_Description** | Information displayed to the user on the screen. |
| 2.2 | The system will display a success message. | To make the employee aware of the success of the maintain product type process. | When an employee submits the maintained product type details. | None | Notification message displayed to user about successful product type maintenance. |
| Alt 2.2 | The system prompts the user to confirm if they would like to delete the selected product type. | To confirm if the user wants to remove the product type details from the system | When an employee selects to remove a product types’ details | None | Notification message displayed requesting if the user wants to continue with removing the product type from the system. |
| 2.3 Search product type | 2.3 | System prompts the user to enter a phrase or keyword. | Make user aware of what details needs to be added to search. | When an employee needs to search for a product type on the system. | **Product\_Type\_Description** | Information displayed to the user on the screen |
| 2.3 | The system returns a list of the product type(s) matching the given search criteria. | Show user results that have been returned from entered search criteria. | After an employee has requested to search entered details. |  | Information displayed to the user on the screen |
| ALT2.3 | The captured keyword or phrase is invalid | Make user aware that search criteria entered is invalid. | After an employee has submitted search request. | None | Error message informing user of invalid search criteria. |
| Alt 2.3 | No values on the system match the given search criteria. The system will display an error message. | To make the user aware that product type details searched do not exist on the system | When an employee searches for a product type that does not exist on the system. | None | Error message displayed to user saying search criteria didn’t match any record in the database. |
| 2.4 Add new product | 2.4 | The system will display the add new product screen and prompt the employee to enter the new product details | Make employee aware of what details need to be added. | When an employee requests to add a new product to the system. | **Product\_Name**, **Product\_Description**, **Product\_Price,Product\_Quantity, Product\_Type** | Information displayed to the user on the screen |
| 2.4 | The system will display a success message. | To make the employee aware of the success of the add new product process. | When an employee submits required product details. | None | Notification message displayed to user about successful product addition. |
| Alt 2.4 | The entered product details are invalid. | To make the user aware that product details entered are not correct | When an employee enters invalid product details. | None | Error message displayed to user stating that product details entered are invalid. |
| Alt 2.4 | Product details already exist on the system. | To make user aware that the entered product details already exist on the system | When an employee enters details that are already on the system. | None | Error message displayed to user stating that product details entered already exist on the system. |
| 2.5 Maintain product | 2.5 | The system will display the maintain product page and prompt the employee to search for an existing product on the system. | To take the user to the maintain product page. | When an employee is using the system to maintain a product | None | Information displayed to the user on the screen. |
| 2.5 | The system will invoke use case 2.6 Search Product and display all results to the user. | To return a list of found products matching the search criteria and display it to user. | When an employee is using the system to maintain a product | **Product\_Name**, **Product\_Description**, **Product\_Price,Product\_Quantity, Product\_Type** | Information displayed to the user on the screen. |
| 2.5 | The system will display a success message. | To make the employee aware of the success of the maintain product process. | When an employee submits the maintained product details. | None | Notification message displayed to user about successful product maintenance. |
| Alt 2.5 | The system prompts the user to confirm if they would like to delete the selected product. | To confirm if the user wants to remove the product details from the system | When an employee selects to remove a products’ details | None | Notification message displayed requesting if the user wants to continue with removing the product from the system. |
| 2.6 Search product | 2.6 | System prompts the user to enter a phrase or keyword. | Make user aware of what details needs to be added to search. | When an employee needs to search for a product on the system. | **Product\_Name**, **Product\_Description**, **Product\_Price,Product\_Quantity, Product\_Type** | Information displayed to the user on the screen |
| 2.6 | The system returns a list of the product(s) matching the given search criteria. | Show user results that have been returned from entered search criteria. | After an employee has requested to search entered details. |  | Information displayed to the user on the screen |
| ALT2.6 | The captured keyword or phrase is invalid | Make user aware that search criteria entered is invalid. | After an employee has submitted search request. | None | Error message informing user of invalid search criteria. |
| Alt 2.6 | No values on the system match the given search criteria. The system will display an error message. | To make the user aware that product details searched do not exist on the system | When an employee searches for a product that does not exist on the system. | None | Error message displayed to user saying search criteria didn’t match any record in the database. |
| 2.7 Add new supplier | 2.7 | The system will display the add new supplier screen and prompt the employee to enter the new supplier details | Make employee aware of what details need to be added. | When an employee requests to add a new supplier to the system. | **Supplier\_Description,**  **Supplier\_Email,**  **Supplier\_Phone,**  **Supplier\_Address,**  **Supplier\_Contact.** | Information displayed to the user on the screen |
| 2.7 | The system will display a success message. | To make the employee aware of the success of the add new supplier process. | When an employee submits required supplier details. | None | Notification message displayed to user about successful supplier addition. |
| Alt 2.7 | The entered supplier details are invalid. | To make the user aware that supplier details entered are not correct | When an employee enters invalid supplier details. | None | Error message displayed to user stating that product type details entered are invalid. |
| 2.8 Maintain supplier | 2.8 | The system will display the maintain supplier page and prompt the employee to search for an existing supplier on the system. | To take the user to the maintain supplier page. | When an employee is using the system to maintain a supplier. |  | Information displayed to the user on the screen. |
| 2.8 | The system will invoke use case 2.9 Search supplier and display all results to the user. | To return a list of found suppliers matching the search criteria and display it to user. | When an employee is using the system to maintain a supplier | **Supplier\_Description,**  **Supplier\_Email,**  **Supplier\_Phone,**  **Supplier\_Address,**  **Supplier\_Contact.** | Information displayed to the user on the screen. |
| 2.8 | The system will display a success message. | To make the employee aware of the success of the maintain supplier process. | When an employee submits the maintained supplier details. |  | Notification message displayed to user about successful supplier maintenance. |
| Alt 2.8 | The system prompts the user to confirm if they would like to delete the selected supplier. | To confirm if the user wants to remove the supplier details from the system | When an employee selects to remove a suppliers’ details |  | Notification message displayed requesting if the user wants to continue with removing the supplier from the system. |
| Alt 2.8 | The user confirms that they would like to delete the selected supplier. | To complete the supplier removal process. | When an employee selects to remove a suppliers’ details |  | Success Message notifying user of successful supplier details removal |
| Alt 2.8 | The updated details are invalid. | The inform the employee that the entered supplier maintenance details are invalid | When an owner submits the supplier maintenance details. |  | Error message notifying owner that maintained e details are invalid. |
| 2.9 Search supplier | 2.9 | System prompts the user to enter a phrase or keyword. | Make user aware of what details needs to be added to search. | When an employee needs to search for a supplier on the system. | **Supplier\_ID,**  **Supplier\_Description,**  **Supplier\_Email,** | Information displayed to the user on the screen |
| 2.9 | The system returns a list of the supplier(s) matching the given search criteria. | Show user results that have been returned from entered search criteria. | After an employee has requested to search entered details. |  | Information displayed to the user on the screen |
| ALT2.9 | The captured keyword or phrase is invalid | Make user aware that search criteria entered is invalid. | After an employee has submitted search request. | None | Error message informing user of invalid search criteria. |
| Alt 2.9 | No values on the system match the given search criteria. The system will display an error message. | To make the user aware that product details searched do not exist on the system | When an employee searches for a product that does not exist on the system. | None | Error message displayed to user saying search criteria didn’t match any record in the database. |
| 2.10 Add new product order | 2.10 | The system will notify an employee that a product’s quantity level has dropped below a pre-defined value | Notify employee when stock levels are low. | When stock levels drop below a pre-determined value. | none | System notifies employee when product levels have dropped below pre-determined value. |
| 2.10 | The system will display the add new product order screen and prompt the user to enter the product order details**.** | Make user aware of product order details needing to be completed. | When product levels are running low and employee request to add new supplier order. | **Supplier\_ID, Product\_ID, Ordered\_Quantity, Ordered\_Amount, Date\_Ordered** | Information displayed to the user on the screen |
| 2.10 | The system will notify the supplier and the owner of the order. | Make the supplier and owner aware of the order | When a new order is product order is placed |  | Notification to owner to ensure he is aware of the new product order. |
| 2.10 | The system will calculate the total amount of the order | Make user aware of the total amount | When a new order is product order is placed |  | Information displayed to the user on the screen |
| 2.10 | The system will display a success message to the employee and return the employee to the product page. | Make user aware of the success of the add new product order process | When a new order is product order is placed |  | Success message informing employee that product order was successfully placed. |
| ALT 2.10 | The entered product order details are invalid. | To make the user aware that product ordered details entered are not correct | When an employee enters invalid product order details. | None | Error message displayed to user stating that product order details entered are invalid. |
| Alt 2.10 | the system is unable to notify the supplier of the product order. | To make the user aware that the supplier has not been notified of the order. | When an employee submits product order details. | None | Error message informing employee that supplier has not been made aware of the product order. |
| 2.11 Pay supplier | 2.11 | The system invokes **Use Case 2.9 search supplier order** and calculates the amount to be paid | To search a sale order that has been placed. | When an employee wants to search for order on the system | **Order\_ID** | Information displayed to the user on the screen |
| 2.11 | The system will notify the supplier  and the owner of the recent payment, | To notify the supplier and owner of recent payment. | When an employee wants to search for order on the system |  | Display success message to the employee notifying them of the successful payment to supplier. |
| 2.12 Confirm supplier order | 2.12 | The system will display the details related to the order ID. | To show the details for the order ID searched. | When an employee submits the order ID to be searched. |  | Information displayed to the user on the screen |
| 2.12 | System will display a success message | To show the employee the employee that the confirm supplier order | When an employee submits product order. |  | Message notifying employee on the success of the confirmation of payment. |
| Alt 2.13 | The **Order\_ID** does not exist on the system. | To notify the supplier that the order does not exist on the system. | When an employee searched for an order on the system. |  | Error message to notify the supplier2.13 that the order does not exist on the system. |
| Alt 2.13 | The supplier order received does not match the supplier order placed | To notify employee to allow them to notify the supplier that the order received does not match the supplier order | When an employee searched for a supplier order on the system. |  | Notify the employee to notify the supplier of the mistake. |
| ALT 2.13 | Employee notifies the supplier of the error | To notify supplier of the error in the order placed and the order received. | When an employee got notified to notify the supplier about the error. |  | Notify the supplier that the supplier order received does not match the supplier order placed. |
| 1. User Subsystem | | | | | | |
| 3.1 Login | 3.1 | The system displays the login page and prompts the user for their login details | To make user aware of what login details are needed from | When a user wants to login into the system. | Username  Password | Information displayed to the user on the screen |
| 3.1 | The system logs the user into the system and navigates to the home page | To inform the employee that have successfully logged in. | When a submits login details. |  | Login and take user to the home page. |
| 3.1 | The user login details do not exist in the system, display error. | To inform the user that their details do not exist on the system. | When a user wants to log into the system. |  | Error message displayed to inform user that their details do not exist on the system. |
| 3.2 Update password | 3.2 | The system requests the user to enter their current password | To inform user that they need to enter their current password. | When a user wants to update their password. | Password | Information displayed to the user on the screen |
| 3.2 | The system requests the user to input a new password. | To inform the user that they need to enter their new password | When a user submits their old password. | New Password | Information displayed to the user on the screen |
| 3.2 | The system displays a confirmation message that the user has successfully updated their password. | To inform the user that their password has successfully been updated. | When user submits updated password |  | Success message to inform the user that their password has successfully been updated. |
| Alt 3.2 | The user has forgotten their password  Invoke use case **3.3 Forgotten Password** | To inform the user that they need to proceed to 3.3 Forgotten password because their current password was entered incorrectly. | When a user wants to update their password but can’t remember current password. | none | Error message displayed to inform user they need to proceed to forgotten password screen. |
| Alt 3.2 | The user password details do not match the data in the **Employee** or **Client table**, display error. | To inform the user that the details they have entered do not match any details in the employee or client table | When a user tries to login into the system, but details do not exist on the system. | **Employee** or **Client table**, | Error message login details entered are do not match any details in the database. |
| Alt 3.2 | The new password matches old password, display error. | To inform user that the new updated password is the same as the old password. | When user enters an updated password that’s the same as the old password. |  | Error message to notify user that |
| 3.3 Forgotten password | 3.3 | The system prompts the user to enter their email address | To inform the user to enter their email address to recover their forgotten password on. | When a user clicks forgotten password. | Employye\_Email\_Address  or  Client\_Email\_Address | Information displayed to the user on the screen |
| 3.3 | The system displays a confirmation message that the user has successfully updated their password. | To inform user that the forgotten password has been successfully updated | When a user submits their new password. | Client\_Paasword  Or  Employee\_Password | Success message notifying user of successful update of forgotten password. |
| Alt 3.3 | The email does not match any data in the database. | To inform the user that their email address did not match any email in the database | When the user submits their email address to recover forgotten password. | Employee\_Email\_Address  or  Client\_Email\_Address | Error message notifying the user that the email address they entered does not exist on the system. |
| 3.4 Log out | 3.4 | The system prompts the user to confirm log out. | To inform the user to confirm logout. | When a user wants to log out of the system. |  | Information displayed to the user on the screen |
| 3.4 | The system redirects the user to the home page. | To take employee tack to the employee home page. | When a user confirms logout. |  | Successful logout, user returned to the user home page |
| Alt 3.4 | The user selects to cancel log out | To cancel logout process. | When a user wants to cancel a logout process. |  | Cancel logout and revert user back to home page |
| 3.5 Add access level | 3.5 | The system prompts the employee to enter the information required details. | To inform the user to enter the required access level details. | When a user selects to add a new access level to the system. | **Access\_Description** | Information displayed to the user on the screen. |
| 3.5 | The system will display a success message on the screen and navigate the employee to the admin screen | To inform the employee that the access levels have been added successfully to the database | When the employee submits the new entered access levels. |  | Success message notifying the employee of the saved access level details. |
| Alt 3.5 | The information entered fails the validation test of the data format. The system displays a message stating that the data format is not correct | To inform the employee that access level details are invalid. | When an employee submits new access levels details. |  | Error message notifying the user aware of the user of invalid details. |
| Alt 3.5 | The details entered already exists | To inform the employee that the access level details already exist in the database | When an employee submits new access levels details. |  | Error message notifying user the system already has a access level matching the entered details on the system. |
| 3.6 Maintain access level | 3.6 | The system Invokes UC 3.7  Search Access Level and display the  relevant access levels. | To show the returned access level whose details were searched. | When an employee searches for a access levels details on the system. |  | Information displayed to the user on the screen. |
| 3.6 | The system displays the maintain access Level screen. In which the details of the access Level are editable. | To allow the user to enter details maintain the access levels. | When an employee selects maintain access level for the specific search. | **Access\_Description** | Information displayed to the user on the screen. |
| 3.6 | The system displays a message stating the that table has been updated successfully and the employee is navigated to the admin page | To inform the employee that the maintained access levels details have been saved. | When the employee submits the maintained access levels to the system. |  | Success message displayed notifying the employee of the successful changes to the maintained access level. |
| Alt 3.6 | The system prompts to confirm the removal. | To inform the employee that they have to confirm the removal before it can happen. | When an employee selects to remove an access levels detail from the system |  | Message will be displayed to employee to confirm removal. |
| Alt 3.6 | The system displays a confirmation message and navigates the employee to the admin page | To inform the employee that the removal of the access level was a success | When an employee confirms the removal of an access level. |  | Confirmation message notifying employee of successful removal of access level. |
| 3.7 Search access level | 3.7 | The system displays the search criteria for the access level | To make the employee aware of the details needed to be used to search for an access level | When employee requests to search for a specific access level on the system. | **Access\_Level**  **Access\_Description** | Information displayed to the user on the screen. |
| 3.7 | The system displays the details of the selected access level | To show the employee the results of the searched details. | When an employee searches by criteria |  | Information displayed to the user on the screen. |
| Alt 3.7 | The system displays a message saying no results found matching the entered criteria | To inform the employee that no results were found on the system. | When an employee searches by criteria |  | Error message notifying employee no details matched searched criteria. |
| 3.8 Add functionality | 3.8 | The system prompts the employee to enter the information required details. | To inform the user to enter the required access level details. | When a user selects to add a new access level to the system. | **Function\_Description** | Information displayed to the user on the screen. |
| 3.8 | The system will display a success message on the screen and navigate the employee to the admin screen | To inform the employee that the functionality has been added successfully to the database | When the employee submits the new entered functionality details. |  | Success message notifying the employee of the saved functionality details. |
| Alt 3.8 | The information entered fails the validation test of the data format. The system displays a message stating that the data format is not correct | To inform the employee that functionality details are invalid. | When an employee submits new functionality details. |  | Error message notifying the user aware of the user of invalid details. |
| Alt 3.8 | The details entered already exists | To inform the employee that the functionality details already exist in the database | When an employee submits new functionality details. |  | Error message notifying user the system already has a functionality matching the entered details on the system. |
| 3.9 Maintain functionality | 3.9 | The system Invokes UC 3.10  Search Functionality and display the  relevant functionality details. | To show the returned functionality whose details were searched. | When an employee searches for a functionalities details on the system. |  | Information displayed to the user on the screen. |
| 3.9 | The system displays the maintain functionality screen. In which the details of the functionality are editable. | To allow the user to enter details maintain the functionality. | When an employee selects maintain functionality for the specific search. | **Functionality\_Description** | Information displayed to the user on the screen. |
| 3.9 | The system displays a message stating the that table has been updated successfully and the employee is navigated to the admin page | To inform the employee that the maintained functionality details have been saved. | When the employee submits the maintained functionality details to the system. |  | Success message displayed notifying the employee of the successful changes to the maintained functionality. |
| Alt 3.9 | The system prompts to confirm the removal. | To inform the employee that they must confirm the removal before it can happen. | When an employee selects to remove a functionality’s details from the system |  | Message will be displayed to employee to confirm removal. |
| Alt 3.9 | The system displays a confirmation message and navigates the employee to the admin page | To inform the employee that the removal of the functionality was a success | When an employee confirms the removal of a functionality. |  | Confirmation message notifying employee of successful removal of functionality details. |
| Alt 3.9 | The information entered fails the validation test of the data format. The system displays a message stating that the data format is not correct | To inform the employee that the maintained functionality details are invalid. | When an employee submits the maintained functionality details. |  | Error message notifying employee that the maintained functionality details are invalid. |
| 3.10 Search functionality | 3.10 | The system displays the search criteria for the functionality | To make the employee aware of the details needed to be used to search for a functionality | When employee requests to search for a specific functionality on the system. | **Function\_ID**  **Function\_Description** | Information displayed to the user on the screen. |
| 3.10 | The system displays the details of the selected functionality | To show the employee the results of the searched details. | When an employee searches by criteria |  | Information displayed to the user on the screen. |
| Alt 3.10 | The system displays a message saying no results found matching the entered criteria | To inform the employee that no results were found on the system. | When an employee searches by criteria |  | Error message notifying employee no details matched searched criteria. |
| 1. Employee Subsystem | | | | | | |
| 4.1 Add new employee | 4.1 | System prompts the owner to enter the required details. | To ensure the owner knows to enter all details. | When an owner is using the system to add a new employee | **Employee\_Name**, **Employee\_Surname**, **Employee\_Email**, **Employee\_Address**, **Employee\_Contact, Employee\_Password, Access\_Level, Employee\_ID\_Number** | Information displayed to the user on the screen. |
| 4.1 | The system will display a success message. | To make the owner aware of the success of the add new employee process. | When an owner submits required employee details. |  | Notification message displayed to user about successful employee addition. |
| Alt 4.1 | The entered employee details are invalid. | To make the user aware that employee details entered are not correct | When an owner enters invalid employee details. |  | Error message displayed to user stating that challenge details entered are invalid. |
| Alt 4.1 | Employee details already exist on the system. | To make user aware that the entered challenge details already exist on the system | When an owner enters details that are already on the system. |  | Error message displayed to user stating that employee details entered already exist on the system. |
| 4.2 Maintain employee | 4.2 | The system will display the maintain employee page and prompt the owner to search for an existing employee on the system. | To take the user to the maintain employee page. | When an owner is using the system to maintain an employee. |  | Information displayed to the user on the screen. |
| 4.2 | The system will invoke use case 4.3 Search employee and display all results to the user. | To return a list of found employees matching the search criteria and display it to user. | When an owner is using the system to maintain an employee | **Employee\_Name**, **Employee\_Surname**, **Employee\_Email**, **Employee\_Address**, **Employee\_Contact, Employee\_Password, Access\_Level, Employee\_ID\_Number** | Information displayed to the user on the screen. |
| 4.2 | The system will display a success message. | To make the owner aware of the success of the maintain employee process. | When an owner submits the maintained employee details. |  | Notification message displayed to user about successful employee maintenance. |
| Alt 4.2 | The system prompts the user to confirm if they would like to delete the selected employee. | To confirm if the user wants to remove the employee details from the system | When an owner selects to remove a employees’ details |  | Notification message displayed requesting if the user wants to continue with removing the employee from the system. |
| Alt 4.2 | The updated employee details are invalid. | The inform the user that the entered employee maintenance details are invalid | When an owner submits the employee maintenance details. |  | Error message notifying owner that maintained employee details are invalid. |
| 4.3 Search employee | 4.3 | The system prompts the owner to search criteria for the employee | To make the owner aware of the details needed to be used to search for an employee | When owner requests to search for a specific employee on the system. | **Employee\_ID**  **Employee\_ID\_number**  Employee\_Email | Information displayed to the user on the screen. |
| 4.3 | The system displays the details of the selected employee | To show the owner the results of the searched details. | When an owner searches by criteria |  | Information displayed to the user on the screen. |
| Alt 4.3 | The system displays a message saying no results found matching the entered criteria | To inform the owner that no results were found on the system. | When an owner searches by criteria |  | Error message notifying owner no details matched searched criteria. |
| Alt 4.3 | The captured keyword or phrase is invalid | To inform the owner that the entered search criteria details are invalid. | When an owner searches by criteria |  | Error message notifying owner that entered search details are invalid. |
| 4.4 Add new employee type | 4.4 | System prompts the owner to enter the required details. | To ensure the owner knows to enter all details. | When an owner is using the system to add a new employee type | Employee\_Type | Information displayed to the user on the screen. |
| 4.4 | The system will display a success message. | To make the owner aware of the success of the add new employee type process. | When an owner submits required employee type details. |  | Notification message displayed to user about successful employee type addition. |
| Alt 4.4 | The entered employee type details are invalid. | To make the user aware that employee type details entered are not correct | When an owner enters invalid employee type details. |  | Error message displayed to user stating that employee details entered are invalid. |
| Alt 4.4 | Employee details already exist on the system. | To make user aware that the entered employee details already exist on the system | When an owner enters details that are already on the system. |  | Error message displayed to user stating that employee type details entered already exist on the system. |
| 4.5 Maintain employee type | 4.5 | The system will display the maintain employee type page and prompt the owner to search for an existing employee type on the system. | To take the user to the maintain employee type page. | When an owner is using the system to maintain an employee type |  | Information displayed to the user on the screen. |
| 4.5 | The system will invoke use case 4.6 Search employee type and display all results to the user. | To return a list of found employees types matching the search criteria and display it to user. | When an owner is using the system to maintain an employee type | **Employee\_Name**, **Employee\_Surname**, **Employee\_Email**, **Employee\_Address**, **Employee\_Contact, Employee\_Password, Access\_Level, Employee\_ID\_Number** | Information displayed to the user on the screen. |
| 4.5 | The system will display a success message. | To make the owner aware of the success of the maintain employee type process. | When an owner submits the maintained employee type details. |  | Notification message displayed to user about successful employee type maintenance. |
| Alt 4.5 | The system prompts the user to confirm if they would like to delete the selected client type. | To confirm if the user wants to remove the client details from the system | When an owner selects to remove a employee types’ details |  | Notification message displayed requesting if the user wants to continue with removing the employee type from the system. |
| Alt 4.5 | The updated employee type details are invalid. | To inform the user that the entered employee type maintenance details are invalid | When an owner submits the employee type maintenance details. |  | Error message notifying owner that maintained employee type details are invalid. |
| 4.6 Search employee type | 4.6 | The system prompts the owner to search criteria for the employee type | To make the owner aware of the details needed to be used to search for an employee type | When owner requests to search for a specific employee type on the system. |  | Information displayed to the user on the screen. |
| 4.6 | The system displays the details of the selected employee type | To show the owner the results of the searched details. | When an owner searches by criteria |  | Information displayed to the user on the screen. |
| Alt 4.6 | The system displays a message saying no results found matching the entered criteria | To inform the owner that no results were found on the system. | When an owner searches by criteria |  | Error message notifying owner no details matched searched criteria. |
| Alt 4.6 | The captured keyword or phrase is invalid | To inform the owner that the entered search criteria details are invalid. | When an owner searches by criteria |  | Error message notifying owner that entered search details are invalid. |
| 1. Client | | | | | | |
| 5.1 Add new client | 5.1 | System prompts the employee to enter the required details. | To ensure the employee knows to enter all details. | When an employee is using the system to add a new client | **Client\_Email, Client\_Contact\_Number, Client\_Address, Client\_Gender, Client\_Password, Client\_Instagram** | Information displayed to the user on the screen. |
| 5.1 | The system will display a success message. | To make the employee aware of the success of the add new client process. | When an employee submits required client details. |  | Notification message displayed to user about successful client addition. |
| Alt 5.1 | The entered client details are invalid. | To make the user aware that client details entered are not correct | When an employee enters invalid client details. |  | Error message displayed to user stating that client details entered are invalid. |
| Alt 5.1 | Client details already exist on the system. | To make user aware that the entered client details already exist on the system. | When an employee enters details that are already on the system. |  | Error message displayed to user stating that client details entered already exist on the system. |
| 5.2 Maintain client | 5.2 | The system will display the maintain client page and prompt the employee to search for an existing client on the system. | To take the user to the maintain client page. | When an employee is using the system to maintain client |  | Information displayed to the user on the screen. |
| 5.2 | The system will invoke use case 5.3 Search client and display all results to the user. | To return a list of found client matching the search criteria and display it to user. | When an employee is using the system to maintain client | **Client\_Email, Client\_Contact\_Number, Client\_Address, Client\_Gender, Client\_Password, Client\_Instagram** | Information displayed to the user on the screen. |
| 5.2 | The system will display a success message. | To make the employee aware of the success of the maintain client process. | When an employee submits the maintained client details. |  | Notification message displayed to user about successful employee type maintenance. |
| Alt 5.2 | The system prompts the user to confirm if they would like to delete the selected client type. | To confirm if the user wants to remove the client details from the system | When an owner selects to remove a employee types’ details |  | Notification message displayed requesting if the user wants to continue with removing the employee type from the system. |
| Alt 5.2 | The updated employee type details are invalid. | To inform the user that the entered employee type maintenance details are invalid | When an owner submits the employee type maintenance details. |  | Error message notifying owner that maintained employee type details are invalid. |
| 5.3 Search client | 5.3 | The system prompts the employee to search criteria for the client. | To make the employee aware of the details needed to be used to search for a client. | When employee requests to search for a specific client type on the system. |  | Information displayed to the user on the screen. |
| 5.3 | The system displays the details of the selected client | To show the employee the results of the searched details. | When an employee searches by criteria |  | Information displayed to the user on the screen. |
| Alt 5.3 | The system displays a message saying no results found matching the entered criteria | To inform the employee that no results were found on the system. | When an employee searches by criteria |  | Error message notifying employee no details matched searched criteria. |
| Alt 5.3 | The captured keyword or phrase is invalid | To inform the employee that the entered search criteria details are invalid. | When an employee searches by criteria |  | Error message notifying employee that entered search details are invalid. |
| 5.4 Signup for subscription | 5.4 | The system will display the subscription options with details regarding the options. | To make client aware of the options. | When an employee requests to sign up for a subscription. |  | Information displayed to the user on the screen. |
| 5.4 | The system will display a success message to the client and return the client to the home page. | To make the client aware of the success of the sign up for subscription process | When a client submits required signup option details |  | Notification message displayed to user about successful sign up for subscription. |
| 5.4 | The client is already signed up for a subscription and is invalid. | Inform client that they are already signed up for the subscription option | When a client tries to subscribe for a subscription option. |  | Error message notifying client that subscribing to subscription you already subscribing to is invalid. |
| 5.5 Cancel Subscription | 5.5 | The system will display the subscription page with available options**.** | Make user aware of the subscription options | When a client wants to cancel a subscription |  | Information displayed to the user on the screen. |
| 5.5 | The system will display a success message to the client and return the employee to the home page. | To inform the client of successful cancelation of subscription | When a client selects the cancel subscription option |  | Success message notifying client of successful cancelation of subscription. |
| 5.5 | The client is not signed up for any subscription thus is invalid. | To inform client that they cannot cancel a subscription they are not subscribed too. | When a client selects the cancel subscription option |  | Error message notifying client that they cannot cancel a subscription they are no subscribed too. |
| 5.6 Register for a challenge | 5.6 | The system will display the challenge details and options. | Inform client of the challenge details and options | When a client wants to register for a challenge on the system. |  | Information displayed to the user on the screen |
| 5.6 | Invoke **Use Case 5.17 Make Payment** | Display make payment screen to complete registering for challenge | When a client wants to pay for registering for a challenge. |  | Information displayed to the user on the screen |
| 5.6 | The system will display a success message to the client and return the client to the home page. | To make the client aware of the success of the sign up for subscription process | When a client submits payment details |  | Notification message displayed to user about successful registration for a challenge. |
| 5.6 | The client is already registered for the challenge and is invalid. | Inform client that they are already registered for the challenge option | When a client tries to register for a challenge option. |  | Error message notifying client that registering to challenge you already subscribing to is invalid. |
| 5.7 Unregister for challenge | 5.7 | The system will display the challenge page with available options**.** | Make client aware of challenge details they are registered too | When a client wishes to deregister from a challenge. |  | Information displayed to the user on the screen |
| 5.7 | The system will display a success to the client and return the client to the home page. | To inform client of successful deregistration from the challenge | When a client selects to deregister from challenge |  | Success message informing client of successful deregistration. |
| 5.7 | The client is not registered for any challenge thus is invalid. The system will display an error message | To inform client they are not registered for any challenges. | When a client selects to deregister from a challenge |  | Error message informing client they are not registered to any challenge. |
| 5.8 Purchase exercise plan | 5.8 | The system displays the exercise plan information on the screen along with an option to purchase the plan. | To make client aware of the purchase exercise plan details. | When a client wants to purchase an exercise plan. |  | Information displayed to the user on the screen |
| 5.8 | The system displays the business’ banking details  on-screen along with instructions which the client needs to follow to successfully purchase the exercise plan | To inform client on how to purchase an exercise plan | When a client wants to purchase an exercise plan. | **Exercise\_Plan\_ID, Sale\_ID**  **Client\_ID**  **Workout\_ID**  **Exercise\_Plan\_ID** | Information displayed to the user on the screen |
| 5.8 | Invoke **Use Case 1.9 Confirm payment** | Display confirm payment screen to complete paying for an exercise plan | When a client wants to pay for an exercise plan. |  | Information displayed to the user on the screen |
| 5.8 | The system will display a confirmation message and notify the client that we are awaiting payment confirmation. | To inform client about payment waiting confirmation. | When a client wants to pay for an exercise plan. |  | Confirmation message notify client that we are awaiting payment confirmation |
| 5.9 Purchase product | 5.9 | The system displays the product information on the screen along with an option to purchase the product. | To make client aware of the purchase product details. | When a client wants to purchase a product |  | Information displayed to the user on the screen |
| 5.9 | The system displays the business’ banking details  on-screen along with instructions which the client needs to follow to successfully purchase the product. | To inform client on how to purchase a product | When a client wants to purchase a product | **Exercise\_Plan\_ID, Sale\_ID**  **Client\_ID**  **Workout\_ID**  **Exercise\_Plan\_ID** | Information displayed to the user on the screen |
| 5.9 | Invoke **Use Case 1.9 Confirm payment** | Display confirm payment screen to complete paying for an exercise plan | When a client wants to pay for a product |  | Information displayed to the user on the screen |
| 5.9 | The system will display a confirmation message and notify the client that we are awaiting payment confirmation. | To inform client about payment waiting confirmation. | When a client wants to pay for a product. |  | Confirmation message notify client that we are awaiting payment confirmation. |
| Alt 5.9 | The client opts to not select a product to purchase | Allow the client to cancel a product purchase. | When a client wants to op out of a product purchase. |  | Use case is terminated and client is navigated to the product page and display message |
| 5.10 Make new personal training booking | 5.10 | The system will display the add new personal training booking screen and inform the client to enter a new personal training booking detail | Make sure client knows all details needed to make a new personal training booking. | When a client requests to make a new personal training booking. |  | Information displayed to the user on the screen |
| 5.10 | The system will display a success message to the client and return the client to the home page. | Inform client of successful personal training booking made. | When a client requests to make a new personal training booking. |  | Success message notifying client of successful personal training booking made. |
| Alt 5.10 | The entered personal training booking details are invalid | Inform client of invalid details | When a client submits a new personal training, booking details |  | Error messages notifying the client of invalid details. |
| 5.11 Cancel personal training booking | 5.11 | The system will display the personal training booking slots which the client has selected to attend on the screen. | Inform client on personal training booking slots they have selected to attended | When a client requests to cancel a personal training booking slot. | Personal\_Training\_Booking\_Table | Information displayed to the user on the screen |
| 5.11 | The system will display a success message to the client, return the client to the home page and notify the employee associated with the relevant personal training booking slot. | Inform client and personal trainer of cancellation of personal training booking slots. | When a client requests to cancel a personal training booking slot. | **Training\_Booking\_ID** | Success message notifying client and personal trainer of successful cancellation of personal training booking. |
| Alt 5.11 | The selected personal training booking slot is invalid. The system will display an error message | Inform client that chosen personal training slot to cancel that personal training booking invalid. | When a client requests to cancel a personal training booking slot. |  | Error message notifying client of invalid |
| 5.12 Make new challenge booking consultation | 5.12 | The system will display the add new challenge consultation booking screen and inform the client to enter a new personal training booking detail | Make sure client knows all details needed to make a new challenge consultation booking. | When a client requests to make a new challenge consultation booking. |  | Information displayed to the user on the screen |
| 5.12 | The system will display a success message to the client and return the client to the home page. | Inform client of successful challenge consultation booking made. | When a client requests to make a new challenge consultation booking. |  | Success message notifying client of successful challenge consultation booking made. |
| Alt 5.12 | The entered challenge consultation booking details are invalid | Inform client of invalid details | When a client submits a new challenge consultatio, booking details |  | Error messages notifying the client of invalid details. |
| 5.13 Cancel challenge consultation booking | 5.13 | The system will display the challenge consultation booking slots which the client has selected to attend on the screen. | Inform client on challenge consultation booking slots they have selected to attended | When a client requests to cancel a challenge consultation booking slot. |  | Information displayed to the user on the screen |
| 5.13 | The system will display a success message to the client, return the client to the home page and notify the employee associated with the relevant challenge consultation booking slot. | Inform client and personal trainer of cancellation of challenge consultation booking slots. | When a client requests to cancel a challenge consultation booking slot. |  | Success message notifying client and personal trainer of successful cancellation of challenge consultation booking. |
| Alt 5.13 | The selected challenge consultation booking slot is invalid. The system will display an error message | Inform client that chosen challenge consultation booking to cancel that challenge consultation booking invalid. | When a client requests to cancel a challenge consultation booking slot. |  | Error message notifying client of invalid |
| 5.14 Make payment | 5.14 | The system notifies the client that it is time to pay their subscription. | Make client aware of subscription needing payment | When a client wants to make a payment |  | Message notifying client of subscription needing payment. |
| 5.14 | The system displays the payment information on the screen:  along with steps to complete the transaction. | Make user aware of payment information and how to complete the transaction | When a client wants to make a payment | **Business\_Banking\_Details, Subscription\_ID, Balance, Start\_Date, End\_Date** | Information displayed to the user on the screen |
| 5.14 | The system will display a confirmation message and notify the client that we are awaiting payment confirmation. | Make user aware that system is waiting for payment confirmation. | When a client wants to make a payment |  | Confirmation message to notify client payment is awaiting confirmation |
| 5.14 | The client opts to not make the payment | To inform client of cancellation of subscription | When a client chooses to opt out of making a payment for renewal of subscription |  | Message notifying client of the canceled subscription. |
| 1. Reporting Subsystem | | | | | | |
| 6.1 Generate sales report | 6.1 | System returns generate sales report screen and prompts owner to enter details for the period they would like to report on. | Make user aware of the details needed. | When an owner would lie to generate a sale report | **Start\_Date** and **End\_Date** | Information displayed to the user on the screen |
| 6.1 | System displays the report to the user. | Present report to the user. | When a user enters a valid start and end date |  | Generated sales report presented to user. |
| Alt 6.1 | Start and end date entered are not valid, | Inform user of invalid details. | When a user enters an invalid start and end date |  | display an error message notifying owner of invalid start and end date details. |
| Alt 6.1 | The system generates a report in pdf format | Present report to user | When a valid start and end date details are entered |  | Generate report in pdf format |
| Alt 6.1 | The system generates a report in word format | Present report to user | When a valid start and end date details are entered |  | Generate report in word format |
| 6.2 Generate subscription report | 6.2 | System will display the report generating screen and will prompt the user to enter details isolating number of subscriptions over a certain period. | Make user aware of the details needed. | When an owner would lie to generate a subscription report | **Start\_Date** and **End\_Date** | Information displayed to the user on the screen |
| 6.2 | System displays the report to the user. | Present report to the user. | When a user enters a valid start and end date |  | Generated subscription report presented to user. |
| Alt 6.2 | Start and end date entered are not valid, | Inform user of invalid details. | When a user enters an invalid start and end date |  | display an error message notifying owner of invalid start and end date details. |
| Alt 6.2 | The system generates a report in pdf format | Present report to user | When a valid start and end date details are entered |  | Generate report in pdf format |
| Alt 6.2 | The system generates a report in word format | Present report to user | When a valid start and end date details are entered |  | Generate report in word format |
| 6.3 Generate client progress report | 6.3 | The system will return the generate client progress report screen and prompt the owner to enter which month they want to generate the report for. | Make user aware of the details needed. | When an owner would like to generate a client progress report | **Start\_Date** and **End\_Date** | Information displayed to the user on the screen |
| 6.3 | Invoke use case **5.3 Search client.** | To continue with the generate client progress report process | When a user enters a valid start and end date |  | Display 5.3 Search client page to allow owner to search client. |
| 6.3 | System displays the report to the user. | Present report to the user. | When a user searches for a valid client |  | Generated subscription report presented to user. |
| Alt 6.3 | Start and end date entered are not valid, | Inform user of invalid details. | When a user enters an invalid start and end date |  | display an error message notifying owner of invalid start and end date details. |
| Alt 6.3 | Client details entered are not valid | Inform user of invalid details. | When a user enters an invalid start and end date |  | display an error message notifying owner of invalid client details. |
| Alt 6.3 | The system generates a report in pdf format | Present report to user | When a user searches for a valid client |  | Generate report in pdf format |
| Alt 6.3 | The system generates a report in word format | Present report to user | When a user searches for a valid client |  | Generate report in word format |
| 6.4 Generate product levels report | 6.4 | System returns generate product levels report screen and prompts owner to enter details for the period they would like to report on. | Make user aware of the details needed. | When an owner would lie to generate a product levels report | **Start\_Date** and **End\_Date** | Information displayed to the user on the screen |
| 6.4 | System displays the report to the user. | Present report to the user. | When a user enters a valid start and end date |  | Generated product levels report presented to user. |
| Alt 6.4 | Start and end date entered are not valid, | Inform user of invalid details. | When a user enters an invalid start and end date |  | display an error message notifying owner of invalid start and end date details. |
| Alt 6.4 | The system generates a report in pdf format | Present report to user | When a valid start and end date details are entered |  | Generate report in pdf format |
| Alt 6.4 | The system generates a report in word format | Present report to user | When a valid start and end date details are entered |  | Generate report in word format |
| 6.5 Generate challenge booking report | 6.5 | System returns generate challenge booking report screen and prompts owner to enter details for the period they would like to report on. | Make user aware of the details  needed. | When an owner would lie to generate a sale report | **Start\_Date** and **End\_Date** | Information displayed to the user on the screen |
| 6.5 | System displays the report to the user. | Present report to the user. | When a user enters a valid start and end date |  | Generated challenge booking report presented to user. |
| Alt 6.5 | Start and end date entered are not valid, | Inform user of invalid details. | When a user enters an invalid start and end date |  | display an error message notifying owner of invalid start and end date details. |
| Alt 6.5 | The system generates a report in pdf format | Present report to user | When a valid start and end date details are entered |  | Generate report in pdf format |
| Alt 6.5 | The system generates a report in word format | Present report to user | When a valid start and end date details are entered |  | Generate report in word format |
| 1. Sale Subsystem | | | | | | |
| 7.1 Search Sale | 7.1 | System will return the search sale screen and prompt the employee to enter details. | Make user aware of what details need to be entered. | When an employee requests to search a sale | **Sale\_ID, Client\_ID, Sale\_Date** | Information displayed to the user on the screen |
| 7.1 | System displayed results to the employee in the form of a list. | Inform employee of retrieved results. | When an employee selects to search entered criteria |  | System displays returned search results for employee to use |
| Alt 7.1 | The employee did not enter a valid | Inform employee of invalid results | When an employee submits invalid search criteria. |  | Error message notifying employee of invalid search criteria |
| Alt 7.1 | No results matched the entered details. | Inform employee that no results match the entered criteria. | When an employee submits search criteria that is not on the system. | **Sale\_ID, Client\_ID, Sale\_Date** | display an error message notifying employee that no results match the search criteria. |
| 7.2 Return Sales | 7.2 | The system will open the returns screen and prompt the client to enter details relating to the sale requested to be returned | Make employee aware of what details need to be entered | When a client requests to return a sale. | **Sale\_ID** | Information displayed to the user on the screen |
| 7.2 | The system will display a success message once the return is completed and return the client to the home page | Inform client that return sale process was completed successfully. | When a client submits a valid sale ID |  | Success message notifying client of the success of the return sale process. |
| Alt 7.2 | Entered search criteria is not valid | Inform client of invalid search criteria | When a client submits invalid search criteria |  | Error message notifying client of invalid details. |
| Alt 7.2 | Product returned has been tampered with. | Inform client that product has been tampered with. | When a client tries to return a tampered product |  | Inform client that the return has been denied. |
| 1. Exercise Subsystem | | | | | | |
| 8.1 Add new exercise plan type | 8.1 | The system will display the add new exercise plan type screen and inform the employee to enter all the required exercise plan type details such as. | Informs the employee what exercise plan type details are required. | When employee requests to add a new exercise plan type to the system. | **Exercise\_Plan\_Type\_Description.** | Information displayed to the user on the screen. |
| 8.1 | The system will display a success message. | To make the employee aware of the success of the add new exercise plan type process. | When an employee submits required exercise plan type details. | None | Notification message displayed to user about successful exercise plan type addition. |
| Alt 8.1 | The entered exercise plan type details are invalid. | To make the user aware that exercise plan type details entered are not correct | When an employee enters invalid exercise plan type details. | None | Error message displayed to user stating that exercise plan type details entered are invalid. |
| Alt 8.1 | exercise plan type details already exist on the system. | To make user aware that the entered exercise plan type details already exist on the system | When an employee enters details that are already on the system. | None | Error message displayed to user stating that exercise plan type details entered already exist on the system. |
| 8.2 Maintain exercise plan type | 8.2 | The system will display the maintain exercise plan type page and prompt the employee to search for an existing exercise plan type on the system. | To take the user to the maintain exercise plan type page. | When an employee is using the system to maintain an exercise plan type. | None | Information displayed to the user on the screen. |
| 8.2 | The system will invoke use case 8.3 Search exercise plan type and display all results to the user. | To return a list of found exercise plan type matching the search criteria and display it to user. | When an employee is using the system to maintain an exercise plan type’s detail. | **Exercise\_Plan\_Type\_Description.** | Information displayed to the user on the screen. |
| 8.2 | The system will display a success message. | To make the employee aware of the success of the maintain exercise plan type process. | When an employee submits the maintained exercise plan type details. | None | Notification message displayed to user about successful exercise plan type maintenance. |
| Alt 8.2 | The system prompts the user to confirm if they would like to delete the selected exercise plan type. | To confirm if the user wants to remove the exercise plan types’ details from the system | When an employee selects to remove an exercise plan types’ detail | None | Notification message displayed requesting if they want to continue with removing the exercise plan type from the system. |
| 8.3 Search exercise plan type | 8.3 | The system prompts the owner to search criteria for the exercise plan type. | To make the owner aware of the details needed to be used to search for an exercise plan type | When owner requests to search for a specific exercise plan type on the system. | **Exercise\_Plan\_Type\_Description.**  **Exercise\_Plan\_Type\_ID** | Information displayed to the user on the screen. |
| 8.3 | The system displays the details of the selected exercise plan type | To show the employee the results of the searched details. | When an employee searches by criteria |  | Information displayed to the user on the screen. |
| Alt 8.3 | The system displays a message saying no results found matching the entered criteria | To inform the employee that no results were found on the system. | When an employee searches by criteria |  | Error message notifying employee no details matched searched criteria. |
| Alt 8.3 | The captured keyword or phrase is invalid | To inform the employee that the entered search criteria details are invalid. | When an employee searches by criteria |  | Error message notifying employee that entered search details are invalid. |
| 8.4 Add new exercise type | 8.4 | The system will display the add new exercise type screen and inform the employee to enter all the required exercise plan type details such as. | Informs the employee what exercise type details are required. | When employee requests to add a new exercise type to the system. | **Exercise\_Type\_Description** | Information displayed to the user on the screen. |
| 8.4 | The system will display a success message. | To make the employee aware of the success of the add new exercise type process. | When an employee submits required exercise type details. | None | Notification message displayed to user about successful exercise type addition. |
| Alt 8.4 | The entered exercise type details are invalid. | To make the user aware that exercise type details entered are not correct | When an employee enters invalid exercise plan type details. | None | Error message displayed to user stating that exercise type details entered are invalid. |
| Alt 8.4 | exercise type details already exist on the system. | To make user aware that the entered exercise type details already exist on the system | When an employee enters details that are already on the system. | None | Error message displayed to user stating that exercise type details entered already exist on the system. |
| 8.5 Maintain exercise type | 8.5 | The system will display the maintain exercise type page and prompt the employee to search for an existing exercise type on the system. | To take the user to the maintain exercise type page. | When an employee is using the system to maintain an exercise type. | None | Information displayed to the user on the screen. |
| 8.5 | The system will invoke use case 8.6 Search exercise type and display all results to the user. | To return a list of found exercise type matching the search criteria and display it to user. | When an employee is using the system to maintain an exercise type’s detail. | **Exercise\_Type\_Description** | Information displayed to the user on the screen. |
| 8.5 | The system will display a success message. | To make the employee aware of the success of the maintain exercise type process. | When an employee submits the maintained exercise type details. | None | Notification message displayed to user about successful exercise type maintenance. |
| Alt 8.5 | The system prompts the user to confirm if they would like to delete the selected exercise type. | To confirm if the user wants to remove the exercise types’ details from the system | When an employee selects to remove an exercise types’ detail | None | Notification message displayed requesting if they want to continue with removing the exercise type from the system. |
| Alt 8.5 | The updated exercise type details are invalid | Inform employee that details are invalid | When an employee |  | Error message notify employee that maintained exercise type details entered are invalid. |
| 8.6 Search exercise type | 8.6 | The system prompts the employee to search criteria for the exercise type. | To make the owner aware of the details needed to be used to search for an exercise type | When owner requests to search for a specific exercise type on the system. | **Exercise\_Type\_Description.**  **Exercise\_Type\_ID** | Information displayed to the user on the screen. |
| 8.6 | The system displays the details of the selected exercise type | To show the employee the results of the searched details. | When an employee searches by criteria |  | Information displayed to the user on the screen. |
| Alt 8.6 | The system displays a message saying no results found matching the entered criteria | To inform the employee that no results were found on the system. | When an employee searches by criteria |  | Error message notifying employee no details matched searched criteria. |
| Alt 8.6 | The captured keyword or phrase is invalid | To inform the employee that the entered search criteria details are invalid. | When an employee searches by criteria |  | Error message notifying employee that entered search details are invalid. |
| 8.7 Add new sets | 8.7 | The system will display the add new sets type screen and inform the employee to enter all the required set details such as. | Informs the employee what exercise type details are required. | When employee requests to add a new exercise type to the system. | **Set\_Description** | Information displayed to the user on the screen. |
| 8.7 | The system will display a success message. | To make the employee aware of the success of the add new set process. | When an employee submits required set details. | None | Notification message displayed to user about successful set addition. |
| Alt 8.7 | The entered set details are invalid. | To make the user aware that set details entered are not correct. | When an employee enters invalid set details. | None | Error message displayed to user stating that set details entered are invalid. |
| Alt 8.7 | Set details already exist on the system. | To make user aware that the entered set details already exist on the system | When an employee enters details that are already on the system. | None | Error message displayed to user stating that set details entered already exist on the system. |
| 8.8 Maintain sets | 8.8 | The system will display the maintain set page and prompt the employee to select a set they want to maintain on the system and allow the user to change details. | Allow the user to select set needing maintenance. | When an employee is using the system to maintain a set | Set\_Description | Information displayed to the user on the screen. |
| 8.8 | The system will display a success message. | To make the employee aware of the success of the maintain exercise process. | When an employee submits the maintained exercise details. | None | Notification message displayed to user about successful exercise maintenance. |
| Alt 8.8 | The system prompts the user to confirm if they would like to delete the selected exercise | To confirm if the user wants to remove the exercise details from the system | When an employee selects to remove an exercise detail | None | Notification message displayed requesting if they want to continue with removing the exercise from the system. |
| Alt 8.8 | The updated exercise details are invalid | Inform employee that details are invalid | When an employee submits invalid updated exercise details |  | Error message notify employee that maintained exercise details entered are invalid. |
| 8.9 Add new exercise | 8.9 | The system will display the add new exercise screen and inform the employee to enter all the required exercise plan type details such as. | Informs the employee what exercise details are required. | When employee requests to add a new exercise to the system. | **Exercise\_Name,**  **Exercise\_Image, Exercise\_Description,**  **Exercise\_Type\_ID** | Information displayed to the user on the screen. |
| 8.9 | The system will display a success message. | To make the employee aware of the success of the add new exercise process. | When an employee submits required exercise details. | None | Notification message displayed to user about successful exercise addition. |
| Alt 8.9 | The entered exercise details are invalid. | To make the user aware that exercise details entered are not correct | When an employee enters invalid exercise details. | None | Error message displayed to user stating that exercise details entered are invalid. |
| Alt 8.9 | exercise details already exist on the system. | To make user aware that the entered exercise type details already exist on the system | When an employee enters details that are already on the system. | None | Error message displayed to user stating that exercise type details entered already exist on the system. |
| 8.10 Maintain exercise | 8.10 | The system will display the maintain exercise page and prompt the employee to search for an existing exercise on the system. | To take the user to the maintain exercise page. | When an employee is using the system to maintain an exercise | None | Information displayed to the user on the screen. |
| 8.10 | The system will invoke use case 8.11 Search exercise and display all results to the user. | To return a list of found exercise matching the search criteria and display it to user. | When an employee is using the system to maintain an exercise detail. | **Exercise\_Description** | Information displayed to the user on the screen. |
| 8.10 | The system will display a success message. | To make the employee aware of the success of the maintain exercise process. | When an employee submits the maintained exercise details. | None | Notification message displayed to user about successful exercise maintenance. |
| Alt 8.10 | The system prompts the user to confirm if they would like to delete the selected exercise | To confirm if the user wants to remove the exercise details from the system | When an employee selects to remove an exercise detail | None | Notification message displayed requesting if they want to continue with removing the exercise from the system. |
| Alt 8.10 | The updated exercise details are invalid | Inform employee that details are invalid | When an employee submits invalid updated exercise details |  | Error message notify employee that maintained exercise details entered are invalid. |
| 8.11 Search exercise | 8.11 | The system prompts the employee to search criteria for the exercise. | To make the owner aware of the details needed to be used to search for an exercise | When owner requests to search for a specific exercise on the system. | **Exercise\_Description.**  **Exercise\_ID** | Information displayed to the user on the screen. |
| 8.11 | The system displays the details of the selected exercise | To show the employee the results of the searched details. | When an employee searches by criteria |  | Information displayed to the user on the screen. |
| Alt 8.11 | The system displays a message saying no results found matching the entered criteria | To inform the employee that no results were found on the system. | When an employee searches by criteria |  | Error message notifying employee no details matched searched criteria. |
| Alt 8.11 | The captured keyword or phrase is invalid | To inform the employee that the entered search criteria details are invalid. | When an employee searches by criteria |  | Error message notifying employee that entered search details are invalid. |
| 8.12 Add new exercise plan | 8.12 | The system will display the add new exercise plan screen and inform the employee to enter all the required exercise plan details such as. | Informs the employee what exercise plan details are required. | When employee requests to add a new exercise plan to the system. | **Exercise\_Plan\_Type\_Description.** | Information displayed to the user on the screen. |
| 8.12 | The system will display a success message. | To make the employee aware of the success of the add new exercise plan process. | When an employee submits required exercise plan details. | None | Notification message displayed to user about successful exercise plan addition. |
| Alt 8.12 | The entered exercise plan details are invalid. | To make the user aware that exercise plan details entered are not correct | When an employee enters invalid exercise plan details. | None | Error message displayed to user stating that exercise plan details entered are invalid. |
| Alt 8.12 | exercise plan details already exist on the system. | To make user aware that the entered exercise plan details already exist on the system | When an employee enters details that are already on the system. | None | Error message displayed to user stating that exercise plan details entered already exist on the system. |
| 8.13 Maintain exercise plan | 8.13 | The system will display the maintain exercise plan page and prompt the employee to search for an existing exercise on the system. | To take the user to the maintain exercise plan page. | When an employee is using the system to maintain an exercise plan | None | Information displayed to the user on the screen. |
| 8.13 | The system will invoke use case 8.14 Search exercise plan and display all results to the user. | To return a list of found exercise plan matching the search criteria and display it to user. | When an employee is using the system to maintain an exercise plan detail. | **Exercise\_Plan\_Description**  **Exercise\_Plan\_ID** | Information displayed to the user on the screen. |
| 8.13 | The system will display a success message. | To make the employee aware of the success of the maintain exercise plan process. | When an employee submits the maintained exercise plan details. | None | Notification message displayed to user about successful exercise plan maintenance. |
| Alt 8.13 | The system prompts the user to confirm if they would like to delete the selected exercise plan | To confirm if the user wants to remove the exercise plan details from the system | When an employee selects to remove an exercise plan detail | None | Notification message displayed requesting if they want to continue with removing the exercise plan from the system. |
| Alt 8.13 | The updated exercise plan details are invalid | Inform employee that details are invalid | When an employee submits invalid updated exercise plan details |  | Error message notify employee that maintained exercise plan details entered are invalid. |
| 8.14 Search exercise plan | 8.14 | The system prompts the employee to search criteria for the exercise plan. | To make the owner aware of the details needed to be used to search for an exercise plan | When owner requests to search for a specific exercise plan on the system. | **Exercise\_Plan\_Description.**  **Exercise\_Plan\_ID** | Information displayed to the user on the screen. |
| 8.14 | The system displays the details of the selected exercise plan type | To show the employee the results of the searched details. | When an employee searches by criteria |  | Information displayed to the user on the screen. |
| Alt 8.14 | The system displays a message saying no results found matching the entered criteria | To inform the employee that no results were found on the system. | When an employee searches by criteria |  | Error message notifying employee no details matched searched criteria. |
| Alt 8.14 | The captured keyword or phrase is invalid | To inform the employee that the entered search criteria details are invalid. | When an employee searches by criteria |  | Error message notifying employee that entered search details are invalid. |
| 8.15 Add exercise body type | 8.15 | The system will display the add new exercise body type screen and inform the employee to enter all the required exercise body type details such as. | Informs the employee what exercise body type details are required. | When employee requests to add a new exercise body type to the system | **Exercise\_Body\_Type\_Description.** | Information displayed to the user on the screen. |
| 8.15 | The system will display a success message. | To make the employee aware of the success of the add new exercise body type process. | When an employee submits required exercise body type details. | None | Notification message displayed to user about successful exercise body type addition. |
| Alt 8.15 | The entered exercise body type details are invalid. | To make the user aware that exercise body type details entered are not correct | When an employee enters invalid exercise body type details. | None | Error message displayed to user stating that exercise body type details entered are invalid. |
| Alt 8.15 | Exercise body type details already exist on the system. | To make user aware that the entered exercise body type details already exist on the system | When an employee enters details that are already on the system. | None | Error message displayed to user stating that exercise body type details entered already exist on the system. |
| 8.16 Maintain exercise body type | 8.16 | The system will display the maintain exercise body type page and prompt the employee to search for an existing exercise body type on the system on the system. | To take the user to the maintain exercise body type page. | When an employee is using the system to maintain an exercise body type | None | Information displayed to the user on the screen. |
| 8.16 | The system will invoke use case 8.17 Search exercise body type and display all results to the user. | To return a list of found exercise body types matching the search criteria and display it to user. | When an employee is using the system to maintain an exercise body type detail. | **Exercise\_Body\_Type\_Description**  **Exercise\_Body\_Type\_ID** | Information displayed to the user on the screen. |
| 8.16 | The system will display a success message. | To make the employee aware of the success of the maintain exercise body type process. | When an employee submits the maintained exercise body type details. | None | Notification message displayed to user about successful exercise body type maintenance. |
| Alt 8.16 | The system prompts the user to confirm if they would like to delete the selected exercise body type | To confirm if the user wants to remove the exercise body type details from the system | When an employee selects to remove an exercise body type detail | None | Notification message displayed requesting if they want to continue with removing the exercise body type from the system. |
| Alt 8.16 | The updated exercise body type details are invalid | Inform employee that details are invalid | When an employee submits invalid updated exercise body type details |  | Error message notify employee that maintained exercise body type details entered are invalid. |
| 8.17 Search exercise body type | 8.17 | The system prompts the employee to search criteria for the exercise body type. | To make the employee aware of the details needed to be used to search for an exercise body type | When owner requests to search for a specific exercise body type on the system. | **Exercise\_Body\_Type\_Description.**  **Exercise\_Body\_Type\_ID** | Information displayed to the user on the screen. |
| 8.17 | The system displays the details of the selected exercise body type | To show the employee the results of the searched details. | When an employee searches by criteria |  | Information displayed to the user on the screen. |
| Alt 8.17 | The system displays a message saying no results found matching the entered criteria | To inform the employee that no results were found on the system. | When an employee searches by criteria |  | Error message notifying employee no details matched searched criteria. |
| Alt 8.17 | The captured keyword or phrase is invalid | To inform the employee that the entered search criteria details are invalid. | When an employee searches by criteria |  | Error message notifying employee that entered search details are invalid. |
| 8.18 Add workout | 8.18 | The system will display the add workout screen and inform the employee to enter all the required work out details such as. | Informs the employee what work out details are required. | When employee requests to add a new workout to the system | **Workout\_Description** | Information displayed to the user on the screen. |
| 8.18 | The system will display a success message. | To make the employee aware of the success of the add work out process. | When an employee submits required work out details. | None | Notification message displayed to user about successful work out addition. |
| Alt 8.18 | The entered work out details are invalid. | To make the user aware that work out entered are not correct | When an employee enters invalid work out details. | None | Error message displayed to user stating that work out details entered are invalid. |
| 8.19 Maintain workout | 8.19 | The system will display the maintain work out page and prompt the employee to search for an existing work out on the system. | To take the user to the maintain work out page. | When an employee is using the system to maintain a work out | None | Information displayed to the user on the screen. |
| 8.19 | The system will invoke use case 8.20 Search work out and display all results to the user. | To return a list of found work out matching the search criteria and display it to user. | When an employee is using the system to maintain a work out detail. | **Workout\_Desc**  **Workout\_ID** | Information displayed to the user on the screen. |
| 8.19 | The system will display a success message. | To make the employee aware of the success of the maintain work out process. | When an employee submits the maintained work out details. | None | Notification message displayed to user about successful work out maintenance. |
| Alt 8.19 | The system prompts the user to confirm if they would like to delete the selected work out | To confirm if the user wants to remove the work out details from the system | When an employee selects to remove a work out details | None | Notification message displayed requesting if they want to continue with removing the work out from the system. |
| Alt 8.19 | The updated work out details are invalid | Inform employee that details are invalid | When an employee submits invalid updated work out details |  | Error message notify employee that maintained work out details entered are invalid. |
| 8.20 Search workout | 8.20 | The system prompts the employee to search criteria for the workout. | To make the owner aware of the details needed to be used to search for a workout | When owner requests to search for a specific workout on the system. | **Workout\_Description.**  **Workout\_ID** | Information displayed to the user on the screen. |
| 8.20 | The system displays the details of the selected exercise plan type | To show the employee the results of the searched details. | When an employee searches by criteria |  | Information displayed to the user on the screen. |
| Alt 8.20 | The system displays a message saying no results found matching the entered criteria | To inform the employee that no results were found on the system. | When an employee searches by criteria |  | Error message notifying employee no details matched searched criteria. |
| Alt 8.20 | The captured keyword or phrase is invalid | To inform the employee that the entered search criteria details are invalid. | When an employee searches by criteria |  | Error message notifying employee that entered search details are invalid. |
| 8.21 Add exercise image | 8.21 | The system will display the add exercise image screen and inform the employee to enter all the required exercise image details such as. | Informs the employee what exercise image details are required. | When employee requests to add a new exercise image to the system | Exercise\_Image\_URL | Information displayed to the user on the screen. |
| 8.21 | The system will display a success message. | To make the employee aware of the success of the add exercise image process. | When an employee submits required exercise image details. | None | Notification message displayed to user about successful exercise image addition. |
| Alt 8.21 | The entered exercise image details are invalid. | To make the user aware that exercise image entered are not correct | When an employee enters invalid exercise image details. | None | Error message displayed to user stating that exercise image details entered are invalid. |
| 8.22 Remove exercise image |  |  |  |  |  |  |
| 8.23 Search exercise image |  |  |  |  |  |  |
| 8.24 Create exercise plan | 8.24 | The system will display the create exercise plan screen and inform the employee to create the exercise plan. | To make the employee aware of the what details need to be added to create exercise. | When an employee request to create a new exercise plan. |  | System will display the create new exercise plan screen. |
|  | 8.24 | The system will display a success message to the employee and return the employee to the home page | To inform the employee of the successful addition of create new exercise plan. | When an submits a create new exercise plan |  | Success message displayed to notify employee of successful addition of create new exercise plan. |
|  | ALT 8.24 | The entered exercise plan details are invalid. | Inform employee that details are invalid | When an employee submits invalid exercise plan details. |  | The system will display an error message notifying the employee of the invalid details |
| 8.25 Create workout | 8.25 | The system will display the create workout screen and inform the employee to create the workout. | Inform employee to create the workout | When an employee requests to create a workout on the system. | **Exercise\_ID**  **Set\_ID** | The system will display the create workout page |
|  | 8.25 | The system will display a success message to the employee and return the employee to the home page. | Inform employee of success | When an employee submits valid work out details. |  | Success message notifying the employee of successful addition of workout to the system. |
|  | ALT 8.25 | The entered work out details are invalid. The system will display an error message. | Inform employee details are invalid | When an employee submits |  | Error message notifying employee of invalid workout details. |

Conclusion

This section assisted the team in understanding all the system outputs in more detail. It indicates the output purposes, when it gets produced, entities and attributes involved and the logical layout; all contained in a tabular document. This table allows the reader to be able to visualize and easily interpret all the outputs from the system.

8. Validation

Introduction

The validation of each functional specification, process and entities is provided to check if they correspond to the business requirements. This section allows the reader to compare the system validation to the business validation rules.

Validation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Subsystem | Use case | Requirement | Processes | Entities |
| 1. Admin susbystem | 1.1 Add new subscription type | 1.1 Add new subscription type | 1.1.1 | None |
| 1.1.2 | None |
| 1.1.3 | None |
| 1.1.4 | None |
| 1.1.5 | Subscription Type |
| 1.1.6 | None |
| 1.1.7 | None |
| 1.1.8 | None |
| 1.1.9 | None |
| 1.2 Maintain subscription | 1.2 Maintain subscription | 1.2.1 | None |
| 1.2.2 | None |
| 1.2.3 | None |
| 1.2.4 | None |
| 1.2.5 | None |
| 1.2.6 | None |
| 1.2.7 | None |
| 1.2.8 | Subscription type |
| 1.2.9 | None |
| 1.2.10 | None |
| 1.3 Search subscription type | 1.3 Search subscription type | 1.3.1 | None |
| 1.3.2 | None |
| 1.3.3 | None |
| 1.3.4 | None |
| 1.3.5 | None |
| 1.3.6 | Subscription type |
| 1.3.7 | None |
| 1.3.8 | None |
| 1.4 Add new challenge | 1.4 Add new challenge | 1.4.1 | None |
| 1.4.2 | None |
| 1.4.3 | None |
| 1.4.4 | None |
| 1.4.5 | Challenge |
| 1.4.6 | None |
| 1.4.7 | None |
| 1.4.8 | None |
| 1.4.9 | None |
| 1.5 Maintain challenge | 1.5 Maintain challenge | 1.5.1 | None |
| 1.5.2 | None |
| 1.5.3 | None |
| 1.5.4 | None |
| 1.5.5 | None |
| 1.5.6 | None |
| 1.5.7 | None |
| 1.5.8 | Challenge |
| 1.5.9 | None |
| 1.5.10 | None |
| 1.6 Search challenge | 1.6 Search challenge | 1.6.1 | None |
| 1.6.2 | None |
| 1.6.3 | None |
| 1.6.4 | None |
| 1.6.5 | None |
| 1.6.6 | None |
| 1.6.7 | None |
| 1.6.8 | Challenge |
| 1.6.9 | None |
| 1.6.10 | None |
| 1.7 Add Personal training books slots | 1.7 Add Personal training books slots | 1.7.1 | None |
| 1.7.2 | None |
| 1.7.3 | None |
| 1.7.4 | None |
| 1.7.5 | None |
| 1.7.6 | None |
| 1.7.7 | Personal training booking |
| 1.7.8 | None |
| 1.7.9 | None |
| 1.7.10 | None |
| 1.7.11 | None |
| 1.8 Add challenge consolation booking | 1.8 Add challenge consolation booking | 1.8.1 | None |
| 1.8.2 | None |
| 1.8.3 | None |
| 1.8.4 | None |
| 1.8.5 | None |
| 1.8.6 | None |
| 1.8.7 | Challenge booking consultation |
| 1.8.8 | None |
| 1.8.9 | None |
| 1.8.10 | None |
| 1.8.11 | none |
| 1.9 confirm payment | 1.9 confirm payment | 1.9.1 | None |
| 1.9.2 | None |
| 1.9.3 | None |
| 1.9.4 | None |
| 1.9.5 | None |
| 1.10 Search subscription | 1.10 Search subscription | 1.10.1 | None |
| 1.10.2 | None |
| 1.10.3 | None |
| 1.10.4 | None |
| 1.10.5 | None |
| 1.10.6 | None |
| 1.10.7 | None |
| 1.10.8 | None |
| 1.11 Search personal training booking | 1.11 Search personal training booking | 1.11.1 | None |
| 1.11.2 | None |
| 1.11.3 | None |
| 1.11.4 | None |
| 1.11.5 | None |
| 1.11.6 | Personal training booking |
| 1.11.7 | None |
| 1.11.8 | none |
| 1.12 Search challenge booking consultation | 1.12 Search challenge booking consultation | 1.12.1 | None |
| 1.12.2 | None |
| 1.12.3 | None |
| 1.12.4 | None |
| 1.12.5 | None |
| 1.12.6 | Challenge consultation booking |
| 1.12.7 | None |
| 1.12.8 | none |
| 1.13 Maintain company | 1.13 Maintain company | 1.13.1 | None |
| 1.13.2 | None |
| 1.13.3 | None |
| 1.13.4 | None |
| 1.13.5 | None |
| 1.13.6 | None |
| 1.13.7 | None |
| 1.13.8 | Company |
| 1.13.9 | None |
| 1.13.10 | None |
| 2. Product Subsystem | 2.1 Add new product type | 2.1 Add new product type | 2.1.1 | None |
| 2.1.2 | None |
| 2.1.3 | None |
| 2.1.4 | None |
| 2.1.5 | none |
| 2.1.6 | Product type |
| 2.1.7 | None |
| 2.1.8 | None |
| 2.1.9 | None |
| 2.2 Maintain product type | 2.2 Maintain product type | 2.2.1 | None |
| 2.2.2 | None |
| 2.2.3 | None |
| 2.2.4 | None |
| 2.2.5 | None |
| 2.2.6 | Product type |
| 2.2.7 | None |
| 2.2.8 | none |
| 2.3 Search product description | 2.3 Search product description | 2.3.1 | None |
| 2.3.2 | None |
| 2.3.3 | None |
| 2.3.4 | None |
| 2.3.5 | None |
| 2.3.6 | Product type |
| 2.3.7 | None |
| 2.3.8 | none |
| 2.4 Add new product | 2.4 Add new product | 2.4.1 | None |
| 2.4.2 | None |
| 2.4.3 | None |
| 2.4.4 | None |
| 2.4.5 | Product |
| 2.4.6 | none |
| 2.4.7 | None |
| 2.4.8 | None |
| 2.4.9 | None |
| 2.5 Maintain product | 2.5 Maintain product | 2.5.1 | None |
| 2.5.2 | None |
| 2.5.3 | None |
| 2.5.4 | None |
| 2.5.5 | none |
| 2.5.6 | none |
| 2.5.7 | None |
| 2.5.8 | None |
| 2.5.9 | product |
| 2.5.10 | None |
| 2.5.11 | none |
| 2.6 Search product | 2.6 Search product | 2.6.1 | None |
| 2.6.2 | None |
| 2.6.3 | None |
| 2.6.4 | None |
| 2.6.5 | none |
| 2.6.6 | product |
| 2.6.7 | None |
| 2.6.8 | None |
| 2.7 Add new suppliers | 2.7 Add new suppliers | 2.7.1 | None |
| 2.7.2 | None |
| 2.7.3 | None |
| 2.7.4 | None |
| 2.7.5 | Supplier |
| 2.7.6 | none |
| 2.7.7 | None |
| 2.7.8 | None |
| 2.7.9 | None |
| 2.8 Maintain supplier | 2.8 Maintain supplier | 2.8.1 | None |
| 2.8.2 | None |
| 2.8.3 | None |
| 2.8.4 | None |
| 2.8.5 | none |
| 2.8.6 | none |
| 2.8.7 | None |
| 2.8.8 | Supplier |
| 2.8.9 | none |
| 2.8.10 | None |
| 2.9 Search supplier | 2.9 Search supplier | 2.9.1 | None |
| 2.9.2 | None |
| 2.9.3 | None |
| 2.9.4 | None |
| 2.9.5 | supplier |
| 2.9.6 | none |
| 2.9.7 | None |
| 2.10 Add new product order | 2.10 Add new product order | 2.10.1 | none |
| 2.10.2 | None |
| 2.10.3 | None |
| 2.10.4 | None |
| 2.10.5 | Supplier order |
| 2.10.6 | None |
| 2.10.7 | None |
| 2.10.8 | None |
| 2.10.9 | none |
| 2.10.10 | none |
| 2.11 Pay supplier | 2.11 Pay supplier | 2.11.1 | none |
| 2.11.2 | None |
| 2.11.3 | None |
| 2.11.4 | None |
| 2.11.5 | Order |
| 2.11.6 | None |
| 2.11.7 | None |
| 2.11.8 | None |
| 2.12 Confirm supplier order | 2.12 Confirm supplier order | 2.12.1 | none |
| 2.12.2 | None |
| 2.12.3 | None |
| 2.12.4 | None |
| 2.12.5 | Order |
| 2.12.6 | None |
| 2.12.7 | None |
| 2.12.8 | None |
| 3. User susbsystem | 3.1 Login | 3.1 Login | 3.1.1 | None |
| 3.1.2 | None |
| 3.1.3 | None |
| 3.1.4 | None |
| 3.1.5 | Employee or client |
| 3.1.6 | None |
| 3.1.7 | None |
| 3.2 Update password | 3.2 Update password | 3.2.1 | None |
| 3.2.2 | None |
| 3.2.3 | None |
| 3.2.4 | None |
| 3.2.5 | none |
| 3.2.6 | none |
| 3.2.7 | None |
| 3.2.8 | None |
| 3.2.9 | Employee or client |
| 3.2.10 | None |
| 3.2.11 | none |
| 3.3 Forgotten password | 3.3 Forgotten password | 3.3.1 | None |
| 3.3.2 | None |
| 3.3.3 | None |
| 3.3.4 | None |
| 3.3.5 | Employee or client |
| 3.3.6 | none |
| 3.3.7 | None |
| 3.3.8 | None |
| 3.4 Log out | 3.4 Log out | 3.4.1 | None |
| 3.4.2 | None |
| 3.4.3 | None |
| 3.4.4 | None |
| 3.4.5 | none |
| 3.5 Add access level | 3.5 Add access level | 3.5.1 | None |
| 3.5.2 | None |
| 3.5.3 | None |
| 3.5.4 | None |
| 3.5.5 | Access level |
| 3.5.6 | none |
| 3.5.7 | None |
| 3.5.8 | None |
| 3.5.9 | none |
| 3.6 Maintain access level | 3.6.1 | None |
| 3.6 Maintain access level | 3.6.2 | None |
| 3.6.3 | None |
| 3.6.4 | None |
| 3.6.5 | None |
| 3.6.6 | none |
| 3.6.7 | Access level |
| 3.6.8 | None |
| 3.6.9 | none |
| 3.7 Search access level | 3.7 Search access level | 3.7.1 | None |
| 3.7.2 | None |
| 3.7.3 | None |
| 3.7.4 | None |
| 3.7.5 | None |
| 3.7.6 | none |
| 3.7.7 | none |
| 3.8 Add functionality | 3.8 Add functionality | 3.8.1 | None |
| 3.8.2 | None |
| 3.8.3 | None |
| 3.8.4 | None |
| 3.8.5 | Functionality |
| 3.8.6 | none |
| 3.8.7 | none |
| 3.8.8 | None |
| 3.8.9 | none |
| 3.9 Maintain functionality | 3.9 Maintain functionality | 3.9.1 | None |
| 3.9.2 | None |
| 3.9.3 | None |
| 3.9.4 | None |
| 3.9.5 | none |
| 3.9.6 | none |
| 3.9.7 | Functionality |
| 3.9.8 | None |
| 3.9.9 | none |
| 3.10 Search functionality | 3.10 Search functionality | 3.10.1 | None |
| 3.10.2 | None |
| 3.10.3 | None |
| 3.10.4 | None |
| 3.10.5 | none |
| 3.10.6 | none |
| 3.10.7 | none |
| 4. Employee Subsystem | 4.1 Add new employee | 4.1 Add new employee | 4.1.1 | None |
| 4.1.2 | None |
| 4.1.3 | None |
| 4.1.4 | None |
| 4.1.5 | Employee |
| 4.1.6 | none |
| 4.1.7 | None |
| 4.1.8 | None |
| 4.1.9 | none |
| 4.2 Maintain employee | 4.2 Maintain employee | 4.2.1 | None |
| 4.2.2 | None |
| 4.2.3 | None |
| 4.2.4 | None |
| 4.2.5 | none |
| 4.2.6 | none |
| 4.2.7 | None |
| 4.2.8 | Employee |
| 4.2.9 | none |
| 4.2.10 | none |
| 4.3 Search employee | 4.3 Search employee | 4.3.1 | None |
| 4.3.2 | None |
| 4.3.3 | None |
| 4.3.4 | None |
| 4.3.5 | none |
| 4.3.6 | Employee |
| 4.3.7 | None |
| 4.3.8 | none |
| 4.4 Add new employee type | 4.4 Add new employee type | 4.2.1 | None |
| 4.2.2 | None |
| 4.2.3 | None |
| 4.2.4 | None |
| 4.2.5 | Employee type |
| 4.2.6 | none |
| 4.2.7 | None |
| 4.2.8 | none |
| 4.2.9 | none |
| 4.5 Maintain employee type | 4.5 Maintain employee type | 4.5.1 | None |
| 4.5.2 | None |
| 4.5.3 | None |
| 4.5.4 | None |
| 4.5.5 | none |
| 4.5.6 | none |
| 4.5.7 | None |
| 4.5.8 | Employee Type |
| 4.5.9 | none |
| 4.5.10 | none |
| 4.6 Search employee type | 4.6 Search employee type | 4.6.1 | None |
| 4.6.2 | None |
| 4.6.3 | None |
| 4.6.4 | None |
| 4.6.5 | none |
| 4.6.6 | none |
| 4.6.7 | Employee Type |
| 4.6.8 | none |
| 5. Client subsystem | 5.1 Add new client | 5.1 Add new client | 5.1.1 | None |
| 5.1.2 | None |
| 5.1.3 | None |
| 5.1.4 | None |
| 5.1.5 | Client |
| 5.1.6 | none |
| 5.1.7 | None |
| 5.1.8 | None |
| 5.2 Maintain client | 5.2 Maintain client | 5.1.1 | None |
| 5.2.2 | None |
| 5.2.3 | None |
| 5.2.4 | None |
| 5.2.5 | Client |
| 5.2.6 | none |
| 5.2.7 | None |
| 5.3 Search client | 5.3 Search client | 5.3.1 | None |
| 5.3.2 | None |
| 5.3.3 | None |
| 5.3.4 | None |
| 5.3.5 | none |
| 5.3.6 | Client |
| 5.3.7 | None |
| 5.3.8 | None |
| 5.4 Signup for subscription | 5.4 Signup for subscription | 5.4.1 | None |
| 5.4.2 | None |
| 5.4.3 | None |
| 5.4.4 | None |
| 5.4.5 | Subscription |
| 5.4.6 | none |
| 5.4.7 | None |
| 5.4.8 | None |
| 5.5 Cancel subscription | 5.5 Cancel subscription | 5.5.1 | None |
| 5.5.2 | None |
| 5.5.3 | None |
| 5.5.4 | None |
| 5.5.5 | none |
| 5.5.6 | none |
| 5.5.7 | None |
| 5.6 Register for a challenge | 5.6 Register for a challenge | 5.6.1 | None |
| 5.6.2 | None |
| 5.6.3 | None |
| 5.6.4 | None |
| 5.6.5 | Challenge |
| 5.6.6 | none |
| 5.6.7 | None |
| 5.6.8 | None |
| 5.7 Unregister for a challenge | 5.7 Unregister for a challenge | 5.7.1 | None |
| 5.7.2 | None |
| 5.7.3 | None |
| 5.7.4 | None |
| 5.7.5 | none |
| 5.7.6 | none |
| 5.7.7 | None |
| 5.8 Purchase exercise plan | 5.8 Purchase exercise plan | 5.8.1 | None |
| 5.8.2 | None |
| 5.8.3 | None |
| 5.8.4 | None |
| 5.8.5 | None |
| 5.8.6 | none |
| 5.8.7 | None |
| 5.8.8 | None |
| 5.8.9 | None |
| 5.9 Purchase product | 5.9 Purchase product | 5.9.1 | None |
| 5.9.2 | None |
| 5.9.3 | None |
| 5.9.4 | None |
| 5.9.5 | None |
| 5.9.6 | none |
| 5.9.7 | None |
| 5.9.8 | None |
| 5.9.9 | None |
| 5.10 Make new personal training booking | 5.10 Make new personal training booking | 5.10.1 | None |
| 5.10.2 | None |
| 5.10.3 | None |
| 5.10.4 | None |
| 5.10.5 | Personal training booking |
| 5.10.6 | none |
| 5.10.7 | None |
| 5.10.8 | None |
| 5.10.9 | None |
| 5.11 Cancel personal training booking | 5.11 Cancel personal training booking | 5.11.1 | None |
| 5.11.2 | None |
| 5.11.3 | None |
| 5.11.4 | None |
| 5.11.5 | Personal training booking |
| 5.11.6 | none |
| 5.11.7 | None |
| 5.12 Make new challenge consultation booking | 5.12 Make new challenge consultation booking | 5.12.1 | None |
| 5.12.2 | None |
| 5.12.3 | None |
| 5.12.4 | None |
| 5.12.5 | Challenge consultation booking |
| 5.12.6 | none |
| 5.12.7 | None |
| 5.12.8 | None |
| 5.12.9 | None |
| 5.13 Cancel challenge consultation booking | 5.13 Cancel challenge consultation booking | 5.13.1 | None |
| 5.13.2 | None |
| 5.13.3 | None |
| 5.13.4 | None |
| 5.13.5 | Challenge consultation booking |
| 5.13.6 | none |
| 5.13.7 | None |
| 5.14 Make payment | 5.14 Make payment | 5.14.1 | None |
| 5.14.2 | None |
| 5.14.3 | None |
| 5.14.4 | None |
| 5.14.5 | None |
| 5.14.6 | none |
| 5.14.7 | None |
| 5.14.8 | None |
| 6. Reporting Subsystem | 6.1 Generate sales report | 6.1 Generate sales report | 6.1.1 | None |
| 6.1.2 | None |
| 6.1.3 | None |
| 6.1.4 | None |
| 6.1.5 | Start date/ end date |
| 6.1.6 | none |
| 6.1.7 | None |
| 6.1.8 | None |
| 6.1.9 | None |
| 6.1.10 | None |
| 6.1.11 | none |
| 6.2 Generate subscription report | 6.2 Generate subscription report | 6.2.1 | None |
| 6.2.2 | None |
| 6.2.3 | None |
| 6.2.4 | None |
| 6.2.5 | Start date/ end date |
| 6.2.6 | none |
| 6.2.7 | None |
| 6.2.8 | None |
| 6.2.9 | None |
| 6.2.10 | None |
| 6.2.11 | none |
| 6.3 Generate client progress report | 6.3 Generate client progress report | 6.3.1 | None |
| 6.3.2 | None |
| 6.3.3 | None |
| 6.3.4 | None |
| 6.3.5 | Start date/ end date |
| 6.3.6 | none |
| 6.3.7 | None |
| 6.3.8 | None |
| 6.3.9 | Client |
| 6.3.10 | None |
| 6.3.11 | none |
| 6.3.12 | None |
| 6.3.13 | None |
| 6.3.14 | None |
| 6.3.15 | none |
| 6.4 Generate product levels report | 6.4 Generate product levels report | 6.4.1 | None |
| 6.4.2 | None |
| 6.4.3 | None |
| 6.4.4 | None |
| 6.4.5 | Start date/ end date |
| 6.4.6 | none |
| 6.4.7 | None |
| 6.4.8 | None |
| 6.4.9 | None |
| 6.4.10 | None |
| 6.5 Generate challenge booking report | 6.5 Generate challenge booking report | 6.5.1 | None |
| 6.5.2 | None |
| 6.5.3 | None |
| 6.5.4 | None |
| 6.5.5 | Start date/ end date |
| 6.5.6 | none |
| 6.5.7 | None |
| 6.5.8 | None |
| 6.5.9 | None |
| 6.5.10 | None |
| 6.5.11 | none |
| 7. Sale susbsystem | 7.1 Search sale |  | 7.1.1 | None |
| 7.1.2 | None |
| 7.1.3 | None |
| 7.1.4 | None |
| 7.1.5 | Start date/ end date/ client and sale |
| 7.1.6 | none |
| 7.1.7 | None |
| 7.2 Return sale |  | 7.2.1 | None |
| 7.2.2 | None |
| 7.2.3 | None |
| 7.2.4 | None |
| 7.2.5 | Sale date & client |
| 7.2.6 | none |
| 7.2.7 | None |
| 7.2.8 | none |
| 8. Exercise subsystem | 8.1 Add new exercise plan type |  | 8.1.1 | None |
| 8.1.2 | None |
| 8.1.3 | None |
| 8.1.4 | None |
| 8.1.5 | Exercise plan type |
| 8.1.6 | none |
| 8.1.7 | None |
| 8.1.9 | none |
| 8.2 Maintain exercise plan type |  | 8.2.1 | None |
| 8.2.2 | None |
| 8.2.3 | None |
| 8.2.4 | None |
| 8.2.5 | none |
| 8.2.6 | none |
| 8.2.7 | None |
| 8.2.8 | Exercise plan type |
| 8.2.9 | none |
| 8.2.10 | none |
| 8.3 Search exercise plan type |  | 8.3.1 | None |
| 8.3.2 | None |
| 8.3.3 | None |
| 8.3.4 | None |
| 8.3.5 | none |
| 8.3.6 | Exercise plan type |
| 8.3.7 | None |
| 8.3.8 | none |
| 8.4 Add new exercise type |  | 8.4.1 | None |
| 8.4.2 | None |
| 8.4.3 | None |
| 8.4.4 | None |
| 8.4.5 | Exercise type |
| 8.4.6 | none |
| 8.4.7 | None |
| 8.4.8 | none |
| 8.4.9 | none |
| 8.4.10 | none |
| 8.4.11 | none |
| 8.5 Maintain exercise type |  | 8.5.1 | None |
| 8.5.2 | None |
| 8.5.3 | None |
| 8.5.4 | None |
| 8.5.5 | none |
| 8.5.6 | none |
| 8.5.7 | None |
| 8.5.8 | Exercise type |
| 8.5.9 | none |
| 8.5.10 | none |
| 8.6 Search exercise type |  | 8.6.1 | None |
| 8.6.2 | None |
| 8.6.3 | None |
| 8.6.4 | None |
| 8.6.5 | none |
| 8.6.6 | Exercise type |
| 8.6.7 | None |
| 8.6.8 | none |
| 8.7 Add new sets |  | 8.7.1 | None |
| 8.7.2 | None |
| 8.7.3 | None |
| 8.7.4 | None |
| 8.7.5 | Set |
| 8.7.6 | none |
| 8.7.7 | None |
| 8.7.8 | none |
| 8.8 Maintain sets |  | 8.8.1 | None |
| 8.8.2 | None |
| 8.8.3 | None |
| 8.8.4 | None |
| 8.8.5 | none |
| 8.8.6 | none |
| 8.8.7 | None |
| 8.8.8 | Set |
| 8.8.9 | none |
| 8.8.10 | none |
| 8.9 Add new exercise |  | 8.9.1 | None |
| 8.9.2 | None |
| 8.9.3 | None |
| 8.9.4 | None |
| 8.9.5 | Exercise |
| 8.9.6 | Exercise type |
| 8.9.7 | None |
| 8.9.8 | none |
| 8.9.9 | none |
| 8.10 Maintain exercise |  | 8.10.1 | None |
| 8.10.2 | None |
| 8.10.3 | None |
| 8.10.4 | None |
| 8.10.5 | Exercise |
| 8.10.6 | none |
| 8.10.7 | None |
| 8.10.8 | none |
| 8.11 Search exercise |  | 8.11.1 | None |
| 8.11.2 | None |
| 8.11.3 | None |
| 8.11.4 | None |
| 8.11.5 | none |
| 8.11.6 | Exercise |
| 8.11.7 | None |
| 8.11.8 | none |
| 8.12 Add new exercise plan |  | 8.12.1 | None |
| 8.12.2 | None |
| 8.12.3 | None |
| 8.12.4 | None |
| 8.12.5 | Exercise Plan |
| 8.12.6 | none |
| 8.12.7 | None |
| 8.12.8 | none |
| 8.12.9 | none |
| 8.13 Maintain exercise plan |  | 8.13.1 | None |
| 8.13.2 | None |
| 8.13.3 | None |
| 8.13.4 | None |
| 8.13.5 | none |
| 8.13.6 | none |
| 8.13.7 | None |
| 8.13.8 | Exercise Plan |
| 8.13.9 | none |
| 8.13.10 | None |
| 8.14 Search exercise plan |  | 8.14.1 | None |
| 8.14.2 | None |
| 8.14.3 | None |
| 8.14.4 | None |
| 8.14.5 | none |
| 8.14.6 | Exercise Plan |
| 8.14.7 | None |
| 8.14.8 | none |
| 8.15 Add new exercise body type |  | 8.15.1 | None |
| 8.15.2 | None |
| 8.15.3 | None |
| 8.15.4 | None |
| 8.15.5 | Exercise body type |
| 8.15.6 | none |
| 8.15.7 | None |
| 8.15.8 | none |
| 8.15.9 | none |
| 8.16 Maintain exercise body type |  | 8.16.1 | None |
| 8.16.2 | None |
| 8.16.3 | None |
| 8.16.4 | None |
| 8.16.5 | none |
| 8.16.6 | none |
| 8.16.7 | None |
| 8.16.8 | Exercise Body Type |
| 8.16.9 | none |
| 8.16.10 | None |
| 8.17 Search exercise body type |  | 8.17.1 | None |
| 8.17.2 | None |
| 8.17.3 | None |
| 8.17.4 | None |
| 8.17.5 | none |
| 8.17.6 | Exercise body type |
| 8.17.7 | None |
| 8.17.8 | none |
| 8.18 Add workout |  | 8.18.1 | None |
| 8.18.2 | None |
| 8.18.3 | None |
| 8.18.4 | None |
| 8.18.5 | Workout |
| 8.18.6 | none |
| 8.18.7 | None |
| 8.18.8 | none |
| 8.19 Maintain workout |  | 8.19.1 | None |
| 8.19.2 | None |
| 8.19.3 | None |
| 8.19.4 | None |
| 8.19.5 | none |
| 8.19.6 | none |
| 8.19.7 | None |
| 8.19.8 | Workout |
| 8.19.9 | none |
| 8.19.10 | None |
| 8.20 Search workout |  | 8.20.1 | None |
| 8.20.2 | None |
| 8.20.3 | None |
| 8.20.4 | None |
| 8.20.5 | none |
| 8.20.6 | Workout |
| 8.20.7 | None |
| 8.20.8 | none |
| 8.21 Add exercise image |  | 8.21.1 | None |
| 8.21.2 | None |
| 8.21.3 | None |
| 8.21.4 | None |
| 8.21.5 | Exercise Image |
| 8.21.6 | none |
| 8.21.7 | None |
| 8.21.8 | none |
| 8.22 Remove exercise image |  | 8.22.1 | None |
| 8.22.2 | None |
| 8.22.3 | None |
| 8.22.4 | None |
| 8.22.5 | None |
| 8.22.6 | none |
| 8.22.7 | None |
| 8.22.8 | none |
| 8.23 Search exercise image |  | 8.23.1 | None |
| 8.23.2 | None |
| 8.23.3 | None |
| 8.23.4 | None |
| 8.23.5 | None |
| 8.23.6 | Exercise image |
| 8.23.7 | None |
| 8.23.8 | none |
| 8.24 Create exercise plan |  | 8.24.1 | None |
| 8.24.2 | None |
| 8.24.3 | None |
| 8.24.4 | None |
| 8.24.5 | Exercise plan |
| 8.24.6 | none |
| 8.24.7 | None |
| 8.24.8 | none |
| 8.25 Create workout |  | 8.25.1 | None |
| 8.25.2 | None |
| 8.25.3 | None |
| 8.25.4 | None |
| 8.25.5 | Workout |
| 8.25.6 | none |
| 8.25.7 | None |
| 8.25.8 | none |

Conclusion

The section contained the validation of each functional specifications and provided an outline, which showed the team which requirements have been dealt with; and which one still needs to be addressed.

9. Sign Off by the Client

Introduction

The development team has presented the final functional specification document to the client for purposes of reviewing the contents therein. It is ensured that all specifications are understood and agreed to, and in this section the client provides the sign off to conclude the document.

|  |
| --- |
| Client Sign-off |

Jesse Pretorius

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Conclusion

By signing this document, the client declares that he/she has read and understood the contents of the document therein and gives approval of the functional specification.

10. Complexity

Introduction

In this section, we will analyze the complexity requirements and indicate the requirements we will meet and the total marks we will achieve.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| **Topic** | **Level** |  | **Marks** | **MAX** |
| **1. Special GUI** | For online applications: Responsive web design | 3 | 3 | **42** |
| For desktop applications: Form design according to design principles (Schneiderman’s golden rule on navigation applies here) |
| Appropriate use of grids/tables |  | 3 |
| Appropriate use of tabs/links |  | 3 |
| Use of graphs in an appropriate business context |  | 4 |
| The storage and display of graphical information, like photos with a good business reason |  | 3 |
| Working e-mail automatically generated from the database in an appropriate business context |  | 2 |
| SMS messages automatically generated from the system in an appropriate business context |  | 2 |
| Extensive user-friendly search facility |  | 3 |
| At least one use of a tree to display data from the database |  | 3 |
| Able to dynamically modify a data tree structure and in doing so adjusting the data in the database |  | 4 |
| At least one use of a calendar view of data (not a date/time picker; not a plug-in such as Google calendar) |  | 3 |
| Uploading a file into the system with appropriate business reason |  | 3 |
| The use of audio/video in an appropriate business context |  | 3 |
| At least one use of an administrator configurable timer in an appropriate business context |  | 3 |
| **2. Database access** | At least 30 tables used (4 member groups) or 40 tables used (5 member groups) | 6 | 6 | **15** |
| Full referential integrity on all tables | 6 | 6 |
| At least one use of master-detail table relationships (Schneiderman’s golden rule on system status applies here) | \* | 3 |
| **3. Reports** | At least 3 simple list reports in a reporting tool (no control breaks, no graphs, single table) | 3 | 3 | **15** |
| At least 2 transactional report with 2 or more control breaks (with heading and calculated values/totals, multiple tables) | \* | 6 |
| At least 1 report with adjustable criteria | 3 | 3 |
| At least 1 management report using a graph |  | 3 |
| **4. Flexibility** | All data that can change in future should not be hard coded but maintained in a sub-module of the system (e.g. Lookup tables) | 6 | 6 | **12** |
| Some business rules are not hard coded, but maintained in a sub-module of the system. | 6 | 6 |
| **5. Error handling** | **All** system-generated errors are trapped and consistent, user-friendly error messages are displayed | 6 | 6 | **12** |
| Appropriate data validation on **all** input fields | 6 | 6 |
| **6. Help** | At least one menu item or other control that opens up a complete help document (HTML, PDF, Help-file) |  | 3 | **15** |
| Extensive context-sensitive help. E.g. calling Help on a specific screen/function will automatically open the specific help for that screen/function. |  | 6 |
| Search Facility on Help |  | 3 |
| Extensive use of hints |  | 3 |
| **7. Security** | Logon screen with user ID and password and fixed user profiles | 3 | 3 | **13** |
| Applying two factor authentications with applicable business reason. |  | 3 |
| Encrypted passwords in database |  | 1 |
| Flexible user profiles (i.e. you can dynamically add user profiles that will enable/disable access to certain parts of the system) | 6 | 6 |
| **8. Audit Trail** | An audit trail of all transactions in the system showing at least date, time, user, transaction type, critical data (such as amount and quantity of transaction) |  | 6 | **9** |
| Able to search the audit trail on any of the following: date, user, transaction type |  | 3 |
| **9. Deployment** | For a desktop application: Fully functional installation disks that take care of application installation requirements (install and uninstall) |  | 3 | **15** |
| For an online application: Deployment of application to a publicly accessible web server |  | 3 |
| For a mobile application: Deployment to an App Market place (such as the Play Store or the AppStore) |  | 6 |
| Deployment of the database to a remote database server |  | 3 |
| **10. Backup and Restore** | A backup and restore subsystem exist that backup/restore all data (system may exit during restore) |  | 3 | **3** |
| **11. Import/Export Data** | Able to open Word or Excel and automatically place data in it based on the parameters provided (with a good business reason) |  | 6 | **9** |
| XML or JSON: At least 1 XMLor JSON file for Importing or Exporting of data (with good business reason) |  | 3 |
| **12. External INPUT device** | Simple Link to an external INPUT device using plug-and-play technology, such as a swipe card reader, bar code reader, etc. or a native component such as a QR reader, a GPS component, etc |  | 3 | **18** |
| Loose Link to an external INPUT device using device specific software. Data or images must seamlessly be stored in the database but device specific software is visible to the user. (This could include a digital camera, scanner, voice recording device, thump print reader, etc.) |  | 6 |
| Tight Link to an external INPUT device using device specific software. Data or images must seamlessly be stored in the database but device specific software is **not** visible to the user. (This could include a digital camera, scanner, voice recording device, thump print reader, etc.) |  | 9 |
| **13. External APPLICATION / Services** | Integrate an existing web service into your application (with good business reason) |  | 3 | **9** |
| A fully functional link to an installed external application system exists and the interface must be shown to work on the external system. Note that this excludes Microsoft Office Applications |  | 6 |
| **14. Multiplatform processing for an appropriate business reason** | Appropriate business use of static views on an alternative platform. |  | 3 | **27** |
| Appropriate use of dynamic views on an alternative platform (i.e. data is displayed from the system’s database) |  | 3 |
| Appropriate use of substantial dynamic views on an alternative platform (i.e. both reading and writing data from the system’s database) |  | 9 |
| Uploading a file through an alternative platform onto the system’s database. |  | 3 |
| Substantial processing on a third platform (i.e. both reading and writing data from the system’s database) |  | 9 |
| **15. Programming Principles** | The use of a data layer to facilitate interaction between your database and your business layer |  | 3 | **12** |
| The use of an API to facilitate interaction between your business layer and your presentation layer |  | 6 |
| Comprehensive use of stored procedures and/or triggers and/or jobs. |  | 3 |
| **16. Innovative addition to the system** | Any very advanced innovative addition to the system (e.g. machine learning, AI, block chain, text mining, IOT, etc.) | # | 9 | **9** |
|  |  |  |  |  |
|  | **Maximum Complexity Marks** |  |  | **235** |
|  | Group 15 Complexity Marks so far: |  |  | 44/150 |

Conclusion

In this section, we analyzed the complexity requirements and indicated the marks we are currently meeting based on content of the functional specification document. Also indicated is the total marks we will achieve.

11. General (Team Sign-Off)

Introduction

This section is composed of the sign-off required by all team members.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Team Sign off |  |  |  |  |
| Janco Steenkamp \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Joshua Eales \_\_\_\_\_\_\_\_\_\_\_\_\_ | Doreen Maloka \_\_\_\_\_\_\_\_\_\_\_\_ | Iain Steyn \_\_\_\_\_\_\_\_\_\_\_\_ | Fhatuwani Dalana \_\_\_\_\_\_\_\_\_\_\_\_\_ |

Conclusion

By signing this document, all team members declare that they have read and agreed to the contents of the document, as well as the accuracy and completeness of the document.

12. Conclusion: Deliverable 02 (Functional Specification)

The Aegis development team provides a detailed functional specification of **each requirement identified** to provide a desired and efficient solution to the client’s current problems regarding their system.

The **functional specification** focusses on **processes, activities, inputs and outputs** of the proposed system. This document divides the system into various related components called subsystems. Each component is **modelled** in various ways by use of either **UML** or **Data Flow** diagrams to clearly display the actor and system interactions as well as the inputs added to the system and outputs received from the system.

The **hierarchy** of the system is displayed using the **decomposition** diagram. **Use Case** diagrams and **Context** diagrams are used to display an **overview** of the entire system to interested stakeholders

The system dives further into the system for a deeper level of understanding of the data flows and processes that occur by use of **High-, Mid-,** and **Primitive Level** diagrams through **process modelling.**

An individual **Primitive Level** diagram and **Activity** diagram is created for every **individual Use Case** narrative which clearly describes to relevant processes and interactions.

This document helps stakeholders understand the flow of processes within the use cases as well as **what** the system will achieve after completion of a use case; without any unnecessary technical specification about **how** the system will achieve its result.

After compiling the functional specification document, the development team has laid out a clear logical map of every aspect regarding the system and is communicated within this document to all interested stakeholders, including but not limited to, the Aegis development team, the client, and the department of Informatics at the University of Pretoria.