|  |  |  |
| --- | --- | --- |
|  |  | Del 2: functional specification |
|  | This functional specification document covers in depth how our system will meet the business requirements for our client  The overview of this document covers a full set of use case diagrams and narratives which breakdown how each subsystem will operate, process models to illustrate the different process within the system, along with a full set of Activity Diagrams. Complete Logical Data Model, Input and output descriptions, Validation of the functional specifications against the requirements and a review of the complexity marks then concluding with a sign-off by the client and the team. |
| client information NdilaTransfers;  A company based in Gauteng and Western Cape with the aim of supplying affordable, professional, safe and reliable shuttle services to the South African tourism industry and also to the corporate industry.  Our services are reliant on the following values Respect, Reliability, Integrity, Flexibility and Professionalism. We have the capacity to transport our esteemed clients safely to any respective destination from either Gauteng or Western Cape.   CONTACT PHONE:  Mr. Thendo  061 402 6227  WEBSITE:  [www.ndilatransfers.com](http://www.ndilatransfers.com)  EMAIL:  Thendo@Ndilatransfers.com |  | the team : Group 20 |
|  | From left to right:  Kwena Maboka, Mpho Mosotho, Nondumiso Mahlangu (**Team Leader**), Paballo Matabane, Mninikhaya Mavundla  **1. u14059867 |** [**u14059867@tuks.co.za**](mailto:u14059867@tuks.co.za) **| 071 872 6779**  **2. u16056762 |** [**u16056762@tuks.co.za**](mailto:u16056762@tuks.co.za) **| 082 833 0426**  **3. u17150312 |** [**u17150312@tuks.co.za**](mailto:u17150312@tuks.co.za) **| 076 602 5272**  **4. u16186835 |** [**u16186835@tuks.co.za**](mailto:u16186835@tuks.co.za) **| 076 889 4071**  **5. u16146060 |** [**u16146060@tuks.co.za**](mailto:u16146060@tuks.co.za) **| 076 496 6076** |
|  |  |
|  |  |

|  |
| --- |
| TABLE OF CONTENTS |

[Del 2: functional specification 0](#_Toc8593170)

[client information 0](#_Toc8593171)

[CONTACT 0](#_Toc8593172)

[the team : Group 20 0](#_Toc8593173)

[TABLE OF CONTENTS 1](#_Toc8593174)

[1. deliverable 2 introduction 3](#_Toc8593175)

[Introduction 3](#_Toc8593176)

[1.1 Synopsis of the Previous Deliverable: Deliverable 1 3](#_Toc8593177)

[conclusion 3](#_Toc8593178)

[2. Use Cases 4](#_Toc8593179)

[Introduction 4](#_Toc8593180)

[USE CASE DIAGRAM 4](#_Toc8593181)

[Use case narritives 4](#_Toc8593182)

[Conclusion 78](#_Toc8593183)

[3. process modelling 79](#_Toc8593184)

[Introduction 79](#_Toc8593185)

[Conclusion 79](#_Toc8593186)

[4. uml modelling 80](#_Toc8593187)

[Introduction 80](#_Toc8593188)

[CONCLUSION 80](#_Toc8593189)

[5. data modelling 81](#_Toc8593190)

[Introduction 81](#_Toc8593191)

[Conclusion 81](#_Toc8593192)

[6. interfaces and other inputs 82](#_Toc8593193)

[Introduction 82](#_Toc8593194)

[4. Driver 83](#_Toc8593195)

[5. VEHICLE 84](#_Toc8593196)

[6. Vehicle Group 86](#_Toc8593197)

[conclusion 87](#_Toc8593198)

[7. reports and other outputs 88](#_Toc8593199)

[Introduction 88](#_Toc8593200)

[Outputs 88](#_Toc8593201)

[conclusion 93](#_Toc8593202)

[8. validation 94](#_Toc8593203)

[Introduction 94](#_Toc8593204)

[validation 94](#_Toc8593205)

[conclusion 101](#_Toc8593206)

[9. sign-off by client 102](#_Toc8593207)

[Introduction 102](#_Toc8593208)

[conclusion 102](#_Toc8593209)

[10. general 103](#_Toc8593210)

[Introduction 103](#_Toc8593211)

[conclusion 103](#_Toc8593212)

[11. SIGN OFF 104](#_Toc8593213)

[Introduction 104](#_Toc8593214)

[conclusion 104](#_Toc8593215)

|  |
| --- |
| deliverable 2 introduction |

## Introduction

The previous deliverable saw the team, K’Runa, present our project proposal. Briefly we will run through what the document contained, as it is the fundamental foundation for Deliverable 2, which is the Functional Specification.

## 1.1 Synopsis of the Previous Deliverable: Deliverable 1

This is the content that went into the first deliverable

* Client information
* Project Request
* Preliminary Investigation
* Problem Analysis
* Requirement Analysis
* Decision Analysis
* Appendices on client documents, other systems investigated and complexity requirements.
* Sign-off by Client
* Sign-off by Team

## conclusion

The following deliverable reveals the Functional Specification of the proposed system. This document will demonstrate what our system will be capable of doing but not focusing on how the system will perform it.

This document is crucial as it serves the purpose of enlightening the stakeholders with an in-depth understanding of the proposed new system, how all the requirements will be met and how the data will be manipulated to assist them to make better business decisions. A description of how the different actors are expected to interact with the system and what will system respond to the actors. This is portrayed through the System Analysis Methods such as a complete set of Use Case Diagrams along with Use Case Narratives to outline the roles of each user and a detailed step by step description of the system requirements will be met. Process Models to breakdown the flow of information such as the data input and output interacting between use cases, actors and entities. UML Modelling to illustrate the activity of the use case, the various decisions and the logical flow to complete the relevant steps towards achieving the requirement. A complete, fully-attributed logical Data Model demonstrating how the data will be structured and which relevant attributes will be stored.

|  |
| --- |
| 2. Use Cases |

## Introduction

This following part of the functional specification document comprises of two main sections. A complete set of UML Use Case Diagrams illustrating the actors and how they play their role in our system. Together with the second section, which is a list of Use Case Narratives detailing each activity and process within each use case. The Use Case Narratives explain each process within the Use Case -step by step detailing how the actors interact with the system and how the system responds also including alternative steps and relevant business rules.

## USE CASE DIAGRAM

## Use case narritives

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Kwena Maboka Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Create Access Level | | **USE CASE TYPE** |
| **USE CASE ID:** | 1.1 | | Business Requirements:🞎 |
| **PRIORITY:** | High | | **System Analysis:** **🞎** |
| **SOURCE:** |  | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Employee | | |
| **PRIMARY SYSTEM ACTOR** | None | | |
| **OTHER PARTICIPATING ACTORS:** | * None | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * Owner | | |
| **DESCRIPTION:** | This use case describes the events of an employee creating a new access level into the system database. | | |
| **PRE-CONDITION:** | Access Level should not already exist in the system database  Employee should have authority to create an access level. | | |
| **TRIGGER:** | Employee | | |
| **TYPICAL COURSE OF EVENTS:** | **Step 1**: Employee would like to create a new access level and selects the option to create access level | Step 2: System checks if user has authority to create an access level | |
|  |  | Step 3: Request Employee to provide details about the new access level such as:  “AccessLevel\_ID”  “AccessLevel\_Name”  “AccessLevel\_Description” | |
|  | Step 4: Employee enters details about the new access level | Step 5: System verifies details provided by the Employee | |
|  |  | **Step 6:** System creates new access level into system database | |
|  |  | Step 7: Systems notifies the employee that the new access level has successfully been added. | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
| **ALTERNATE COURSES:** | [Alt Step 2]: System notifies the employee that they do not have the authority to create an access level. Use case terminates, | | |
|  | [Alt Step 5] : System verification of the details provided by the employee fails and return to step 4 | | |
|  |  | | |
| **CONCLUSION:** | New access level has been created and stored into the system database | | |
| **POST-CONDITION:** | None. | | |
| **BUSINESS RULES** | * An employee without the authority may not create an access level | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * None. | | |
| **ASSUMPTIONS:** | * Access level does not exist already in the database | | |
| **OPEN ISSUES:** | None. | | |

\

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Kwena Maboka Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Update Access Level | | **USE CASE TYPE** |
| **USE CASE ID:** | 1.2 | | Business Requirements:🞎 |
| **PRIORITY:** | High | | **System Analysis:** **🞎** |
| **SOURCE:** |  | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Employee | | |
| **PRIMARY SYSTEM ACTOR** | None | | |
| **OTHER PARTICIPATING ACTORS:** | * None | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * Owner | | |
| **DESCRIPTION:** | This use case describes the events of an employee updating an access level into the system database. | | |
| **PRE-CONDITION:** | Access Level should not already exist in the system database  Employee should have authority to create an access level. | | |
| **TRIGGER:** | Employee | | |
| **TYPICAL COURSE OF EVENTS:** | **Step 1**: The employee would like to update an existing access level and selects the option to update access level | Step 2: System checks if user has authority to update an access level | |
|  |  | Step 3: System Display all existing access levels  And prompts the employee to select the access level they wish to update | |
|  | Step 4: Employee selects the access level they’d like to update | Step 5: The system then displays the attributes that the employee can edit  AccessLevel\_ID”  “AccessLevel\_Name”  “AccessLevel\_Description” | |
|  | Step 6: Employee updates the details they’d like update:  AccessLevel\_ID”  “AccessLevel\_Name”  “AccessLevel\_Description” |  | |
|  | Step 7: Employee selects the save option to save the updated details into the system database | **Step 8: System verifies the details updated by the employee** | |
|  |  | Step 9: System then saves the updated access level details into the system database. | |
|  |  | Step 10: System notifies the employee that the update has been successfully. | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
| **ALTERNATE COURSES:** | [Alt Step 2]: System notifies the employee that they do not have the authority to update an access level. Use case terminates, | | |
|  | [Alt Step 8] : System verification of the details provided by the employee fails and return to step 4 | | |
|  |  | | |
|  |  | | |
| **CONCLUSION:** | Access level has been updated and stored into the system database | | |
| **POST-CONDITION:** | None. | | |
| **BUSINESS RULES** | * An employee without the authority may not update an access level | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * None. | | |
| **ASSUMPTIONS:** | * Access level does exist already in the database | | |
| **OPEN ISSUES:** | None. | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Kwena Maboka Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Search Access Level | | **USE CASE TYPE** |
| **USE CASE ID:** | 1.3 | | Business Requirement:🞎 |
| **PRIORITY:** | High | | **System Analysis:** **🞎** |
| **SOURCE:** |  | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Employee | | |
| **PRIMARY SYSTEM ACTOR** | None | | |
| **OTHER PARTICIPATING ACTORS:** | * None | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * None | | |
| **DESCRIPTION:** | This use case describes the events where an employee is looking for a certain Access level. It involves the employee entering the access level name so that the system can return the access level that has been searched for by the employee. | | |
| **PRE-CONDITION:** | Access level already exists in the system database | | |
| **TRIGGER:** | Employee | | |
| **TYPICAL COURSE OF EVENTS:** | **Step 1**:The employee selects the ‘search access level’ option | Step 2: System checks if the employee has authority to search access levels | |
|  |  | Step 3: System requests the employee to enter the details of the access level to search for  AccessLevel\_ID”  “AccessLevel\_Name”  “AccessLevel\_Description” | |
|  | Step 4: Employee enters the access level details for which they would like to search for:  AccessLevel\_ID”  “AccessLevel\_Name”  “AccessLevel\_Description” | Step 5: System validates the details entered by the employee | |
|  |  | **Step 6: System then checks system database for a access level that the employee is searching for** | |
|  |  | Step 7: System then displays the access level that has been searched for | |
|  |  |  | |
|  |  |  | |
| **ALTERNATE COURSES:** | [Alt Step 2]: System notifies the employee that they do not have the authority to search an access level. Use case terminates, | | |
|  | [Alt Step 5] : System verification of the details provided by the employee fails and return to step 3 | | |
|  | [Alt Step 7]: System notifies the employee that the search could not find the access level they have searched for | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
| **CONCLUSION:** | Access level search has been found in the system database | | |
| **POST-CONDITION:** | None. | | |
| **BUSINESS RULES** | * An employee without the authority may not search for an access level | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * None. | | |
| **ASSUMPTIONS:** | * Access level does exist already in the database | | |
| **OPEN ISSUES:** | None. | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Kwena Maboka Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Register New Employee | | **USE CASE TYPE** |
| **USE CASE ID:** | 2.1 | | Business Requirements:🞎 |
| **PRIORITY:** | High | | **System Analysis:** **🞎** |
| **SOURCE:** |  | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Employee | | |
| **PRIMARY SYSTEM ACTOR** | Manager | | |
| **OTHER PARTICIPATING ACTORS:** | * None | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * Owner | | |
| **DESCRIPTION:** | This use case describes the events where a manager would like to register a new employee into the system database. This involves the entering the new employee details into the system and verifying the entered details. | | |
| **PRE-CONDITION:** | Employee does not already exist in the current system database. | | |
| **TRIGGER:** | Employee | | |
| **TYPICAL COURSE OF EVENTS:** | **Step 1**: Manager selects the register new employee option | Step 2: System checks whether the manager has authority to register a new employee | |
|  |  | Step 3: System then request the manager to enter all employee details into the system:  “EMPID”  “EMP\_Name”  “EMP\_Surname”  “AuditID”  “EMP\_TypeID”  “EMP\_EmailAddress”  “EMP\_Contact”  “EMP\_Password”  “EMP\_IDNum”  “TitleID”  “GenderID” | |
|  | Step 4: Manager enters all required employee details to register a new employee  “EMPID”  “EMP\_Name”  “EMP\_Surname”  “AuditID”  “EMP\_TypeID”  “EMP\_EmailAddress”  “EMP\_Contact”  “EMP\_Password”  “EMP\_IDNum”  “TitleID”  “GenderID” |  | |
|  | Step 5: Manager selects the save option to save the entered details | **Step 6: System then verifies all entered details by the manager** | |
|  |  | Step 7: System then adds the entered employee details into the system database. | |
|  |  | Step 8: System notifies the manager that the new employee has successfully been added. | |
|  |  |  | |
|  |  |  | |
| **ALTERNATE COURSES:** | [Alt Step 2]: [Alt Step 2]: System notifies the employee that they do not have the authority to register a new employee. Use case terminates. | | |
|  | [Alt Step 6]: System verification of the details provided by the manager fails and return to step 3 | | |
|  |  | | |
|  |  | | |
| **CONCLUSION:** | The manager has successfully register a new employee into the system | | |
| **POST-CONDITION:** | None | | |
| **BUSINESS RULES** | * Only the owner and the managers are allowed to register new employees | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * None | | |
| **ASSUMPTIONS:** | * None | | |
| **OPEN ISSUES:** | None | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Update Employee | | **USE CASE TYPE** |
| **USE CASE ID:** | 2.2 | | Business Requirements:🞎 |
| **PRIORITY:** | High | | **System Analysis:** **🞎** |
| **SOURCE:** |  | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Employee | | |
| **PRIMARY SYSTEM ACTOR** | None | | |
| **OTHER PARTICIPATING ACTORS:** | * None | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * None | | |
| **DESCRIPTION:** | This use case describes the events where a manager would like to update employee’s details in the system database. This involves the manger selecting the employee they wish to update as well as entering the new details of this employee. | | |
| **PRE-CONDITION:** | An employee should not already exist in the system database  The owner and the managers should have authority to update an employee’s details | | |
| **TRIGGER:** | Employee | | |
| **TYPICAL COURSE OF EVENTS:** | **Step 1**: The employee would like to update an existing employee and selects the option to update employee | Step 2: System checks if user has authority to update an employee | |
|  |  | Step 3: System Display all existing employees  And prompts the employee to select the employee they wish to update | |
|  | Step 4: Employee selects the employee they’d like to update | Step 5: The system then displays the attributes that the employee can edit  EMP\_Name”  “EMP\_Surname”  “EMP\_TypeID”  “EMP\_EmailAddress”  “EMP\_Contact”  “EMP\_Password”  “EMP\_IDNum”  “TitleID”  “GenderID” | |
|  | Step 6: Employee updates the details they’d like update  EMP\_Name”  “EMP\_Surname”  “EMP\_TypeID”  “EMP\_EmailAddress”  “EMP\_Contact”  “EMP\_Password”  “EMP\_IDNum”  “TitleID”  “GenderID” |  | |
|  | Step 7: Employee selects the save option to save the updated details into the system database | **Step 8: System verifies the details updated by the employee**  EMP\_Name”  “EMP\_Surname”  “EMP\_TypeID”  “EMP\_EmailAddress”  “EMP\_Contact”  “EMP\_Password”  “EMP\_IDNum”  “TitleID”  “GenderID” | |
|  |  | Step 9: System then saves the updated employee details into the system database. | |
|  |  | Step 10: System notifies the employee that the update has been successfully. | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
| **ALTERNATE COURSES:** | [Alt Step 2]: System notifies the employee that they do not have the authority to update an employee’s details. Use case terminates, | | |
|  | [Alt Step 8] : System verification of the details provided by the employee fails and return to step 4 | | |
|  |  | | |
|  |  | | |
| **CONCLUSION:** | Employee has been updated and stored into the system database | | |
| **POST-CONDITION:** | None. | | |
| **BUSINESS RULES** | * An employee without the authority may not update an employee’s details | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * None. | | |
| **ASSUMPTIONS:** | * Employee does exist already in the database | | |
| **OPEN ISSUES:** | None. | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Kwena Maboka Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Search Employee | | **USE CASE TYPE** |
| **USE CASE ID:** | 2.3 | | Business Requirements:🞎 |
| **PRIORITY:** | High | | **System Analysis:** **🞎** |
| **SOURCE:** |  | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Manager | | |
| **PRIMARY SYSTEM ACTOR** | None | | |
| **OTHER PARTICIPATING ACTORS:** | * None | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * None | | |
| **DESCRIPTION:** | This use case describes the events where the manager is looking for a specific employee. It involves the manager entering the employee details so that the system can return the employee that has been searched for by the manager. | | |
| **PRE-CONDITION:** | Employee already exists in the system database | | |
| **TRIGGER:** | Manager | | |
| **TYPICAL COURSE OF EVENTS:** | **Step 1**:The manager selects the ‘search employee’ option | **Step 2**: System checks if the manager has authority to search employees | |
|  |  | **Step 3**: System requests the manager to enter the details of the employee to search for  EMP\_Name”  “EMP\_Surname”  “EMP\_TypeID | |
|  | **Step 4:** Manager enters the employee details for which they would like to search for  EMP\_Name”  “EMP\_Surname”  “EMP\_TypeID | **Step 5**: System validates the details entered by the manager | |
|  |  | **Step 6: System then checks system database for the employee that the** manager **is searching for** | |
|  |  | Step 7: System then displays the employee that has been searched for  “EMP\_Name”  “EMP\_Surname”  “EMP\_TypeID”  “EMP\_EmailAddress”  “EMP\_Contact”  “EMP\_Password”  “EMP\_IDNum”  “TitleID”  “GenderID” | |
|  |  |  | |
|  |  |  | |
| **ALTERNATE COURSES:** | [Alt Step 2]: System notifies the manager that they do not have the authority to search an employee. Use case terminates, | | |
|  | [Alt Step 5] : System verification of the details provided by the manager fails and return to step 3 | | |
|  | [Alt Step 7]: System notifies the manager that the search could not find the employee they have searched for | | |
|  |  | | |
|  |  | | |
| **CONCLUSION:** | Employee search has been found in the system database | | |
| **POST-CONDITION:** | None | | |
| **BUSINESS RULES** | None. | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** |  | | |
| **ASSUMPTIONS:** | * Employee does exist already in the database | | |
| **OPEN ISSUES:** |  | | |
|  | None. | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Kwena Maboka Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Create New Employee Type | | **USE CASE TYPE** |
| **USE CASE ID:** | 3.1 | | Business Requirements:🞎 |
| **PRIORITY:** | High | | **System Analysis:** **🞎** |
| **SOURCE:** |  | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Employee | | |
| **PRIMARY SYSTEM ACTOR** | Manager | | |
| **OTHER PARTICIPATING ACTORS:** | * None | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * Owner | | |
| **DESCRIPTION:** | This use case describes the events where a manager would like to register a new type into the system database. This involves the entering the new employee type details into the system and verifying the entered details. | | |
| **PRE-CONDITION:** | Employee type does not already exist in the current system database. | | |
| **TRIGGER:** | Employee | | |
| **TYPICAL COURSE OF EVENTS:** | **Step 1**: Manager selects the create new employee type option | Step 2: System checks whether the manager has authority to create a new employee type | |
|  |  | Step 3: System then request the manager to enter all employee type details into the system  “EMP\_TypeID”  “EMP\_TypeName”  “EMP\_TypeDes” | |
|  | Step 4: Manager enters all required employee type details to register a new employee  “EMP\_TypeID”  “EMP\_TypeName”  “EMP\_TypeDes” |  | |
|  | Step 5: Manager selects the save option to save the entered details | **Step 6: System then verifies all entered details by the manager** | |
|  |  | Step 7: System then adds the entered employee type details into the system database. | |
|  |  | Step 8: System notifies the manager that the new employee type has successfully been added. | |
|  |  |  | |
|  |  |  | |
| **ALTERNATE COURSES:** | [Alt Step 2]: System notifies the employee that they do not have the authority to register a new employee type. Use case terminates. | | |
|  | [Alt Step 6]: System verification of the details provided by the manager fails and return to step 3  EMP\_TypeID”  “EMP\_TypeName”  “EMP\_TypeDes” | | |
|  |  | | |
|  |  | | |
| **CONCLUSION:** | The manager has successfully register a new employee type into the system | | |
| **POST-CONDITION:** | None | | |
| **BUSINESS RULES** | * Only the owner and the managers are allowed to register new employee type | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * None | | |
| **ASSUMPTIONS:** | * None | | |
| **OPEN ISSUES:** | None | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Kwena Maboka Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Update Employee Type | | **USE CASE TYPE** |
| **USE CASE ID:** | 3.2 | | Business Requirements:🞎 |
| **PRIORITY:** | High | | **System Analysis:** **🞎** |
| **SOURCE:** |  | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Employee | | |
| **PRIMARY SYSTEM ACTOR** | None | | |
| **OTHER PARTICIPATING ACTORS:** | * None | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * None | | |
| **DESCRIPTION:** | This use case describes the events where a manager would like to update an employee type details in the system database. This involves the manger selecting the employee type they wish to update as well as entering the new details of this employee type | | |
| **PRE-CONDITION:** | An employee type should not already exist in the system database  The owner and the managers should have authority to update an employee type details | | |
| **TRIGGER:** | Employee | | |
| **TYPICAL COURSE OF EVENTS:** | **Step 1**: The employee would like to update an existing employee type and selects the option to update employee type | Step 2: System checks if user has authority to update an employee type | |
|  |  | Step 3: System Display all existing employee types  And prompts the employee to select the employee type they wish to update | |
|  | Step 4: Employee selects the employee type they’d like to update | Step 5: The system then displays the attributes that the employee can edit  EMP\_TypeName”  “EMP\_TypeDes” | |
|  | Step 6: Employee updates the details they’d like update  “EMP\_TypeName”  “EMP\_TypeDes” |  | |
|  | Step 7: Employee selects the save option to save the updated details into the system database | **Step 8: System verifies the details updated by the employee** | |
|  |  | Step 9: System then saves the updated employee type details into the system database.  EMP\_TypeName”  “EMP\_TypeDes” | |
|  |  | Step 10: System notifies the employee that the update has been successfully. | |
|  |  |  | |
| **ALTERNATE COURSES:** | [Alt Step 2]: System notifies the employee that they do not have the authority to update an employee type details. Use case terminates, | | |
|  | [Alt Step 8] : System verification of the details provided by the employee fails and return to step 4  “EMP\_TypeName”  “EMP\_TypeDes | | |
|  |  | | |
|  |  | | |
| **CONCLUSION:** | Employee type has been updated and stored into the system database | | |
| **POST-CONDITION:** | None. | | |
| **BUSINESS RULES** | * An employee without the authority may not update an employee type details | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * None. | | |
| **ASSUMPTIONS:** | * Employee type does exist already in the database | | |
| **OPEN ISSUES:** | None. | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Kwena Maboka Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Search Employee Type | | **USE CASE TYPE** |
| **USE CASE ID:** | 3.3 | | Business Requirements:🞎 |
| **PRIORITY:** | High | | **System Analysis:** **🞎** |
| **SOURCE:** |  | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Manager | | |
| **PRIMARY SYSTEM ACTOR** | None | | |
| **OTHER PARTICIPATING ACTORS:** | * None | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * None | | |
| **DESCRIPTION:** | This use case describes the events where the manager is looking for a specific employee type. It involves the manager entering the employee type details so that the system can return the employee type that has been searched for by the manager. | | |
| **PRE-CONDITION:** | Employee type already exists in the system database | | |
| **TRIGGER:** | Manager | | |
| **TYPICAL COURSE OF EVENTS:** | **Step 1**:The manager selects the ‘search employee type option | Step 2: System checks if the manager has authority to search employee type | |
|  |  | Step 3: System requests the manager to enter the details of the employee type to search for  “EMP\_TypeID”  “EMP\_TypeName”  “EMP\_TypeDes” | |
|  | Step 4: Manager enters the employee type details for which they would like to search for  “EMP\_TypeID”  “EMP\_TypeName”  “EMP\_TypeDes” | Step 5: System validates the details entered by the manager | |
|  |  | **Step 6: System then checks system database for an** employee type **that the** manager **is searching for** | |
|  |  | Step 7: System then displays the employee type that has been searched for  “EMP\_TypeID”  “EMP\_TypeName”  “EMP\_TypeDes” | |
|  |  |  | |
|  |  |  | |
| **ALTERNATE COURSES:** | [Alt Step 2]: System notifies the manager that they do not have the authority to search an employee type. Use case terminates, | | |
|  | [Alt Step 5] : System verification of the details provided by the manager fails and return to step 3  “EMP\_TypeID”  “EMP\_TypeName”  “EMP\_TypeDes” | | |
|  | [Alt Step 7]: System notifies the manager that the search could not find the employee type they have searched for | | |
|  |  | | |
|  |  | | |
| **CONCLUSION:** | Employee type search has been found in the system database | | |
| **POST-CONDITION:** | None | | |
| **BUSINESS RULES** | None. | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * None | | |
| **ASSUMPTIONS:** | * Employee type does exist already in the database | | |
| **OPEN ISSUES:** | * None. | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Kwena Maboka Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Remove Employee Type | | **USE CASE TYPE** |
| **USE CASE ID:** | 3.4 | | Business Requirements:🞎 |
| **PRIORITY:** | High | | **System Analysis:** **🞎** |
| **SOURCE:** |  | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Manager | | |
| **PRIMARY SYSTEM ACTOR** | None | | |
| **OTHER PARTICIPATING ACTORS:** | * None | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * Owner | | |
| **DESCRIPTION:** | This use case describes the events where a manager would like to remove an employee type from being used for any new transactions. This involves the manager selecting the employee type to remove and confirming that the employee type should not be accessible any longer. | | |
| **PRE-CONDITION:** | The employee type should already exist in the system database | | |
| **TRIGGER:** | Manager/Owner | | |
| **TYPICAL COURSE OF EVENTS:** | **Step 1**: The manager selects the ‘remove employee type’ option | Step 2: System checks to see if the manager has authority to remove an employee type | |
|  |  | Step 3: System then displays all the employee types in the database  “EMP\_TypeID”  “EMP\_TypeName”  “EMP\_TypeDes” | |
|  |  | Step 4: System prompts the manager to select the employee type they wish to remove | |
|  | Step 5: Manager selects the employee type they would like to remove  “EMP\_TypeID”  “EMP\_TypeName”  “EMP\_TypeDes” | **Step 6: System requests confirmation that the selected employee type is the correct one to be removed** | |
|  | Step 7: Manager selects the correct option | Step 8: System then disables the employee type to be used again in the database | |
|  |  | Step 9: System then displays a confirmation message to let the manager that the removal has been successful | |
|  |  |  | |
|  |  |  | |
| **ALTERNATE COURSES:** | [Alt Step 2]: System finds the manager to not have authority to remove an employee type. Terminates use case | | |
|  | [Alt Step 8]: System fails to remove the employee type and returns to step 4 | | |
|  |  | | |
|  |  | | |
| **CONCLUSION:** | The manager selected employee type has successfully been removed from the system | | |
| **POST-CONDITION:** | Employee Type still exists In previous records | | |
| **BUSINESS RULES** | * None | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * None | | |
| **ASSUMPTIONS:** | * Employee Type already exists in the system database | | |
| **OPEN ISSUES:** | None | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Nondumiso Mahlangu Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Search driver | | **USE CASE TYPE** |
| **USE CASE ID:** | 4.1 | | Business Requirements:◻ System Analysis:◻ System Design: ◻ |
| **PRIORITY:** | High | |
| **SOURCE:** | Ndila Transfers | |
| **PRIMARY BUSINESS ACTOR** | Booking consultant | | |
| **PRIMARY SYSTEM ACTOR** | None | | |
| **OTHER PARTICIPATING ACTORS:** | * None | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * None | | |
| **DESCRIPTION:** | This use case describes an event where booking consultant wants to search a driver on the system and enters the Driver name they want to search and retrieve the details of the driver such as the driver name, surname, license number and license type from the Driver table in the database. | | |
| **PRE-CONDITION:** | The booking consultant has to be logged onto the system | | |
| **TRIGGER:** | The booking consultant selects the Search Driver option on the system | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**: The booking consultant selects the Search Driver option of the driver on the system | **Step 2**: The system displays a page for the booking system to enter the search criteria. | |
|  | **Step 3**: The booking consultant enters the characters of the name of the driver they want to search | **Step 4:** The system verifies the character input | |
|  |  | **Step 5:** The system reads from the **Driver** table the  **Driver\_Name** | |
|  |  | **Step 6**: The system compares the characters received with the characters in the **Driver** table in the **Driver\_Name** field until a match is found | |
|  |  | **Step 7**: The system retrieves the  **Driver\_ID,**  **Driver\_Name,**  **Driver\_Surname,**  **Driver\_LicenceNumber,**  **Driver\_LincenceType,**  **EMPID**  information related to the characters given from the **Driver** Table | |
|  |  | **Step 8:** The system displays the driver information retrieved | |
| **ALTERNATE COURSES:** | Alt step 3: The characters given did not match system requirements and an error message is displayed | | |
|  | Alt step 6: A match is not found and an alert message is displayed | | |
| **CONCLUSION:** | The system retrieves the information related to the Driver characters given | | |
| **POST-CONDITION:** | The driver information searched has to be displayed for the booking consultant | | |
| **BUSINESS RULES** | * The booking consultant and the team leader work with the driver | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * None | | |
| **ASSUMPTIONS:** | * None | | |
| **OPEN ISSUES:** | None | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Nondumiso Mahlangu Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Check driver availability | | **USE CASE TYPE** |
| **USE CASE ID:** | 4.2 | | Business Requirements:◻ System Analysis: ◻ System Design: ◻ |
| **PRIORITY:** | High | |
| **SOURCE:** | Ndila transfers | |
| **PRIMARY BUSINESS ACTOR** | Booking consultant | | |
| **PRIMARY SYSTEM ACTOR** | None | | |
| **OTHER PARTICIPATING ACTORS:** | * None | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * Owner * Driver supervisor | | |
| **DESCRIPTION:** | This use case describes an instance where the booking consultant wants to confirm a trip and checks the whether or not a driver is available. The booking consultant can then select the check availability option and the system returns a list of drivers which are available. | | |
| **PRE-CONDITION:** | The booking consultant has to be logged onto the system | | |
| **TRIGGER:** | The booking consultant wants to check whether or not a driver is available | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**: The booking consultant wants to check whether or not a driver is available and selects the Check Driver option | **Step 2:** The system retrieves the  **Trip Date**  **TripTime**  From the information given through use case | |
|  |  | **Step 3:** The system reads from the Slot table the: **Slot\_Date**  **Slot\_Time** | |
|  |  | **Step 4**: The system compares the  **Slot\_Date** and **Slot\_Time**  To the **Trip Date** and **Trip Time** and returns a list of the  **Driver\_ID’**s and  **Driver\_Name** whose **Slot\_Date** and **Slot\_Time** did not match the **Trip Date** and **Trip Time** | |
|  |  | **Step 5:** The system displays the list of drivers that have been returned | |
| **ALTERNATE COURSES:** | Alt step 4: There are no drivers available on the given date and time and a message is displayed that there are drivers available and the use case ends | | |
| **CONCLUSION:** | The system displays the drivers who are available | | |
| **POST-CONDITION:** | The schedule should be up to date at all times | | |
| **BUSINESS RULES** | * None | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * None | | |
| **ASSUMPTIONS:** | * None | | |
| **OPEN ISSUES:** | None | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Nondumiso Mahlangu Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Assign driver to trip | | **USE CASE TYPE** |
| **USE CASE ID:** | 4.3 | | Business Requirements:◻ System Analysis: ◻ System Design: ◻ |
| **PRIORITY:** | High | |
| **SOURCE:** | Ndila transfers | |
| **PRIMARY BUSINESS ACTOR** | Booking consultant | | |
| **PRIMARY SYSTEM ACTOR** | None | | |
| **OTHER PARTICIPATING ACTORS:** | * None | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * Team leader * Driver supervisor | | |
| **DESCRIPTION:** | This use case describes the event when the booking consultant wants to assign a specific driver to a vehicle. They check the availability of a driver and use case 4.2 Check Driver Availability is invoked. The booking consultant then chooses an available driver The name of the driver is then added on the slot. | | |
| **PRE-CONDITION:** | There has to be a trip to assign a driver to. | | |
| **TRIGGER:** | The booking consultant wants to assign a specific driver to a trip | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**: The booking consultant wants to assign a specific driver to a trip and selects assign driver option | **Step 2:** Invoke use case 4.2 | |
|  | **Step 3**: The booking consultant selects **Driver\_Name** to be booked | **Step 4**: The system verifies the information received | |
|  |  | **Step 5:** The system writes the **Driver\_Name** to the **Slot**. | |
|  |  | **Step 6:** The system displays a message confirming the assignment. | |
|  |  |  | |
|  |  |  | |
| **ALTERNATE COURSES:** | Alt step 2: The Use case returns a negative result, a message is displayed to the booking consultant and the use case ends | | |
|  |  | | |
| **CONCLUSION:** | The slot is updated | | |
| **POST-CONDITION:** | If a driver is unavailable, they cannot be assigned to another trip | | |
| **BUSINESS RULES** | * A vehicle has to be assigned to a booked trip before a driver can be assigned to it | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * None | | |
| **ASSUMPTIONS:** | * None | | |
| **OPEN ISSUES:** | None | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Nondumiso Mahlangu Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Outsource Driver | | **USE CASE TYPE** |
| **USE CASE ID:** | 4.4 | | Business Requirements:🞎 **System Analysis:** **🞎** System Design: 🞎 |
| **PRIORITY:** | High | |
| **SOURCE:** | Ndila Transfers | |
| **PRIMARY BUSINESS ACTOR** | Booking consultant | | |
| **PRIMARY SYSTEM ACTOR** | None | | |
| **OTHER PARTICIPATING ACTORS:** | * None | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * Team leader * Driver supervisor | | |
| **DESCRIPTION:** | This use case describes the event when the booking consultant wants to assign a specific driver to a vehicle. The booking consultant prompts the system to display the outsourced driver who is then added on the slot. | | |
| **PRE-CONDITION:** | The booking consultant has to be logged onto the system | | |
| **TRIGGER:** | The booking consultant wants to assign a specific driver to a trip | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**: The booking consultant wants to assign a driver to a trip and selects Outsource Driver option | **Step 2:** The system reads and displays information of:  **OutSoucreDriver\_ID**  **OutSource\_Driver\_Name**  **OutSource\_Driver\_Surname**  **LicenceType\_ID** from the **OutSourced\_Driver** table | |
|  | **Step 3**: The booking consultant selects the **driver** desired to be booked | **Step 4:** The system writes the **OutSoucreDriver\_ID**  and **OutSource\_Driver\_Name** to the **Slot**. | |
|  |  | **Step 5:** The system displays a message confirming the assignment | |
| **ALTERNATE COURSES:** | None | | |
|  |  | | |
| **CONCLUSION:** | The slot is updated | | |
| **POST-CONDITION:** | If a driver is unavailable, they cannot be assigned to another trip | | |
| **BUSINESS RULES** | * There has to be a pre-existing booking before a vehicle can be assigned | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * None | | |
| **ASSUMPTIONS:** | * None | | |
| **OPEN ISSUES:** | None | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Nondumiso Mahlangu Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Add vehicle | | **USE CASE TYPE** |
| **USE CASE ID:** | 5.1 | | Business Requirements: ◻ System Analysis: ◻ System Design: ◻ |
| **PRIORITY:** | High | |
| **SOURCE:** | Ndila Transfers | |
| **PRIMARY BUSINESS ACTOR** | Booking consultant | | |
| **PRIMARY SYSTEM ACTOR** | Booking consultant | | |
| **OTHER PARTICIPATING ACTORS:** | * None | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * None | | |
| **DESCRIPTION:** | This use case describes when the booking consultant wants to add another vehicle onto the system database. They retrieve all the relevant vehicle information and a new vehicle is saved onto the system database. | | |
| **PRE-CONDITION:** | The vehicle cannot be an existing vehicle in the system | | |
| **TRIGGER:** | The booking consultant wants to add another vehicle | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**: The booking consultant wants to add another vehicle and selects the Add Vehicle option. | **Step 2**: The system displays a page where the booking consultant can add the required details of the vehicle such as: **Vehicle\_Model**  **Vehicle\_Make\_Name**  **Vehicle\_Make\_Description**  **Vehicle\_Colour**  **ServiceProvider** | |
|  | **Step 3**: The booking consultant adds the necessary details of the new vehicle | **Step 4**: The system verifies the information gathered | |
|  |  | **Step 5**: The system creates a new  **Vehicle\_ID,**  **VehicleMake\_ID,**  **VehicleMaintenance\_ID**  by retrieving the last line in  **Vehicle\_ID,**  **VehicleMake\_ID,**  **VehicleMaintenance\_ID**  in the **Vehicle, VehicleMake,** **VehicleMaintenance**  table, increments by 1 and assigns a new **Vehicle\_ID**  **VehicleMake\_ID,**  **VehicleMaintenance\_ID** | |
|  |  | **Step 6:** The system adds the new **Vehicle\_ID VehicleMake\_ID,** **VehicleMaintenance\_ID** and  the given **Vehicle\_Model**, **Vehicle\_Colour\_Description**  **Vehicle\_Make\_Name**,  **Vehicle\_Make\_Description**  **Vehicle\_ID**  **Se**r**viceProvider** into the **Vehicle**, **Vehicle\_Make, VehicleMaintenance** tables | |
|  |  | **Step 7**: The system displays a message that a new vehicle has been added to the system | |
|  |  |  | |
| **ALTERNATE COURSES:** | Alt step 4: The booking consultant made an error with the entered information and an error message is provided | | |
|  |  | | |
| **CONCLUSION:** | A new vehicle is added onto the system database | | |
| **POST-CONDITION:** | The vehicle should be ready to book and use | | |
| **BUSINESS RULES** | * None | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * None | | |
| **ASSUMPTIONS:** | * None | | |
| **OPEN ISSUES:** | None | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Nondumiso Mahlangu Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Search vehicle | | **USE CASE TYPE** |
| **USE CASE ID:** | 5.2 | | Business Requirements: ◻ System Analysis:◻ System Design: ◻ |
| **PRIORITY:** | High | |
| **SOURCE:** | Ndila Transfers | |
| **PRIMARY BUSINESS ACTOR** | Booking consultant | | |
| **PRIMARY SYSTEM ACTOR** | None | | |
| **OTHER PARTICIPATING ACTORS:** | * None | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * Team Leader * Driver supervisor | | |
| **DESCRIPTION:** | This use case describes an event where booking consultant wants to search for a vehicle on the system and enters the vehicle name they want to search and retrieve the details of the vehicle from the Vehicle table in the database. | | |
| **PRE-CONDITION:** | The booking consultant has to be logged onto the system | | |
| **TRIGGER:** | The booking consultant selects the Search Vehicle option on the system | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**: The booking consultant selects the Search Vehicle option of the driver on the system | **Step 2**: The system displays a page for the booking system to enter the search criteria. | |
|  | **Step 3**: The booking consultant enters the characters of the vehicle they want to search | **Step 4:** The system verifies the character input | |
|  |  | **Step 5:** The system reads from the **Vehicle** table the  **Vehicle\_Model** | |
|  |  | **Step 6**: The system compares the characters received with the characters in the Vehicle table in the **Vehicle\_Model** field until a match is found | |
|  |  | **Step 7**: The system retrieves the  **Vehicle\_ID**  **Vehicle\_Model, the VehicleMake\_ID,**  **VehicleMaintenance\_ID,**  **VehicleColour\_ID,**  **VehicleLicencePlates** information related to the characters given from the **Vehicle** Table | |
|  |  | **Step 8:** The system displays the retrieved information. | |
| **ALTERNATE COURSES:** | Alt step 4: The characters given did not match system requirements and an error message is displayed | | |
|  | Alt step 6: A match is not found and an alert message is displayed | | |
| **CONCLUSION:** | The system retrieves the information related to the Vehicle characters given | | |
| **POST-CONDITION:** | The vehicle information searched has to be displayed for the booking consultant | | |
| **BUSINESS RULES** | * The booking consultant and the team leader work with the driver | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * None | | |
| **ASSUMPTIONS:** | * None | | |
| **OPEN ISSUES:** | None | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Nondumiso Mahlangu Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Confirm Trip | | **USE CASE TYPE** |
| **USE CASE ID:** | 5.3 | | Business Requirements:◻ System Analysis: ◻ System Design: ◻ |
| **PRIORITY:** | High | |
| **SOURCE:** | Ndila Transfers | |
| **PRIMARY BUSINESS ACTOR** | Booking consultant | | |
| **PRIMARY SYSTEM ACTOR** | None | | |
| **OTHER PARTICIPATING ACTORS:** | * None | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * Team Leader * Driver supervisor * Passenger * Driver | | |
| **DESCRIPTION:** | This use case describes the event where the trip information such as the driver name, driver surname, departure point and destination point as well as the vehicle information is sent to the relevant people to inform them of the trip details. The system sends an SMS to the passenger and the driver | | |
| **PRE-CONDITION:** | The trip has to be an existing trip | | |
| **TRIGGER:** | The booking consultant wants to send out confirmation messages | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**: The booking consultant wants to send out a confirmation message and clicks on the Confirm booking option | Step 2: The system retrieves the current date and time and increments the day of the current day by 1 | |
|  |  | Step 3: The system reads from the **Slot** table in the database the following fields:  **BookingReference\_ID**  **Passenger\_ID,**  **Driver\_ID,**  **DeparturePoint,**  **DestinationPoint**  **Time**  for every line | |
|  |  | Step 4: The system retrieves the  **Passenger\_Tel** in the **Passenger** table and the  **Driver\_Contact**  **Driver\_Name**  from the **Driver** table | |
|  |  | Step 5: The system sends a confirmation message to the contact number of the **Passenger** with the:  **Booking reference,**  **Driver name,**  **Departure point,**  **Destination point and**  **Time**  of trip via SMS | |
|  |  | Step 6: The system runs a function that reads the first  **Driver\_ID**  **Driver\_Contact**  **AddedToLog**  line in **Slot** Table, compares it to the rest and stores it in a temporary function | |
|  |  | Step 7: The system sends a message to the **Driver\_Contact** with the following:  **Passenger\_Name,**  **Departure point,**  **Destination Point**  **Time**  of trip via SMS | |
|  |  | Step 8: The system displays a message to that the trips have been confirmed | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
| **ALTERNATE COURSES:** | None | | |
|  |  | | |
| **CONCLUSION:** | A confirmation message of the trip is sent to the Passenger and the Driver | | |
| **POST-CONDITION:** | The driver and the passenger must be aware of the confirmation of the trip | | |
| **BUSINESS RULES** | * The booking consultant sends out the trips at the end of a business day | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * None | | |
| **ASSUMPTIONS:** | * None | | |
| **OPEN ISSUES:** | None | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Nondumiso Mahlangu Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Check Vehicle Availability | | **USE CASE TYPE** |
| **USE CASE ID:** | 5.4 | | Business Requirements:🞎 **System Analysis:** **🞎** System Design: 🞎 |
| **PRIORITY:** | High | |
| **SOURCE:** | Ndila Transfers | |
| **PRIMARY BUSINESS ACTOR** | Booking consultant | | |
| **PRIMARY SYSTEM ACTOR** | None | | |
| **OTHER PARTICIPATING ACTORS:** | * None | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * Owner * Driver supervisor | | |
| **DESCRIPTION:** | This use case describes an instance where the booking consultant wants to confirm a trip and checks the whether or not a vehicle is available. The booking consultant can then select the check availability option and the system returns a list of Vehicles which are available. | | |
| **PRE-CONDITION:** | The booking consultant has to be logged onto the system | | |
| **TRIGGER:** | The booking consultant wants to check whether or not a vehicle is available | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**: The booking consultant wants to check whether or not a driver is available and selects the Check availability option | **Step 2:** The system retrieves the  **Trip Date**  **TripTime**  **Number\_of\_Passengers**  From the **Booking** table | |
|  |  | **Step 3:** The system reads from the **Slot** table the: **Slot\_Date**  **Slot\_Time** and the  **Number\_of\_Passengers** from the **Vehicle Group** table | |
|  |  | **Step 4**: The system compares the  **Slot\_Date** and **Slot\_Time**  To the **Trip Date** and **Trip Time** and returns a list of the  **Vehicle\_ID’**s and  **Vehicle\_Model**  **Number\_of\_Passengers** whose **Slot\_Date** and **Slot\_Time** did not match the **Trip Date** and **Trip Time** | |
|  |  | **Step 5:** The system displays the list of vehicles that have been returned | |
|  |  |  | |
| **ALTERNATE COURSES:** | Alt step 4: There are no vehicle available on the given date and time and a message is displayed that there are vehicles available and the use case ends | | |
|  |  | | |
| **CONCLUSION:** | The system displays the vehicles that are available | | |
| **POST-CONDITION:** | The schedule should be up to date at all times | | |
| **BUSINESS RULES** | * None | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * None | | |
| **ASSUMPTIONS:** | * None | | |
| **OPEN ISSUES:** | None | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Nondumiso Mahlangu Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Assign vehicle to trip | | **USE CASE TYPE** |
| **USE CASE ID:** | 5.5 | | Business Requirements:🞎 **System Analysis:** **🞎** System Design: 🞎 |
| **PRIORITY:** | High | |
| **SOURCE:** | Ndila Transfers | |
| **PRIMARY BUSINESS ACTOR** | Booking consultant | | |
| **PRIMARY SYSTEM ACTOR** | None | | |
| **OTHER PARTICIPATING ACTORS:** | * None | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * Team leader * Driver supervisor | | |
| **DESCRIPTION:** | This use case describes the event when the booking consultant wants to assign a specific driver to a vehicle. They check the availability of a driver and use case 4.2 Check Driver Availability is invoked. The booking consultant then chooses an available vehicle that is then added on the slot. | | |
| **PRE-CONDITION:** | The booking consultant has to be logged onto the system | | |
| **TRIGGER:** | The booking consultant wants to assign a specific driver to a trip | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**: The booking consultant wants to assign a specific vehicle to a trip and selects assign vehicle option | **Step 2:** Invoke use case 4.2 | |
|  | **Step 3**: The booking consultant selects **Vehicle\_Model** to be booked | **Step 4:** The system writes the **Vehicle\_ID and Vehicle\_Model** to the **Slot**. | |
|  |  | **Step 5:** The system displays a message confirming the assignment. | |
|  |  |  | |
| **ALTERNATE COURSES:** | None | | |
|  |  | | |
| **CONCLUSION:** | The slot is updated | | |
| **POST-CONDITION:** | If a driver is unavailable, they cannot be assigned to another trip | | |
| **BUSINESS RULES** | * There has to be a pre-existing booking before a driver can be assigned | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * None | | |
| **ASSUMPTIONS:** | * None | | |
| **OPEN ISSUES:** | None | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Nondumiso Mahlangu Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Outsource Vehicle | | **USE CASE TYPE** |
| **USE CASE ID:** | 5.6 | | Business Requirements:🞎 **System Analysis:** **🞎** System Design: 🞎 |
| **PRIORITY:** | High | |
| **SOURCE:** | Ndila Transfers | |
| **PRIMARY BUSINESS ACTOR** | Booking consultant | | |
| **PRIMARY SYSTEM ACTOR** | None | | |
| **OTHER PARTICIPATING ACTORS:** | * None | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * Team leader * Driver supervisor | | |
| **DESCRIPTION:** | This use case describes the event when the booking consultant wants to assign a specific driver to a vehicle. The booking consultant prompts the system to display the vehicle then chooses an available vehicle that is then added on the slot. | | |
| **PRE-CONDITION:** | The booking consultant has to be logged onto the system | | |
| **TRIGGER:** | The booking consultant wants to assign a specific driver to a trip | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**: The booking consultant wants to assign a specific vehicle to a trip and selects assign vehicle option | **Step 2:** The system displays information of:  **OutSoucreVehicle\_ID**  **OutSource\_Vehicle\_Model**  **OutSource\_Make** from the **OutSource\_Vehicle** table | |
|  | **Step 3**: The booking consultant selects the **v**ehicle model desired to be booked | **Step 4:** The system writes the **Vehicle\_ID and Vehicle\_Model** to the **Slot**. | |
|  |  | **Step 5:** The system displays a message confirming the assignment | |
|  |  | . | |
| **ALTERNATE COURSES:** | None | | |
|  |  | | |
| **CONCLUSION:** | The slot is updated | | |
| **POST-CONDITION:** | If a vehicle is unavailable, they cannot be assigned to another trip | | |
| **BUSINESS RULES** | * There has to be a pre-existing booking before a vehicle can be assigned | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * None | | |
| **ASSUMPTIONS:** | * None | | |
| **OPEN ISSUES:** | None | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Nondumiso Mahlangu Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Create vehicle group | | **USE CASE TYPE** |
| **USE CASE ID:** | 6.1 | | Business Requirements:◻ System Analysis: ◻ System Design: ◻ |
| **PRIORITY:** | High | |
| **SOURCE:** | Ndila Transfers | |
| **PRIMARY BUSINESS ACTOR** | Booking consultant | | |
| **PRIMARY SYSTEM ACTOR** | Booking consultant | | |
| **OTHER PARTICIPATING ACTORS:** | * None | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * None | | |
| **DESCRIPTION:** | This use case describes an instance when the booking consultant wants to add another vehicle group onto the system database. They retrieve all the relevant vehicle group information and a new vehicle is saved onto the system database. | | |
| **PRE-CONDITION:** | The vehicle group cannot be an existing vehicle group in the system | | |
| **TRIGGER:** | The booking consultant wants to add another vehicle group | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**: The booking consultant wants to add another vehicle group and selects the Add Vehicle Group option. | **Step 2**: The system displays a page where the booking consultant can add the required details of the vehicle group such as:  **VehicleGroup\_Name**  **VehicleGroup\_Description** | |
|  | **Step 3**: The booking consultant adds the necessary details of the new vehicle group | **Step 4**: The system verifies the information gathered | |
|  |  | **Step 5**: The system retrieves the last line in **Vehicle group ID**, increments by 1 and assigns a new **Vehicle group ID** | |
|  |  | **Step 6:** The system writes the information given into the **Vehicle group** table under the new **Vehicle group ID** | |
|  |  | **Step 7**: The system sends a message that a new vehicle group is added | |
|  |  |  | |
| **ALTERNATE COURSES:** | Alt step 4: The booking consultant made an error with the entered information and an error message is provided | | |
|  |  | | |
| **CONCLUSION:** | A new vehicle group is added onto the system database | | |
| **POST-CONDITION:** | The vehicle group information has to be complete | | |
| **BUSINESS RULES** | * None | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * None | | |
| **ASSUMPTIONS:** | * None | | |
| **OPEN ISSUES:** | None | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Nondumiso Mahlangu Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Search vehicle group | | **USE CASE TYPE** |
| **USE CASE ID:** | 6.2 | | Business Requirements:◻ System Analysis: ◻ System Design: ◻ |
| **PRIORITY:** | High | |
| **SOURCE:** | Ndila Transfers | |
| **PRIMARY BUSINESS ACTOR** | Booking consultant | | |
| **PRIMARY SYSTEM ACTOR** | None | | |
| **OTHER PARTICIPATING ACTORS:** | * None | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * None | | |
| **DESCRIPTION:** | This use case describes the booking consultant wanting to search a vehicle group uses the Vehicle group ID to search and retrieve the details of the vehicle from the Vehicle group table in the database. | | |
| **PRE-CONDITION:** | The vehicle group has to have a unique ID | | |
| **TRIGGER:** | The booking consultant enters the Vehicle group ID of the driver onto the system | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**: The booking consultant selects the Search Vehicle Group option of the driver on the system | **Step 2**: The system displays a page for the booking system to enter the search criteria. | |
|  | **Step 3**: The booking consultant enters the characters of the vehicle group they want to search | **Step 4:** The system verifies the character input | |
|  |  | **Step 5:** The system reads the **VehicleGroup\_Name** from the table  **Vehicle Group** | |
|  |  | **Step 6**: The system compares the characters received with the characters in the Vehicle table in the **Vehicle\_Model** field until a match is found | |
|  |  | **Step 7**: The system retrieves the  **VehicleGroup\_ID**  **VehicleGroup\_Name,**  **VehicleGroup\_Description,**  information related to the characters given from the **Vehicle Group** Table | |
|  |  | **Step 8:** The system displays the retrieved information. | |
| **ALTERNATE COURSES:** | Alt step 3: The characters given did not match system requirements and an error message is displayed | | |
|  | Alt step 6: A match is not found and an alert message is displayed | | |
| **CONCLUSION:** | The system retrieves the information related to the Vehicle group ID given | | |
| **POST-CONDITION:** | The vehicle group information searched has to be displayed for the booking consultant | | |
| **BUSINESS RULES** | * None | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * None | | |
| **ASSUMPTIONS:** | * None | | |
| **OPEN ISSUES:** | None | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Matabane Mathopatona Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Register client | | **USE CASE TYPE** |
| **USE CASE ID:** | 7.1 | | Business Requirements: |
| **PRIORITY:** | HIGH | | **System Analysis:** **🞎** |
| **SOURCE:** | Ndila Transfers | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Client | | |
| **PRIMARY SYSTEM ACTOR** | Booking Consultants | | |
| **OTHER PARTICIPATING ACTORS:** | * Operational Manager | | |
| **OTHER INTERESTED STAKEHOLDERS:** |  | | |
| **DESCRIPTION:** | The use case describes an event where a new client is registered on the system. The use case begins when the client wants to start using Ndila services or has interest in using the Ndila services, client details such as name of agency in case of agency, physical address, email, contact number, Client reference can either be client id number (Private clients) or Company registration number(Agencies) will be required to be captured and stored under the **Client Table**. The use case concludes when the client is added to the **Client** table | | |
| **PRE-CONDITION:** | The client must have not used Ndila services before | | |
| **TRIGGER:** | Request to be added to the system through phone call | | |
| **TYPICAL COURSE OF EVENTS:** | **Step 1**: Client will request to use the Ndila services or associate themselves with Ndila | **Step 2:** The booking consultant selects to add client | |
|  |  | **Step 3:** The system prompts the booking consultant to enter the following client details:  **Name of client/agency**  **Physical address of the agency**  **Email address**  **Phone number**  **Client type**  **Client\_Reference** | |
|  | **Step 4:** The client provides the requested details | **Step 5:** The booking consultant captures the details of the new client | |
|  |  | **Step 6:** The booking consultant confirms with the client if the information is correct | |
|  | **Step 7:** The client confirms the information is correct | **Step 8**: The Booking consultant submits, and the system validates the information provided for in case of incorrect format | |
|  |  | **Step 9:** The system validates if the client exists in the **CLIENT** table | |
|  |  | **Step 10:** The system stores the details of the client under **Client** table.  **Name ->Client\_Name**  **PhysAddress-> Client\_Address**  **EmailAddress->Client\_Email**  **PhoneNo-> Client\_Tel**  **ClientRef -> Client\_Reference**    Select **Client type ID** retrieved from  **Client\_Type** table | |
|  |  | **Step 11:** The system confirms that the client is successfully added to the **Client** table and it is communicated to the client by the Booking Consultant | |
| **ALTERNATE COURSES:** | **Alt Step 7:**  The client declines  **Step 7a:**  The booking consultant requests for the details to be rectified***. Return to step 5*** | | |
|  | **Step 8:** Returns an error message explaining wrong information is provided and the first place where the error is identified it will be highlighted requesting correct details. ***Return to step 3*** | | |
|  | **ALT Step 9:** The system retrieves the details of the existing client and the booking consultants communicate with the client that they have already been added to the system.  **Step 9a**:The booking consultant requests the client to confirm the details  **Step 9b:** The client confirms the details.  **Step 9c:**  The booking consultant ends the task | | |
|  | **Step 11:** The system fails to add the client to the table and returns an error message. The use case end | | |
| **CONCLUSION:** | The client is added to the system | | |
| **POST-CONDITION:** | The client should use our services | | |
| **BUSINESS RULES** |  | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** |  | | |
| **ASSUMPTIONS:** |  | | |
| **OPEN ISSUES:** |  | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Matabane Mathopatona Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | SEARCH CLIENT | | **USE CASE TYPE** |
| **USE CASE ID:** | 7.2 | | Business Requirements:🞎 |
| **PRIORITY:** | HIGH | | **System Analysis:🞎** |
| **SOURCE:** | Ndila Transfers | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Booking Consultant | | |
| **PRIMARY SYSTEM ACTOR** |  | | |
| **OTHER PARTICIPATING ACTORS:** | * Manager | | |
| **OTHER INTERESTED STAKEHOLDERS:** |  | | |
| **DESCRIPTION:** | The use case describes an event where a user searches for the client in the system. The use case begins by requesting the name of the agency or client to search, Information about the client if it exists in the **Client** table, else an error message is displayed informing the booking consultant that the user is not registered on the system. The use case concludes when the details are retrieved from the **Client** table provided it exists | | |
| **PRE-CONDITION:** |  | | |
| **TRIGGER:** | Retrieve a client’s details | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1:** Selects the option to search for a client | **Step 2:** The system displays the following information about all the clients in the **Client** table:  **Client\_Name**  **Client\_Reference**  **Client\_PhoneNumber**  **Client\_Email** | |
|  |  | **Step 3**: The system prompts the booking consultant to enter:  **Client\_Name**  or  **Client\_Reference** | |
|  | **Step 4**: The booking consultant enters the client name/agency or Client reference and selects search | **Step 5:** The system validates for correct input formats | |
|  |  | **Step 6:** System narrows the list of clients based on the client/agency name | |
|  | **Step 7:** The booking consultant selects the specific client | **Step 8:** The system displays the following details of the client:  **Client\_Name**  **Client\_Reference**  **Client\_PhoneNumber**  **Client\_Email**  **Client\_Address**  **Client\_type**  Retrieved from the **Client** table. | |
| **ALTERNATE COURSES:** | **ALT Step 5:**  An error message is display by the system informing the booking consultant with input error. The booking consultant can either go back to **Step 3** or terminate the search | | |
|  | **ALT Step 6:**  An error message informing the booking consultant that the specific client does not exist in the **Client** table. The booking consultant can either go back to **Step 3** or terminate the search. | | |
|  |  | | |
|  |  | | |
| **CONCLUSION:** | Client details are retrieved from the **client s**table | | |
| **POST-CONDITION:** |  | | |
| **BUSINESS RULES** |  | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** |  | | |
| **ASSUMPTIONS:** |  | | |
| **OPEN ISSUES:** |  | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Matabane Mathopatona Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Update Client | | **USE CASE TYPE** |
| **USE CASE ID:** | 7.3 | | Business Requirements:X |
| **PRIORITY:** | High | | **System Analysis:** **🞎** |
| **SOURCE:** | Ndila Transfer | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Client | | |
| **PRIMARY SYSTEM ACTOR** | Booking Consultant | | |
| **OTHER PARTICIPATING ACTORS:** |  | | |
| **OTHER INTERESTED STAKEHOLDERS:** |  | | |
| **DESCRIPTION:** | This use case describes an event where a user details are to be updated by the booking consultant. The use case begins by the client requesting to update their details such as location, contact details and so forth. The client provides their name and the client is searched on the system. Client provides the specific details that needs to be updated. The booking consultant will then capture the details and update them on the system | | |
| **PRE-CONDITION:** | The client must exist in the system | | |
| **TRIGGER:** | The client calls in to update their details | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**: The client calls in to request to update their details | **Step 2:** The booking consultant will select client Section | |
|  |  | **Step 3:** The system invokesUse case 7.3 search client | |
|  |  | **Step 4:** The booking consultant selects update client. | |
|  |  | **Step 6:** The system allows the following fields to be modified:  **Client\_Name**  **Client\_Reference**  **Client\_PhoneNumber**  **Client\_Email**  **Client\_Address**  **Client\_Fax**  **Client\_type** | |
|  |  | **Step 7:** The booking consultant requests for the details of the specific fields that the client would like to update | |
|  | **Step 8:** The client provides the specific details that needs to be updated | **Step 9:** The booking consultant captures the information of the specific details that needs to be updated and selects update | |
|  |  | **Step 10:** The system validates the information in the input fields | |
|  |  | **Step 11:** The system saves the updated details in the **Client** table | |
|  |  | **Step 12:** The system informs the booking consultant of successfully updating the information in the **Client** table | |
| **ALTERNATE COURSES:** | **Alt Step 10:** The information provided is invalid the booking consultants’ requests for correct information and return to step 7 | | |
|  | **ALT Step 12:** The system notifies the booking consultant that it failed to update the client details returns to step 7 or terminates | | |
| **CONCLUSION:** | The client’s information is updated in the **Client** table. | | |
| **POST-CONDITION:** |  | | |
| **BUSINESS RULES** |  | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** |  | | |
| **ASSUMPTIONS:** |  | | |
| **OPEN ISSUES:** |  | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Matabane Mathopatona Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Add Client Type | | **USE CASE TYPE** |
| **USE CASE ID:** | 8.1 | | Business Requirements:X |
| **PRIORITY:** | High | | **System Analysis:** **🞎** |
| **SOURCE:** | Ndila Transfers | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Operations Manager | | |
| **PRIMARY SYSTEM ACTOR** |  | | |
| **OTHER PARTICIPATING ACTORS:** |  | | |
| **OTHER INTERESTED STAKEHOLDERS:** |  | | |
| **DESCRIPTION:** | The use case describes an event where client type is added to the system the client type can either be private or public, public clients are agencies, governments and companies. Private clients are individual people requesting for a personal transportation. The name, description of the client type will be added to the system. The system concludes when the type is added to the database. | | |
| **PRE-CONDITION:** | Operation Manager should be logged in | | |
| **TRIGGER:** | Request to add new Client Type | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**: The operations manager logs on the system and request to add new client type | **Step 2:**  The system prompts the Operational Manager to capture following client type details: **Client\_Type\_Name**  **Client\_Type\_Description** | |
|  | **Step 3:** The operations manager enters the required details of the client type, selects submit | **Step 4:**  The system checks if the information provided is in valid format | |
|  |  | **Step 5:**  The system validates if the client type exists in the **Client\_Type** table | |
|  |  | **Step 6**: The system stores the required details in the **Client\_Type** table | |
|  |  | **Step 7:** The system confirms the successful adding of the client type to the **Client\_Type** table | |
|  |  |  | |
| **ALTERNATE COURSES:** | **ALT Step 4:** The system displays an error message informing the operations manager that the information provided is invalid and request the operations to either enter information or terminate. | | |
|  | **ALT Step 5:** The system displays an error message Informing the operations manager that the client type exists. | | |
|  | **ALT Step 7:**  The system informs the operational manager that it failed to store the new client\_type ,the manager can return to step 3 or terminate | | |
|  |  | | |
| **CONCLUSION:** | The client type is added to the database | | |
| **POST-CONDITION:** |  | | |
| **BUSINESS RULES** |  | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** |  | | |
| **ASSUMPTIONS:** |  | | |
| **OPEN ISSUES:** |  | | |
|  |  | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Matabane Mathopatona Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Search Client Type | | **USE CASE TYPE** |
| **USE CASE ID:** | 8.2 | | Business Requirements: 🞎 |
| **PRIORITY:** | High | | **System Analysis:** **X** |
| **SOURCE:** | Ndila Transfers | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Booking Consultants | | |
| **PRIMARY SYSTEM ACTOR** |  | | |
| **OTHER PARTICIPATING ACTORS:** |  | | |
| **OTHER INTERESTED STAKEHOLDERS:** |  | | |
| **DESCRIPTION:** | This use case describes an event of searching a client type in the database. The name of the client type is typed and it is search if it exists. The details of the client type are retrieved if it exists | | |
| **PRE-CONDITION:** |  | | |
| **TRIGGER:** | Checking if the client type exists | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**: The Booking consultant wants to search and selects the Client section on the system | **Step 2:** The system displays operations that the user can perform such as update, search, add new client type | |
|  | **Step 3:** The Booking consultant selects the search operation | **Step 4:** The system prompts the name of the client type to be searched  And displays all the client types in the **Client\_Type table**  **Client\_Type\_Name**  **Client\_Type\_Description** | |
|  | **Step 5:** The user enters the name of the client type in the search field | **Step 6:** The system validates the name provided in the input box if it’s in correct format such as no numbers or the search field is not empty | |
|  |  | **Step 7:** The system displays the following details retrieved from **Client\_Type** table of a client types matching the name entered in the search field:  **Client\_Type\_Name**  **Client\_Type\_Description** | |
|  | **Step 8:**  The Booking consultant selects the specific client type |  | |
|  |  | **Step 9**: The details of the client type retrieved from the **Client\_Type** table are displayed on the screen:  **Client\_Type\_Name-> Client\_Type\_Name**  **Client\_Type\_Description-> Client\_Type\_Description** | |
|  |  |  | |
|  |  |  | |
| **ALTERNATE COURSES:** | **ALT Step 6**: The system displays an error message informing the user incorrect input and request for correct input or the user can terminate the search | | |
|  | **ALT Step 9:** The specific client type does not exist in the **ClientType** table | | |
|  |  | | |
|  |  | | |
| **CONCLUSION:** | The details of the client type are retrieved from the **ClientType** table. | | |
| **POST-CONDITION:** |  | | |
| **BUSINESS RULES** |  | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** |  | | |
| **ASSUMPTIONS:** |  | | |
| **OPEN ISSUES:** |  | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Matabane Mathopatona Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Update Client Type | | **USE CASE TYPE** |
| **USE CASE ID:** | 8.3 | | Business Requirements:X |
| **PRIORITY:** | High | | **System Analysis:** **🞎** |
| **SOURCE:** | Ndila Transfer | | System Design:🞎 |
| **PRIMARY BUSINESS ACTOR** | Operations Manager | | |
| **PRIMARY SYSTEM ACTOR** |  | | |
| **OTHER PARTICIPATING ACTORS:** |  | | |
| **OTHER INTERESTED STAKEHOLDERS:** |  | | |
| **DESCRIPTION:** | The use case describes an event where a client type is to be updated on the system. The use case begins by searching the specific client type and retrieving all the updatable field such as description or the name of the client type. The use case concludes when the Client type is updated in the **ClientType** table. | | |
| **PRE-CONDITION:** | Operations manager needs to be logged in | | |
| **TRIGGER:** | To update the specific client type | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**: The Operations manager requests to update a specific client type by selecting the Client section on the screen | **Step 2:** The system displays the available operations to be performed under client type | |
|  | **Step 3:** The operations manager selects the update client type | **Step 4:** The system requests the user to enter the name of the specific type, the user wants to update. | |
|  | **Step 5:** The operations manager enters the name of the specific client type | **Step 6:** The system invokes use case 8.2(Search Client type). | |
|  | **Step 7:** The operations manager selects the update options | **Step 7:** The system allows the following fields to be updated :  **Client\_Type\_Description**  **Client\_Type\_Name** | |
|  | **Step 8:** The operations manager enters new information on the fields they want to update and clicks the button update | **Step 9:**  The system validates the information in the updated fields | |
|  |  | **Step 10:**  The system saves the updated information in the **ClientType** table. And alerts the operations manager that it is successfully added. | |
| **ALTERNATE COURSES:** | **ALT Step 9:** The system informs the operations manager that the information provided is not in a correct format. The operations manager can either terminate or Go to **Step 5.** | | |
|  | **ALT Step 10:** The system informs the operations manager that it failed to update the client type details and can either terminate or return to step 5 | | |
| **CONCLUSION:** | The client type is updated in the **ClientType**  table | | |
| **POST-CONDITION:** |  | | |
| **BUSINESS RULES** |  | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** |  | | |
| **ASSUMPTIONS:** |  | | |
| **OPEN ISSUES:** |  | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Matabane Mathopatona Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | CREATE BOOKING | | **USE CASE TYPE** |
| **USE CASE ID:** | 9.1 | | Business Requirements: X |
| **PRIORITY:** | High | | **System Analysis:** **🞎** |
| **SOURCE:** | Ndila Transfers | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Client (Agency or Private Client) | | |
| **PRIMARY SYSTEM ACTOR** | Booking Consultant | | |
| **OTHER PARTICIPATING ACTORS:** | * Driver * Driver Supervisor | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * Operations Manager | | |
| **DESCRIPTION:** | This use case describes an event where a client request to make a booking for a trip. The use case begins when the client calls in or sends an email to request for services. The name of the client/agency will be required to check if they have used the services before else they are added to the system. The booking information such as date, time, location (Departure and destination), Vehicle group/number of people to be transported and Passenger details. A quote is generated and communicated to the client. The use case concludes when the Booking is saved in the booking table. | | |
| **PRE-CONDITION:** | Must be registered client | | |
| **TRIGGER:** | Calls or sends an email to request to book a trip | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**: Client calls/emails in to request a booking | **Step 2**: The Booking consultant selects to create a Booking | |
|  |  | **Step 3**: The system prompts the booking consultant to enter the following details  **ClientName** | |
|  | **Step 4**: The client provides their agency name | **Step 5**: The booking consultant captures the name of the agency in the input field and select search | |
|  |  | **Step 6**: The system validates the information entered in the input field | |
|  |  | **Step 7**: invoke use case search client and retrieves the client details | |
|  |  | **Step 8**: The booking consultant confirms the details of the client retrieved from the Client table | |
|  | **Step 9:** The client confirms the details | **Step 10:** The System saves the client information retrieved from the Client table in the input fields | |
|  |  | **Step 11:** The system prompts the booking consultant to enter the following details:  ***DateOfPickUp***  ***DateOfArrival***  ***TimeOfPickUp***  ***TimeOfArrival***  ***PickupLocation***  ***DropOffLocation***  ***NumberOfPassengers*** | |
|  | **Step 12:** The client provides the requested details | **Step 13:** The booking consultant captures the details provided by the client | |
|  |  | **Step 14:** The system validates the input if it’s in correct format | |
|  |  | **Step 15:** The system validates if the booking exists in the **Booking** table | |
|  |  | **Step 16:** invokes use case *Generate quote and retrieves the generated quote* | |
|  |  | **Step 17:** The system prompts confirmation of the estimated price of the trip | |
|  | **Step 18**: The client agrees to the trip price | **Step 19**: The system prompts the Booking consultant to enter the following details of **Passenger**  **Passenger Name**  **Passenger Surname**  **Passenger PhoneNo**  **Booking Order** | |
|  | **Step 20:** The client provides the details of the person to be collected | **Step 21:** The booking consultant captures the details provided to their respective fields | |
|  |  | **Step 22:** The system validates the information in the input fields | |
|  |  | **Step 23:** The system validates if the passenger exists in the **Passenger** table | |
|  |  | **Step 24:** System saves the Passenger details in  **Passenger Table**  **Name -> Passenger\_Name**  **Surname -> Passenger\_Surname**  **PhoneNo-> Passenger\_PhoneNo**  **BookingOrder-> Booking\_Order** | |
|  |  | **Step 25:** The system prompts the booking consultant to enter pickup Instructions | |
|  | **Step 26**: The client provides instructions about the collection | **Step 27:** The booking consultant captures the instructions | |
|  |  | **Step 28:** The system prompts for Booking confirmation | |
|  | **Step 29:** The client confirms the booking details | **Step 30:** The system  Auto Generate Booking\_Reference,  Saves the Booking details in **Booking\_Trip table**  ***DateOfPickup->*Date\_of\_PickUp**  ***DateOfArrival->*Date\_of\_Arrival**  ***PickUpTime->*Time\_Of\_PickUp**  ***ArrivalTime->*Time\_Of\_Arrival**  ***“Awaiting Voucher” ->* Booking\_Status**  Retrieve ***Pickup\_Location\_ID*** from ***Location*** table  ***PLocationID->*Pickup\_Location\_ID**  ***DLocationID->*DropOff\_Location\_ID**  ***PassNumber->*Number\_Of\_Passengers**  saves the following quote details in the **Invoice** table  **InvoiceNo->Invoice\_Number**  **InvoiceDate->Invoice\_Date**  **“Outstanding”-> Payment\_Status**  **BookingRef->Booking\_Reference**  **Booking\_Cost->Booking\_Cost** | |
|  |  | **Step 31:** The System displays the booking reference. | |
|  |  | **Step 32:** The booking consultant notifies the client that the trip has been captured and it will be approved upon receiving a voucher card | |
| **ALTERNATE COURSES:** |  | | |
|  | **ALT step 6:** Incorrect input returns to step 4 | | |
|  | **ALT Step 7:** Error message informing that the client doesn’t exist and invokes use 7.1 Add Client | | |
|  | | |
| **ALT Step 9:** Declines and request to update their details by invoking use case 7.3 Update client | | |
| **ALT Step 14:** The input is incorrect and request valid input returns to step 12 | | |
|  | **ALT Step 15:** The booking already exists, and it is communicated to the client terminates. | | |
|  | **ALT Step 18:** The client declines and the price is negotiated, The price gets overwritten | | |
|  | **ALT Step 22:** The input is invalid returns to step 31 | | |
| **ALT Step 23:** The Passenger already exists in the Passenger table  Retrieve the Passenger\_ID from **Passenger table**  , return step 25 | | |
| **ALT Step 24:** The system failed to store Passenger , terminates | | |
| **ALT Step 29:** The client declines the booking details and provides the specifics that needs update | | |
| **ALT Step 31:** The system failed to save the booking and terminates | | |
|  | | |
| **CONCLUSION:** | The trip is successfully stored in the booking table | | |
| **POST-CONDITION:** | A voucher card must be submitted for the trip to be executed | | |
| **BUSINESS RULES** | * Booking reference will be assigned to every booking | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * Other agencies submit their voucher card late, System must be overwritten to allow such cases | | |
| **ASSUMPTIONS:** |  | | |
| **OPEN ISSUES:** |  | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Matabane Mathopatona Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Search Booking | | **USE CASE TYPE** |
| **USE CASE ID:** | 9.2 | | Business Requirements: X |
| **PRIORITY:** | High | | **System Analysis:** **🞎** |
| **SOURCE:** | Ndila Trasnfers | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Client | | |
| **PRIMARY SYSTEM ACTOR** | Booking Consultant | | |
| **OTHER PARTICIPATING ACTORS:** | * Manager | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * Drivers | | |
| **DESCRIPTION:** | This use case describes an event where a booking is searched by a booking consultant in a case where they want to update it or retrieve its details. The search can be performed by either using name of client or booking reference. Booking details are retrieved from the following tables; **Booking** table, **Invoice**, **Vehicle\_Group, Driver, Passenger** and are displayed on the screen when the booking consultant types in characters like the booking reference or client name.The use case concludes when the results are found | | |
| **PRE-CONDITION:** | The booking reference should be valid | | |
| **TRIGGER:** | Search for the booking | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**: The booking consultant selects search Booking. | **Step 2:** The system displays a list of bookings containing the following details retrieved from the **Booking** table:  **Booking\_Reference**  **Client\_Name**  **Date\_of\_PickUp**  **Date\_of\_Arrival**  **Time\_Of\_PickUp**  **Time\_Of\_Arrival**  **Pickup\_Location**  **DropOff\_Location** | |
|  |  | **Step 3:** The system prompts the booking consultant to enter the **Booking\_Reference/agency name** | |
|  | **Step 4:** The Booking consultant enters the booking reference /agency name | **Step 5:** The system validates the input for valid format | |
|  |  | **Step 6:** Bookings that have similar characters as the one typed in the search field are retrieved from the **Booking** table displayed | |
|  | **Step 7:** The booking consultant selects the specific client | **Step 8**: Information about the booking such as  **Booking\_Reference**  **Client\_Name**  **Date\_of\_PickUp**  **Date\_of\_Arrival**  **Time\_Of\_PickUp**  **Time\_Of\_Arrival**  **Pickup\_Location**  **DropOff\_Location**  **Booking\_Status**  **Vehicle\_Assigned** retrieved from the **Vehicle\_Group table**  **Driver\_Assigned** retrieved from the **Driver table**  **Number\_Of\_Passengers**  **Passenger\_Name**  **Passenger\_Surname**  **Passenger\_PhoneNo**  **Booking\_Order**  Retrieved from **Booking** table is displayed on the screen. | |
| **ALTERNATE COURSES:** | **ALT Step 5:** The system prompts Invalid input return to step 3 | | |
|  | **ALT Step 6:** An empty list is displayed | | |
|  | **ALT step 7:** The Booking doesn’t exist can either return to step 3 or terminate | | |
| **CONCLUSION:** | Details of the specific booking are retrieved | | |
| **POST-CONDITION:** |  | | |
| **BUSINESS RULES** |  | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** |  | | |
| **ASSUMPTIONS:** |  | | |
| **OPEN ISSUES:** |  | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Matabane Mathopatona Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Update Booking | | **USE CASE TYPE** |
| **USE CASE ID:** | 9.3 | | Business Requirements: X |
| **PRIORITY:** | High | | **System Analysis:** **🞎** |
| **SOURCE:** | Ndila Transfers | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Client | | |
| **PRIMARY SYSTEM ACTOR** | Booking Consultant | | |
| **OTHER PARTICIPATING ACTORS:** |  | | |
| **OTHER INTERESTED STAKEHOLDERS:** |  | | |
| **DESCRIPTION:** | This use case describes an event where a user wants to update a booking made by a client. The booking consultant will request the booking reference or the name of the client/agency who made the booking. The booking will invoke the search booking and details of the booking are retrieved enabling the booking consultant to modify the details. The use case concludes when the booking is updated | | |
| **PRE-CONDITION:** | A valid booking reference should be provided | | |
| **TRIGGER:** | Request to update booking details | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**: The client will request to update the details of the Booking | **Step 2:** The booking consultant selects the Booking Section | |
|  |  | **Step 3:** The system prompts the booking consultant to enter the **Booking\_Reference** | |
|  | **Step 4:** The client provides the booking reference | **Step 5**: The booking consultant enters the booking reference and invoke use case search booking | |
|  |  | **Step 6:** The booking consultant selects update booking | |
|  |  | **Step 7:** The system enables the booking consultant to update the following fields  **Date\_of\_PickUp**  **Date\_of\_Arrival**  **Time\_Of\_PickUp**  **Time\_Of\_Arrival**  **Pickup\_Location**  **DropOff\_Location**  **Number\_Of\_Passengers**  **Passenger\_Name**  **Passenger\_Surname**  **Passenger\_PhoneNo**  **Booking\_Order**  **Booking\_status**  **Client\_Name** | |
|  |  | **Step 8:**  The booking consultants asks the client which details they will like to update. | |
|  | **Step 9**: The client provides the details that they want to update. | **Step 10**: The booking consultant enters the new information and submit | |
|  |  | **Step 11**: The system verifies if the information provided is in a correct format | |
|  |  | **Step 12**: The system stores the updated information under Booking table | |
|  |  | **Step 13**: The System confirms that the update is complete | |
|  |  | **Step 14**: The booking consultant confirms with the client that the update is complete. | |
| **ALTERNATE COURSES:** | **ALT Step 11:** The system displays an error message informing the booking consultant that incorrect input return to step 10 | | |
|  | **ALT Step 13:** The system informs the booking consultant that the system failed to update the booking details | | |
| **CONCLUSION:** | This use case concludes when the booking is updated. | | |
| **POST-CONDITION:** |  | | |
| **BUSINESS RULES** |  | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** |  | | |
| **ASSUMPTIONS:** |  | | |
| **OPEN ISSUES:** |  | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Matabane Mathopatona Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Cancel Booking | | **USE CASE TYPE** |
| **USE CASE ID:** | 9.4 | | Business Requirements:X |
| **PRIORITY:** | High | | **System Analysis:** **🞎** |
| **SOURCE:** | Ndila Transfers | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Client | | |
| **PRIMARY SYSTEM ACTOR** | Booking consultant | | |
| **OTHER PARTICIPATING ACTORS:** |  | | |
| **OTHER INTERESTED STAKEHOLDERS:** |  | | |
| **DESCRIPTION:** | This use case describes an event where a client will like to cancel a booking. The booking consultant will request the name of the client and a reference provided to the client for the booking. The booking consultant will check if the booking really exists in the **Booking** table and check if the driver has been dispatched already. If driver has been dispatched the client will be requested to pay the amount charged, Else the booking is cancelled, and the driver is informed about the cancelation of the trip. The system concludes when the booking has been cancelled | | |
| **PRE-CONDITION:** | A valid booking reference should be provided | | |
| **TRIGGER:** | The client calls in and requests to cancel a trip | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**: The client calls in to request a cancelation of a booking | **Step 2:** The booking consultant selects the Booking section and select Cancel Booking | |
|  |  | **Step 3:** The system prompts the booking consultant to enter the Booking reference | |
|  | **Step 4:** The client provides the booking reference | **Step 5:** The booking consultant enters the booking reference and invokes use case Search Booking | |
|  |  | **Step 6:** The booking consultant checks if the trip is due by comparing departure time with the current time the client is requesting for a cancelation | |
|  |  | **Step 7:** The booking consultant selects proceed | |
|  |  | **Step 8:** The system prompts the booking consultant to enter reasons for cancelation | |
|  | **Step 9:** The client provides the specific reasons of trip cancellation | **Step 10:** The booking consultant captures the cancelation reason and submits | |
|  |  | **Step 11:**  The system updates the availability of a vehicle for that specific date and time by removing it in **slot table** | |
|  |  | **Step 12:** The system updates the availability of driver and the driver is informed about the cancellation of the trip. | |
|  |  | **Step 13:** The system updates the schedule to indicate that the trip is canceled | |
|  |  | **Step 14:** The system updates the **Payment\_Status** under **Invoice** table that the trip is cancelled. | |
|  |  | **Step 15:** The system updates the **Booking\_Status** to cancelled | |
|  |  | **Step 16:** The booking consultant communicates with the client the completion of the booking cancellation | |
| **ALTERNATE COURSES:** |  | | |
|  | **Alt Step 7:** The booking consultant informs the client that the trip is due already and they must pay 50% of the total cost of the trip **then proceed** | | |
|  | **Alt Step 14:** The booking consultant will update the payment status under Invoice table to payment outstanding. | | |
|  | **ALT Step 16:** Failed to cancel the booking then terminates | | |
| **CONCLUSION:** | The use case concludes when the trip is cancelled | | |
| **POST-CONDITION:** |  | | |
| **BUSINESS RULES** | * Client must pay up 50% of the amount if the trip is already due | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** |  | | |
| **ASSUMPTIONS:** |  | | |
| **OPEN ISSUES:** |  | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Matabane Mathopatona Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Confirm Booking | | **USE CASE TYPE** |
| **USE CASE ID:** | 9.5 | | Business Requirements:X |
| **PRIORITY:** | High | | **System Analysis:** **🞎** |
| **SOURCE:** | Ndila | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Booking Consultant | | |
| **PRIMARY SYSTEM ACTOR** |  | | |
| **OTHER PARTICIPATING ACTORS:** | * Agency | | |
| **OTHER INTERESTED STAKEHOLDERS:** | * Operational /financial Manager | | |
| **DESCRIPTION:** | This use case describes an event where a booking is to be confirmed after receiving a voucher card from the agency. The agency sends an email with a voucher card which is used to confirm that the agency is aware of the trip and approves of the costs of the trip. The booking consultants assigns vehicle and driver upon receiving the email from the agency. This use case concludes when the trip is booked, and its status is updated. | | |
| **PRE-CONDITION:** | The booking should exist and a voucher card | | |
| **TRIGGER:** | Voucher card received from the agency | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**: The booking consultant receives an email from the agency containing the voucher card for a trip | **Step 2:** The system requests the booking reference from the booking consultant | |
|  | **Step 3:** The booking consultant inputs the booking reference | **Step 4**: The system invokes use case search Booking and retrieves the details of the booking and allows modification | |
|  | **Step 5:** The booking consultant checks for available vehicle by invoking use case *check vehicle availability*. | **Step 6:** The available vehicles are displayed based on the criteria given | |
|  | **Step 7:** The booking consultant selects the specific vehicle suitable for the specific trip | **Step 8:** The system invokes use case *assign vehicle* | |
|  | **Step 9**: The booking consultant checks for available driver and invoke use case *check driver availability.* | **Step 10:** The available drivers are displayed based on the criteria given. | |
|  | **Step 11:** The booking consultant selects the specific driver | **Step 12:** The system invokes use case *assign driver* | |
|  | **Step 13:** The booking consultant invokes use case *Create schedule* | **Step 14**: The system updates the booking’s status to complete in the **booking** table | |
|  | **Step 15:** The booking consultant calls the agency to confirm that the voucher is received the booking is approved. |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
| **ALTERNATE COURSES:** | **ALT step 4:** The booking is not found and use case *create booking*is invoked. | | |
|  | **ALT Step 7:**  Outsource a vehicle and a driver and skip steps to step 13 | | |
|  | **ALT step 11:** Outsource a driver. | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
| **CONCLUSION:** | This use concludes when the trip’s is confirmed and updated in the schedule | | |
| **POST-CONDITION:** |  | | |
| **BUSINESS RULES** |  | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** |  | | |
| **ASSUMPTIONS:** |  | | |
| **OPEN ISSUES:** |  | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Mpho Mosotho Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Add to Schedule | | **USE CASE TYPE** |
| **USE CASE ID:** | 10.1 | | Business Requirements: 🞎 |
| **PRIORITY:** | High | | **System Analysis:** **√** |
| **SOURCE:** | Requirement List | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Booking Consultant | | |
| **PRIMARY SYSTEM ACTOR** | Booking Consultant | | |
| **OTHER PARTICIPATING ACTORS:** |  | | |
| **OTHER INTERESTED STAKEHOLDERS:** |  | | |
| **DESCRIPTION:** | This use case describes the event where a booking is added to the schedule. The Consultant will add a booking that is being made to the schedule by providing all the necessary details, allowing the system to validate the input information and save it to a database. | | |
| **PRE-CONDITION:** | Consultant has captured all booking details | | |
| **TRIGGER:** | Use Case Confirm Booking | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**: A Consultant wants to add a booking to the Main schedule | **Step 2**: The system retrieves the relevant information from the booking Use Case that has been made:   * **BookingReference** * **Client\_ID** * **Driver\_ID** * **Date** * **Time** * **Destination Time** * **DestinationLocation** * **TripDuration (Which is calculated using Time and Destination Time)** | |
|  |  | **Step 3:** System validates the format for the retrieved details | |
|  |  | **Step 4:** System retrieves the last Schedule\_ID from the Schedule Table and increments it by one. | |
|  |  | **Step 5:** System will store the booking using the retrieved information (**Date, Time, TripDuration**) on the day chosen using the **Date** details alongside the new Schedule\_ID | |
|  |  | **Step 6:** System displays a confirmation message to the manager | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
| **ALTERNATE COURSES:** | **ALT Step 3**: The entered information is invalid. The system displays an error message stating that the formatting is incorrect and terminates the use case and returns to the **Create Booking** Use Case. | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
| **CONCLUSION:** | A new booking has been added to the schedule and has been saved | | |
| **POST-CONDITION:** | The Consultant will be able to view the entered booking on the schedule | | |
| **BUSINESS RULES** |  | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** |  | | |
| **ASSUMPTIONS:** |  | | |
| **OPEN ISSUES:** |  | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Mpho Mosotho Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Update Schedule | | **USE CASE TYPE** |
| **USE CASE ID:** | 10.2 | | Business Requirements:🞎 |
| **PRIORITY:** | Medium | | **System Analysis:√** |
| **SOURCE:** | Requirement List | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Client | | |
| **PRIMARY SYSTEM ACTOR** | Booking Consultant | | |
| **OTHER PARTICIPATING ACTORS:** |  | | |
| **OTHER INTERESTED STAKEHOLDERS:** |  | | |
| **DESCRIPTION:** | |  | | --- | | This use case describes the event where a Manager wants to update the details of the bookings that are already slotted and saved onto the system as well as in the Schedule table in the database, this may be due to outdated or incorrect information that was entered. The use case concludes with the system updating the required details on the system. | | | |
| **PRE-CONDITION:** | Booking has already been stored in the schedule | | |
| **TRIGGER:** | Consultant wants to update the scheduled booking | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**: Consultant wants to update the scheduled booking | **Step 2**: System displays the schedule options | |
|  | **Step 3:** Consultant selects the View Schedule option | **Step 4:** System invokes the View Schedule Use case | |
|  |  | **Step 5:** System displays the booking details which are retrieved from the **Schedule** table.  The following details can be edited:   * Date * Time * TripDuration | |
|  | **Step 6:** Consultant updates the relevant information based upon the given changes | **Step 7:** The system reads the updated details entered and validates them | |
|  |  | **Step 8:** The system displays a validation message indicating the correct format for the details | |
|  |  | **Step 9:** System then saves the updated information in the **schedule** table | |
|  |  | **Step 10:** System then displays the updated schedule. | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
| **ALTERNATE COURSES:** | **ALT Step 8**: The entered information is invalid. The system displays an error message stating that the format is incorrect and returns to **step 6** | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
| **CONCLUSION:** | The schedule details have been entered and stored on the system | | |
| **POST-CONDITION:** | |  | | --- | | The schedule details are now correctly updated and saved | | | |
| **BUSINESS RULES** | * n/a | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * n/a | | |
| **ASSUMPTIONS:** | * n/a | | |
| **OPEN ISSUES:** | n/a | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Mpho Mosotho Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | View Schedule | | **USE CASE TYPE** |
| **USE CASE ID:** | 10.3 | | Business Requirements:🞎 |
| **PRIORITY:** | High | | **System Analysis:** **√** |
| **SOURCE:** | Requirement List | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Booking Consultant | | |
| **PRIMARY SYSTEM ACTOR** | Booking Consultant | | |
| **OTHER PARTICIPATING ACTORS:** |  | | |
| **OTHER INTERESTED STAKEHOLDERS:** |  | | |
| **DESCRIPTION:** | This use case describes the event where the consultant selects an option that makes the system displays a schedule of all the bookings that have been made. Allowing the consultant to view bookings that have been made on a specific day, month and/or year. | | |
| **PRE-CONDITION:** | Consultant is logged onto the system | | |
| **TRIGGER:** | Consultant wants to view bookings that have been made | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**: Consultant wants to view bookings that have been made | **Step 2**: The system displays the schedule allowing the consultant to choose a date or search for a specific date. | |
|  | **Step 3:** Consultant clicks on the chosen date he/she wants to view | **Step 4:** System displays all bookings that have been made on the specific date chosen. The following information will be displayed from the **Schedule** table:   * **BookingRefernce** * **Client\_ID** * **Driver\_ID** * **Date** * **Time** * **ContactPerson\_ID** * **DestinationLocation** * **TripDuration** | |
|  | **Step 5:** The consultant chooses the booking in which he/she wants to view | **Step 6:** System displays the bookings chosen by the consultant. The following information will be displayed from the **Schedule** table:   * **BookingRefernce** * **Client\_ID** * **Driver\_ID** * **Date** * **Time** * **ContactPerson\_ID** * **DestinationLocation** * **TripDuration** | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
| **ALTERNATE COURSES:** | **ALT Step 3**: Consultant enters the date which he/she wants to view. The following details are entered   * Day * Month * Year | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
| **CONCLUSION:** | A booking has been displayed | | |
| **POST-CONDITION:** |  | | |
| **BUSINESS RULES** | * n/a | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * n/a | | |
| **ASSUMPTIONS:** | * n/a | | |
| **OPEN ISSUES:** | n/a | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Mpho Mosotho Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Generate Bookings Report | | **USE CASE TYPE** |
| **USE CASE ID:** | 11.1 | | Business Requirements:🞎 |
| **PRIORITY:** | High | | **System Analysis:** **x** |
| **SOURCE:** | Requirement List | | System Design:🞎 |
| **PRIMARY BUSINESS ACTOR** | Operational Manager | | |
| **PRIMARY SYSTEM ACTOR** | n/a | | |
| **OTHER PARTICIPATING ACTORS:** |  | | |
| **OTHER INTERESTED STAKEHOLDERS:** |  | | |
| **DESCRIPTION:** | This Use case describes an event where a Booking Report is generated based on two criteria. One being the list of all bookings made by a certain client. Second being A list generated to list all bookings made in a certain month of a certain year | | |
| **PRE-CONDITION:** | Operational Manager must be logged on to the system | | |
| **TRIGGER:** | Manager requests for a report | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**:Operational manager Requests to Create a Booking | **Step 2:** System shows all the reports the manager will be able to generate:   * Generate Bookings * Generate Vehicle Report | |
|  | **Step 3:** Manager chooses to choose the booking report generator. | **Step 4:** System shows him the two options of the kind of reports he can choose from. Whether he will search by client or search by date. | |
|  |  | **Step 5:** System requires manger to enter a **Surname** or  **Date** | |
|  | **Step 6:** Operational manager searches client by **Surname** or **Date** | **Step 7:** System uses the information entered by client and reads the **Booking Table** | |
|  |  | **Step 8:** Once client is found system returns list of all bookings made by client searched from the **Booking Table.**  The system will return the following details:  **Client\_ID**  **Client\_Surname**  **Booking\_Date**  **Pickup-Location**  **DropOff-location** | |
|  | **Step 9:** After report is generated Operational Manger can now choose to download the returned results into a pdf document. |  | |
|  |  |  | |
|  |  |  | |
| **ALTERNATE COURSES:** | **ALT Step 8:**  System could not find a booking and displays the error message to the Operational Manager | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
| **CONCLUSION:** | A booking report is generated | | |
| **POST-CONDITION:** |  | | |
| **BUSINESS RULES** |  | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** | * The “**To Date**” must not be a date after the current date. | | |
| **ASSUMPTIONS:** |  | | |
| **OPEN ISSUES:** |  | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Mpho Mosotho Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Generate Vehicle Report | | **USE CASE TYPE** |
| **USE CASE ID:** | 11.2 | | Business Requirements:🞎 |
| **PRIORITY:** | High | | **System Analysis:** **x** |
| **SOURCE:** | Requirement List | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Operational Manager | | |
| **PRIMARY SYSTEM ACTOR** | n/a | | |
| **OTHER PARTICIPATING ACTORS:** |  | | |
| **OTHER INTERESTED STAKEHOLDERS:** |  | | |
| **DESCRIPTION:** | This Use case describes an event where a Vehicle Report is generated based on two criteria. One being the list of all bookings made by a certain client. Second being A list generated to list all bookings made in a certain month of a certain year | | |
| **PRE-CONDITION:** | Operational Manager must be logged on to the system | | |
| **TRIGGER:** | Manager requests for a report | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**: Operational manager Requests to Create a Booking | **Step 2:** System shows all the reports the manager will be able to generate:   * Generate Bookings * Generate Vehicle Report | |
|  | **Step 3:** Manager chooses to choose the Vehicle report generator. | **Step 4:** System shows him the two options of the kind of reports he can choose from. Whether he will search by client or search by date. | |
|  |  | **Step 5:** System requires manger to enter a **Surname** or  **Date** | |
|  | **Step 6:** Operational manager searches client by **Surname** or **Date** | **Step 7:** System uses the information entered by client and reads the **Booking Table** | |
|  |  | **Step 8:** Once client is found system returns list of all bookings made by client searched from the **Booking Table.**  The system will return the following details:  **Client\_ID**  **Client\_Surname**  **Booking\_Date**  **Pickup-Location**  **DropOff-location** | |
|  | **Step 9:** After report is generated Operational Manger can now choose to download the returned results into a pdf document. |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
| **ALTERNATE COURSES:** | **ALT Step 8:**  System could not find a booking and displays the error message to the Operational Manager | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
| **CONCLUSION:** | A Vehicle report is generated | | |
| **POST-CONDITION:** |  | | |
| **BUSINESS RULES** |  | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** |  | | |
| **ASSUMPTIONS:** |  | | |
| **OPEN ISSUES:** |  | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Mpho Mosotho Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Generate Mileage Report | | **USE CASE TYPE** |
| **USE CASE ID:** | 11.3 | | Business Requirements:🞎 |
| **PRIORITY:** | Medium | | **System Analysis √** |
| **SOURCE:** | Requirement List | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Operational Manager | | |
| **PRIMARY SYSTEM ACTOR** | Operational Manager | | |
| **OTHER PARTICIPATING ACTORS:** |  | | |
| **OTHER INTERESTED STAKEHOLDERS:** |  | | |
| **DESCRIPTION:** | This use case describes the event where a manager requests the mileage report from the system. The system generates the report by retrieving information matching certain criteria. The report is then saved to a database. This use case concludes with the report being saved to the system. | | |
| **PRE-CONDITION:** | The manager must be logged in. | | |
| **TRIGGER:** | The manager wants to generate the mileage report | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1:** The manager selects the generate the mileage report. | **Step 2**: The system displays the mileage generation report screen. | |
|  | **Step 3:** The manager enters the requested report criteria based on the following options:   * Vehicle\_ID * DistanceTravelled | **Step 4:** The system reads the entered report criteria information used for the report. | |
|  |  | **Step 4:** System retrieves the information and returns of all vehicles that match the input criteria  From the **Vehicle Maintenance** Table | |
|  |  | **Step 5:** System the generates the Mileage report | |
|  |  | **Step 6:** System stores the report in the documents table | |
|  |  | **Step 7:** System displays a confirmation message to the manager | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
| **ALTERNATE COURSES:** | ALT Step 5: Once the report generation is completed, no information is displayed as no data matched the report criteria. Error notification is sent to the manager stating there were no matches and requests the manager to re-enter the report criteria.   * Return to step 3. | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
| **CONCLUSION:** | The report is generated | | |
| **POST-CONDITION:** | The complete report is generated and stored | | |
| **BUSINESS RULES** |  | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** |  | | |
| **ASSUMPTIONS:** |  | | |
| **OPEN ISSUES:** |  | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Mpho Mosotho Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Schedule Vehicle Maintenance | | **USE CASE TYPE** |
| **USE CASE ID:** | 12.1 | | Business Requirements:🞎 |
| **PRIORITY:** | High | | **System Analysis:** **x** |
| **SOURCE:** | Requirement List | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Operational Manager | | |
| **PRIMARY SYSTEM ACTOR** | n/a | | |
| **OTHER PARTICIPATING ACTORS:** |  | | |
| **OTHER INTERESTED STAKEHOLDERS:** |  | | |
| **DESCRIPTION:** | |  | | --- | | This use case describes the event where vehicle maintenance is added to the schedule. The manager will add to the schedule by providing all the necessary details, allowing the system to validate the input information and save it to a database. The use case concludes with the new maintenance or maintenance appointment being saved to the system. | | | |
| **PRE-CONDITION:** | The vehicle maintenance must not currently be in the schedule. The manager must be logged onto the system. | | |
| **TRIGGER:** | Manager requests to add a vehicle on the Maintenance schedule | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**:Operational manager Requests to add a vehicle to the maintenance schedule | **Step 2:** The system requests that the manager inputs the required information for a new vehicle maintenance   * Vehicle\_ID * Vehicle\_LicseneceNumber * DateTime * ServiceProvider\_ID * DistanceTraveled | |
|  | **Step 3:** Manager inputs the maintenance details:   * Vehicle\_ID * Vehicle\_LicseneceNumber * DateTime * ServiceProvider\_ID * DistanceTraveled | **Step 4:** System reads the entered maintenance details | |
|  |  | **Step 5:** System validates the format for the entered details | |
|  |  | **Step 6:** System retrieves the last schedule number and increments it by one | |
|  |  | **Step 7:**The system stores the maintenance details in the **Vehicle\_Maintenance** Tablewith its new unique number | |
|  |  | **Step 9:**  The System sends the manager a confirmation message | |
|  |  |  | |
|  |  |  | |
| **ALTERNATE COURSES:** | |  | | --- | | **ALT Step 6:** The entered information is invalid. The system displays an error message stating that the formatting is incorrect and asks the user to re-enter the information in a valid format.   * Return to step 3. | | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
| **CONCLUSION:** | The maintenance has been added to the maintenance schedule | | |
| **POST-CONDITION:** |  | | |
| **BUSINESS RULES** |  | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** |  | | |
| **ASSUMPTIONS:** |  | | |
| **OPEN ISSUES:** |  | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Mpho Mosotho Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | Update Vehicle Maintenance Schedule | | **USE CASE TYPE** |
| **USE CASE ID:** | 12.2 | | Business Requirements:🞎 |
| **PRIORITY:** | Medium | | **System Analysis:** **x** |
| **SOURCE:** | Requirement List | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Operational Manager | | |
| **PRIMARY SYSTEM ACTOR** | n/a | | |
| **OTHER PARTICIPATING ACTORS:** |  | | |
| **OTHER INTERESTED STAKEHOLDERS:** |  | | |
| **DESCRIPTION:** | |  |  | | --- | --- | | |  | | --- | | This use case describes the event where a Manager wants to update the details of maintenance that are already registered and entered onto the system as well as in the **Maintenance Schedule** table in the database, this may be due to outdated or incorrect information that was entered. The manager may also remove maintenance / maintenance appointment if necessary. The use case concludes with the system updating the required details on the system. | | | | |
| **PRE-CONDITION:** | The manager must be logged onto the system. | | |
| **TRIGGER:** | Schedule details need to be updated or removed | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**:Operational manager Requests to update or remove the details on the schedule using the system | **Step 2:** The System displays the update schedule menus | |
|  | **Step 3:** Manager selects the search schedule option | **Step 4:** ***System invokes use case 12.3 - “View Vehicle Maintenance schedule”*** | |
|  |  | **Step 5:** Step 5: The system displays all the maintenance details that is currently on the schedule, with the following details that can be edited:   * Vehicle\_ID * Vehicle\_LicseneceNumber * DateTime * ServiceProvider\_ID * DistanceTraveled | |
|  | **Step 6:** Manager enters all the relevant details that need to be updated and saves them | **Step 7:** The system reads the updated details entered by the user | |
|  |  | **Step 8:**  The System validates the format of the updated information entered by the Manager | |
|  |  | **Step 9:**  The System saves the edited schedule details in the **Maintenance schedule** table | |
|  |  | **Step 10:** The system displays the updated schedule details and notifies the Manager of the changes | |
|  |  |  | |
| **ALTERNATE COURSES:** | ALT Step 6: The user selects the remove vehicle procedure to be processed. The system prompts   |  | | --- | | The system prompts the user with a confirmation message asking if the user really wants to remove the maintenance / maintenance appointment from the schedule.  ALT ALT Step 6: The manager does not confirm removal.   * Return to step 5.   The system removes the maintenance / maintenance appointment from the schedule.   * Continue to step 11 | | | |
|  | ALT Step 9: The details are invalid; an error message is displayed telling the user that the format is invalid.   * Return to step 6. | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
| **CONCLUSION:** | The schedule details have been entered and stored on the system or removed where necessary. | | |
| **POST-CONDITION:** | The schedule details are now correctly stored on the system ensuring all functions relating to the schedule are handled correctly. | | |
| **BUSINESS RULES** |  | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** |  | | |
| **ASSUMPTIONS:** |  | | |
| **OPEN ISSUES:** |  | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Mpho Mosotho Date: 04-22-2019**  **Version: 2** |

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE NAME:** | View Vehicle Maintenance Schedule | | **USE CASE TYPE** |
| **USE CASE ID:** | 12.3 | | Business Requirements:🞎 |
| **PRIORITY:** | High | | **System Analysis’s : X** |
| **SOURCE:** | Requirement List | | System Design: 🞎 |
| **PRIMARY BUSINESS ACTOR** | Operational Manager | | |
| **PRIMARY SYSTEM ACTOR** | n/a | | |
| **OTHER PARTICIPATING ACTORS:** |  | | |
| **OTHER INTERESTED STAKEHOLDERS:** |  | | |
| **DESCRIPTION:** | |  | | --- | | This use case describes the event where the Operational Manager searches for a vehicle that has been scheduled for maintenance on the system. This will allow the manager to view all details associated with the schedule. The manager enters based on a search criteria where the system will then search for the corresponding vehicle on the schedule | | | |
| **PRE-CONDITION:** | The manager must be logged onto the system. | | |
| **TRIGGER:** | Manager requests to view Maintenance schedule | | |
| **TYPICAL COURSE** | **Actor Action** | **System Response** | |
| **OF EVENTS:** | **Step 1**:Operational manager Requests to search the schedule for a specific vehicle on the schedule | **Step 2:** The system displays the schedule search menu and requests the search criteria | |
|  | **Step 3:** Manager searches based on the following criteria   * Vehicle\_ID * Vehicle\_LicseneceNumber * DateTime * ServiceProvider\_ID * DistanceTraveled | **Step 4:** System reads the entered maintenance details | |
|  |  | **Step 5:** System validates the format for the entered details and searches using criteria provided in the **Maintenance schedule** table | |
|  |  | **Step 6:** System displays a list of all scheduled maintenances for that specific vehicle | |
|  | **Step 7:** The manager then selects the specific scheduled maintenance he/she wishes to see | **Step 8:** The system the displays all the information based on the manager’s selection:   * Vehicle\_ID * Vehicle\_LicseneceNumber * DateTime * ServiceProvider\_ID * DistanceTraveled | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
| **ALTERNATE COURSES:** | |  | | --- | | **ALT Step 6:** Once the search is completed, no information is displayed as no data matched the entered search criteria. An error notification is sent to the manager stating that they need to re-enter the search criteria.   * Return to step 3 | | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
| **CONCLUSION:** | The vehicle maintenance has been searched | | |
| **POST-CONDITION:** |  | | |
| **BUSINESS RULES** |  | | |
| **IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS** |  | | |
| **ASSUMPTIONS:** |  | | |
| **OPEN ISSUES:** |  | | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Mninikhaya Mavundla Date: 04-22-2019**  **Version: 2** |

|  |  |  |
| --- | --- | --- |
| USE CASE NAME: | Add New Location | USE CASE TYPE |
| USE CASE ID: | 13.1 | Business Requirements: ◻  System Analysis:◻  System Design: ◻ |
| PRIORITY: | High |
| SOURCE: | Ndila Transfers |
|  |  | |
| PRIMARY SYSTEM ACTOR | Booking consultant | |
| OTHER PARTICIPATING ACTORS: | * None | |
| OTHER INTERESTED STAKEHOLDERS: | * None | |
| DESCRIPTION: | The booking consultant wants to add a new location to the system. The system prompts the Consultant to enter all relevant information. The system must allow user to add a new location with details of the location such as Country, Province, City, town/Suburb, Street Number and name. The consultant enters all requested details, the system then captures, validates and stores all information regarding the new location. | |
| PRE-CONDITION: | * The consultant must be logged into the system. * The added new location must not already exist on the system | |
| TRIGGER: | The booking consultant wants to add a new location to the system | |
| TYPICAL COURSE | **Actor Action** | **System Response** |
| OF EVENTS: | **Step 1**: The booking consultant wants to add a new location to the system | * **Step 2**: The system requests details of the location such as Location name   [**Location\_name**]   * Province name   [**Province\_name**]   * City   [**City\_Name**]   * Suburb   [**Suburb\_name**]   * Street   **[Street\_name**]   * LocationType   [**Locationtype]** |
|  | **Step 3**: The booking consultant enters requested details of the all new relevant details  requested in **Step 2.** |  |
|  | **Step 4:** The Consultant selects the option save details. | **Step 5:** The system prompts the  Consultant to confirm details. |
|  | **Step 6:** The Consultant selects to confirm details. | **Step 7:** The system captures and validates the input data. Validation includes:   * + Location Name – cannot be null   + Province Name – cannot be null   + City Name – cannot be null   + Suburb Name – cannot be null   + street Name – cannot be null |
|  |  | **Step 8:** The system does not detect duplicates of Location  being added. |
|  |  | **Step 9:** The system generates a unique Land ID **[LocationID]** by adding 1 to the last **[LocationID]** found in the **Location** table. |
|  |  | **Step 10:** The system creates and stores the validated information for the Location in the **Location** table using the captured details in **Step 7** and generated Unique ID  in **Step 9.** |
|  |  | **Step 11:** The system creates and stores a new audit entry with the following details:   * Audit ID (Generated)   **[AuditID]**   * Audit Type ID **[AuditTypeID]** (Retrieved from the **AuditType** Table) * Audit Reference Table **[AuditRefTable]** (Table where transaction was performed) * Employee ID **[EmployeeID]** (Person initiating the transaction) – Retrieved from the **Employee** Table |
|  |  | **Step 12**: The system displays a confirmation message stating that the new location has successfully been added to the system. |
| ALTERNATE COURSES: | **[ALT] Step 6:** The consultant revokes the decision to confirm details and selects the option to cancel the confirmation.   Return to **Step 2**. | |
| **[ALT] Step 7:** The system fails to pass validation checks for input data. The system displays an error message stating which fields need attention.   Return to **Step 2**. | |
| **[ALT] Step 8**: The system detects a duplicate Land already on the system with the same **[locationName]**   The system displays an error message.   Return to **Step 2**. | |
|  | **[ALT] Step 9**: The system fails to find a **[LocationID]** in the **Location**  Table. System initializes the **[LocationID]** with 1.   Continue to **Step 10**. | |
| CONCLUSION: | The system captures the details of the new location and saves it in a database | |
| POST-CONDITION: | * The new location should be added to the systems database for later use. * The transaction details are saved in the **Audit** Table. | |
| BUSINESS RULES | * None | |
| IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS | * None | |
| ASSUMPTIONS: | * None | |
| OPEN ISSUES: | None | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Mavundla Mninikhaya Date: 04-22-2019**  **Version: 2** |

|  |  |  |
| --- | --- | --- |
| USE CASE NAME: | Update location | USE CASE TYPE |
| USE CASE ID: | 13.2 | Business Requirements: ◻  System Analysis:◻  System Design: ◻ |
| PRIORITY: | High |
| SOURCE: | Ndila Transfers |
| PRIMARY BUSINESS ACTOR | Booking consultant | |
| PRIMARY SYSTEM ACTOR | Booking consultant | |
| OTHER PARTICIPATING ACTORS: | * None | |
| OTHER INTERESTED STAKEHOLDERS: | * None | |
| DESCRIPTION: | The booking consultant wants to update a location. They select the Update location option and selects location they want to update. The system updates the location information. | |
| PRE-CONDITION: | Location must exsit in the systems database | |
| TRIGGER: | The booking consultant wants to update a location | |
| TYPICAL COURSE | **Actor Action** | **System Response** |
| OF EVENTS: | **Step 1**: The booking consultant wants to update a location and selects the location in given options | **Step 2**: The system retrieves the information of the location and displays it for the booking consultant |
|  | **Step 3**: The booking consultant enters the information they want to change on the existing location | **Step 4**: The system verifies the new information entered |
|  |  | **Step 5**: The system saves the changes made to the location table and the status of the location is updated. |
|  |  |  |
| ALTERNATE COURSES: | None | |
|  |  | |
| CONCLUSION: | The system updates the information of the selected location | |
| POST-CONDITION: | The location information has to be correct | |
| BUSINESS RULES | * None | |
| IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS | * None | |
| ASSUMPTIONS: | * None | |
| OPEN ISSUES: | None | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Mavundla Mninikhaya Date: 04-22-2019**  **Version: 2** |

|  |  |  |
| --- | --- | --- |
| USE CASE NAME: | Search Location | USE CASE TYPE |
| USE CASE ID: | 13.2 | Business Requirements: ◻  System Analysis:◻  System Design: ◻ |
| PRIORITY: | High |
| SOURCE: | Ndila Transfers |
| PRIMARY BUSINESS ACTOR | Booking consultant | |
| PRIMARY SYSTEM ACTOR | None | |
| OTHER PARTICIPATING ACTORS: | * None | |
| OTHER INTERESTED STAKEHOLDERS: | * None | |
| DESCRIPTION: | The booking consultant uses the location name to search and retrieve the details of the location from the location table in the database. | |
| PRE-CONDITION: | The location has to be in the systems database | |
| TRIGGER: | The booking consultant enters the location name to search and retrieve the details on the system | |
| TYPICAL COURSE | **Actor Action** | **System Response** |
| OF EVENTS: | **Step 1**: The booking consultant enters the location name to search on the system | **Step 2**: The system verifies the location name received. |
|  |  | **Step 3**:  The system compares the location name received with the location name hat are in the location table until a match is found |
|  |  | **Step 4**: The system retrieves the location details such as Country, Province, City, town/Suburb, Street Number and name given and displays it. |
|  |  |  |
| ALTERNATE COURSES: | Alt step 2: The Location name given does not meet the system requirements. The booking consultant has to type the correct Location name. | |
|  | Alt step 3: The system cannot find a match for Location name given and sends an error message. “no results found” | |
| CONCLUSION: | The system retrieves the information related to the location name given | |
| POST-CONDITION: | The location information searched has to be displayed for the booking consultant | |
| BUSINESS RULES | * None | |
| IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS | * None | |
| ASSUMPTIONS: | * None | |
| OPEN ISSUES: | None | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Mavundla Mninikhaya Date: 04-22-2019**  **Version: 2** |

|  |  |  |
| --- | --- | --- |
| USE CASE NAME: | Add Zone | USE CASE TYPE |
| USE CASE ID: | 14.1 | Business Requirements: ◻  System Analysis:◻  System Design: ◻ |
| PRIORITY: | High |
| SOURCE: | Ndila Transfers |
| PRIMARY BUSINESS ACTOR | Booking consultant | |
| PRIMARY SYSTEM ACTOR | Booking consultant | |
| OTHER PARTICIPATING ACTORS: | * None | |
| OTHER INTERESTED STAKEHOLDERS: | * None | |
| DESCRIPTION: | The booking consultant wants to add a zone onto the system database. They retrieve all the relevant information and a new zone and it is saved onto the system database. | |
| PRE-CONDITION: | The Zone rate cannot be an existing vehicle group in the system | |
| TRIGGER: | The booking consultant wants to add another vehicle group | |
| TYPICAL COURSE | **Actor Action** | **System Response** |
| OF EVENTS: | **Step 1**: The booking consultant wants to add a zone rate and selects the Add Zone option. | **Step 2**: The system displays the zone details from the Zone table:   * Zone Id   **[Zone\_ID]**   * Zone name   **[Zone\_name]**   * Zone range   **[zone\_range]** |
|  | **Step 3**: The booking consultant adds the necessary details of the new Zone rate such as name and range in kilometres **Step 2** | **Step 4**: The system verifies the information gathered |
|  | **Step 4:** The Consultant selects the option save details. | **Step 5:** The system prompts the  Consultant to confirm details. |
|  | **Step 6:** The Consultant selects to confirm details. **[ALT]** | **Step 7:** The system captures updated details for Zone and validates input data. Validation includes.   * **Zone \_name**-- cannot be null, must be in string * **Zone \_range-**- cannot be null, must be in number**[ALT]** |
|  |  | **Step 8:** The system does not detect duplicates of Zone  being added. **[ALT]** |
|  |  | **Step 9:** The system generates a unique Zone ID **[ZoneID]** by adding 1 to the last **[ZoneID]** found in the **Zone** table.  **[ALT]** |
|  |  | **Step 10:** The system creates and stores the validated information for the Zone in the **zone** table using the captured details in **Step 7** and generated Unique ID in **Step 9.** |
|  |  | **Step 11:** The system creates and stores a new audit entry with the following details:   * Audit ID (Generated)   **[AuditID]**   * Audit Type ID **[AuditTypeID]** (Retrieved from the **AuditType** Table) * Audit Reference Table **[AuditRefTable]** (Table where transaction was performed) * Employee ID **[EmployeeD]** (Person initiating the transaction) – Retrieved from the **Employee** Table |
|  |  | **Step 12**: The system displays a confirmation message stating that the new Zone has successfully been added to the system. |
| ALTERNATE COURSES: | **[ALT] Step 6:** The consultant revokes the decision to confirm details and selects the option to cancel the confirmation.   Return to **Step 2**. | |
| **[ALT] Step 7:** The system fails to pass validation checks for input data. The system displays an error message stating which fields need attention.   Return to **Step 2**. | |
| **[ALT] Step 8**: The system detects a duplicate Zone already on the system with the same **[ZoneName]**   The system displays an error message.   Return to **Step 2**. | |
| **[ALT] Step 9**: The system fails to find a **[ZoneID]** in the **Zone**  Table. System initializes the **[ZoneID]** with 1.   Continue to **Step 10**. | |
| CONCLUSION: | A zone rate is added onto the system database | |
| POST-CONDITION: | * The zone rate information has to be complete * The transaction details are saved in the **Audit** Table. | |
| BUSINESS RULES | * None | |
| IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS | * None | |
| ASSUMPTIONS: | * None | |
| OPEN ISSUES: | None | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Mavundla Mninikhaya Date: 04-22-2019**  **Version: 2** |

|  |  |  |
| --- | --- | --- |
| USE CASE NAME: | Search Zone | USE CASE TYPE |
| USE CASE ID: | 14.2 | Business Requirements: ◻  System Analysis:◻  System Design: ◻ |
| PRIORITY: | High |
| SOURCE: | Ndila Transfers |
| PRIMARY BUSINESS ACTOR | Booking consultant | |
| PRIMARY SYSTEM ACTOR | None | |
| OTHER PARTICIPATING ACTORS: | * None | |
| OTHER INTERESTED STAKEHOLDERS: | * None | |
| DESCRIPTION: | The booking consultant uses the Zone name to search and retrieve the details of the Zone from the Zone table in the database. | |
| PRE-CONDITION: | The Zone has to be in the systems database | |
| TRIGGER: | The booking consultant enters the Zone name to search and retrieve the details on the system | |
| TYPICAL COURSE | **Actor Action** | **System Response** |
| OF EVENTS: | **Step 1**: The booking consultant enters the Zone name to search on the system | **Step 2**: The system verifies the Zone name received. |
|  |  | **Step 3**:  The system compares the Zone name received with the Zone name that are in the Zone table until a match is found |
|  |  | **Step 4**: The system retrieves the Zone. details such as price and range in kilometres |
|  |  |  |
| ALTERNATE COURSES: | **Alt step 2**: The Zone name given does not meet the system requirements. The booking consultant has to type the correct Zone name. | |
|  | **Alt step 3:** The system cannot find a match for Zone name given and sends an error message. “no results found” | |
| CONCLUSION: | The system retrieves the information related to the Zone name given | |
| POST-CONDITION: | The Zone information searched has to be displayed for the booking consultant | |
| BUSINESS RULES | * None | |
| IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS | * None | |
| ASSUMPTIONS: | * None | |
| OPEN ISSUES: | None | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Mavundla Mninikhaya Date: 04-22-2019**  **Version: 2** |

|  |  |  |
| --- | --- | --- |
| USE CASE NAME: | Update Zone | USE CASE TYPE |
| USE CASE ID: | 14.3 | Business Requirements: ◻  System Analysis:◻  System Design: ◻ |
| PRIORITY: | High |
| SOURCE: | Ndila Transfers |
| PRIMARY BUSINESS ACTOR | Booking consultant | |
| PRIMARY SYSTEM ACTOR | Booking consultant | |
| OTHER PARTICIPATING ACTORS: | * None | |
| OTHER INTERESTED STAKEHOLDERS: | * None | |
| DESCRIPTION: | This use case describes the event The booking consultant wants to update a zone rate. They select the Update Zone rate option and selects the zone they want to update. The system prompts the Consultant to enter updated details for the Zone. The system then captures, validates and stores updated information The system updates the Zone information. | |
| PRE-CONDITION: | * Zone must exist in the systems database * The booking consultant must be logged into the system. | |
| TRIGGER: | The booking consultant wants to update a zone | |
| TYPICAL COURSE | **Actor Action** | **System Response** |
|  | **Step 1**: The booking consultant wants to update a zone | **Step 2.** The system Invokes Use Case **13.3 “Search zone”** |
| OF EVENTS: | **Step 3**: The booking consultant enters the information they want to change on the existing zone in the search box. | **Step 4**: The system retrieves the information of the Zone and displays it for the booking consultant from the zone table to be updated:   * Zone Id   **[Zone\_ID]**   * Zone name   **[Zone\_name]**   * Zone range   **[zone\_range]** |
|  | **Step 5**: The booking consultant enters the information they want to change on the existing Zone | **Step 6**: The system verifies the new information entered and confirms with the booking consultant. |
|  | **Step 7:** The Consultant selects to confirm changes to be updated for the Zone. **[ALT]** | **Step 8**: The system captures updated details for location and validates input data. Validation includes.   * **Zone \_name**-- cannot be null, must be in string * **Zone \_range-**- cannot be null, must be in number**[ALT]** |
|  |  | **Step 9:** The system creates and stores a new audit entry with the following details:   * Audit ID (Generated)   **[AuditID]**   * Audit Type ID **[AuditTypeID]** (Retrieved from the **AuditType** Table) * Audit Reference Table **[AuditRefTable]** (Table where transaction was performed) * Employee ID **[EmployeeD]** (Person initiating the transaction) – Retrieved from the **Employee** Table   . |
|  |  | **Step 10**: The system saves the changes made to the Zone table and the status of the zone is updated. |
|  |  |  |
| ALTERNATE COURSES: | **[ALT] Step 7:** The Consultant selects to cancel changes made to the location details.   * Return to **Step 3**.   **[ALT] Step 8:** The system is unable to validate the details entered and displays an invalid details provided error message.  Return to **Step 3.** | |
|  |  | |
| CONCLUSION: | * The system updates the information of the selected Zone * The transaction details are saved in the **Audit** Table. | |
| POST-CONDITION: | The Zone information has to be correct | |
| BUSINESS RULES | * None | |
| IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS | * None | |
| ASSUMPTIONS: | * None | |
| OPEN ISSUES: | None | |



|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Mavundla Mninikhaya Date: 04-22-2019**  **Version: 2** |

|  |  |  |
| --- | --- | --- |
| USE CASE NAME: | Remove Zone | USE CASE TYPE |
| USE CASE ID: | 14.4 | Business Requirements: ◻  System Analysis: ◻  System Design:◻ |
| PRIORITY: | High |
| SOURCE: |  |
| PRIMARY BUSINESS ACTOR | Manager | |
| PRIMARY SYSTEM ACTOR | None | |
| OTHER PARTICIPATING ACTORS: | * None | |
| OTHER INTERESTED STAKEHOLDERS: | * Owner | |
| DESCRIPTION: | This use case describes the events where a manager would like to remove a Zone from being used for any new transactions. This involves the manager selecting the zone to remove and confirming that the zone should not be accessible any longer. | |
| PRE-CONDITION: | The Zone should already exist in the system database  The Consultant Owner must be logged into the system | |
| TRIGGER: | Manager/Owner wants to remove a zone | |
| TYPICAL COURSE | **Actor Action** | **System Response** |
|  | **Step 1**: The manager Owner requests to remove a Location from the system selects the ‘remove Zone option | **Step 2:** System checks to see if the manager has authority to remove a Zone**[ALT]** |
| OF EVENTS: |  | **Step 3:** The system invokes U.C  14.3 “Search Zone”. |
|  | **Step 4:**The booking consultant enters the location name to search on the system | **Step 5**: System then displays all the Zones matches in the database[ALT] |
|  |  | **Step 6:** System prompts the manager to select the Zone they wish to remove |
|  | **Step 7**: Manager selects the Zone they would like to remove | **Step 8**: System requests confirmation that the selected Zone is the correct one to be removed |
|  | **Step 9**: Manager confirms that they want to remove the selected Zone**[ALT]** | **Step 10:** System then disables and removes the selected the Zone to be used again in the database. |
|  |  | **Step 11:** The system creates and stores a new audit entry with the following details:  • Audit ID (Generated)  **[AuditID]**  • Audit Type ID  **[AuditTypeID**] (Retrieved from the Audit Type Table)  • Audit Reference Table [**AuditRefTable]** (Table where transaction was performed)  • Employee ID  **[EmployeeID]** (Person initiating the transaction) – Retrieved from the Employee Table |
|  |  | **Step 12**: System then displays a confirmation message to let the manager that the removal has been successful |
|  |  |  |
|  |  |  |
| ALTERNATE COURSES: | **[Alt Step 2**]: System finds the manager to not have authority to remove a Zone. Terminates use case | |
|  | **[Alt] step 5:** The Zone name given does not meet the system requirements or any matches   * Return to **Step 4.** | |
|  | **[ALT] Step 9:** The Consultant revokes the decision to confirm details and selects the option to cancel the confirmation.   * Return to **Step 4**. | |
|  |  | |
| CONCLUSION: | The manager selected Zone has successfully been removed from the system | |
| POST-CONDITION: | Zone has been removed from the system. | |
| BUSINESS RULES | * None | |
| IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS | * None | |
| ASSUMPTIONS: | * Zone already exists in the system database | |
| OPEN ISSUES: | None | |

|  |
| --- |
| **Siyaya Travel Assist** |
| **Author (s): Mavundla Mninikhaya Date: 04-22-2019**  **Version: 2** |

|  |  |  |
| --- | --- | --- |
| USE CASE NAME: | Generate Quote | USE CASE TYPE |
| USE CASE ID: | **15.1** | Business Requirements:  System Analysis:  System Design: |
| PRIORITY: | High |
| SOURCE: | Requirements Description & Detail |
| PRIMARY BUSINESS ACTOR: | The booking Consultant | |
| PRIMARY SYSTEM ACTOR: | None | |
| OTHER PARTICIPATING ACTORS: | None | |
| OTHER INTERESTED STAKEHOLDERS: | None | |
| DESCRIPTION: | This use case describes the process of a Consultant generating a Quote. The use case starts with the Consultant selecting to generate a Quote, it continues with the Consultant entering the dates for the Quote, and the system generating the Quote. The use case concludes when the Quote is generated and displayed. | |
| PRE-CONDITION: | * The Consultant needs to be logged in to the system. | |
| TRIGGER: | The Consultant selects to generate an Quote . | |
| TYPICAL COURSE | **Actor Action** | **System Response** |
| OF EVENTS: | **Step 1.** The Consultant selects to generate a Quote . | Step 2. The system prompts the Consultant to enter the start and end date for the Quotation . |
| **Step** 3. The Consultant provides the start and end date. | Step 4. The system requests details of booking such as,   * Booking\_Reference   [Booking\_Reference] |
| **Step** 5. The Consultant enters details and selects to continue with generating the Quote . | Step 6. The system validates that the dates and details are provided are before the current date. [alt] |
|  | Step 7: The system retrieves the following information for the Quote from the Quote and Booking table:   * The quote reference,   [Quote\_Reference\_No]  [Quote]   * Booking\_Reference   [Booking\_Reference]   * Quote   [Quote \_ID]  [Payment\_status]  [Date] |
|  | Step 8. The system notifies the Consultant that the Quote has been generated. |
|  | Step 9. The system displays the generated Quote . |
| ALTERNATE COURSES: | ALT 6: The entered dates are not before the current date. The system displays a notification and returns to Step 2. | |
| CONCLUSION: | The use case concludes when an Quote has been created. | |
| POST-CONDITION: | * No changes have been made to the system after the Quote has been generated. | |
| BUSINESS RULES: |  | |
| IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS: | * None | |
| ASSUMPTIONS: | * None | |
| OPEN ISSUES: | * There are currently no open issues | |

## 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.

|  |
| --- |
| 3. process modelling |

## 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.

Context Appendix

Functional Appendix

High Levels

















Book Trip Appendix













Mid Level































## 

## 

## 

## 

## 



## 



























Primitives























****





****







****



## 

## 

## 

## 

## 

## 

## 

## 

## 





## 

## 

## 

## 



## 

## 

## 

## Conclusion

The section seen was a combination of the 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.

|  |
| --- |
| 4. uml modelling |

## Introduction

This section of the document contains UML Activity diagrams for each of our functional requirements. The activity diagrams will show the interaction between the user and the system by means of swim-lanes in each diagram.







































## 

## 



****

****











****

## 









## CONCLUSION

The section above provided an overview of how each user interacts with the system and how the system will respond when certain tasks are completed by the user.

|  |
| --- |
| 5. data modelling |

## 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.

## Conclusion

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

|  |
| --- |
| 6. 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 also 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.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UseCase** | **Flow Line Description** | **Purpose** | **When Used** | **Entities and Attributes** | **Logical Layout** |
| 1.1.1 | Add new access level | To trigger the process of adding an access level | Adding a new access level to the system |  | Provided by the employee |
| 1.1.4 | Enter access level details | To keep details about the new access level | Adding a new access level to the system | “AccessLevel\_ID”  “AccessLevel\_Name”  “AccessLevel\_Description”  AccessLevel Entity | Provided by the employee |
| 1.1.6 | Store access level details | To keep the new access level in the system | Adding a new access level to the system | AccessLevel\_ID”  “AccessLevel\_Name”  “AccessLevel\_Description”  **AccessLevel Entity** | Selected by the employee |
| 1.2.1 | Update access level | To trigger the process of updating an access level | When an edit is required to an existing access level |  | Selected by the employee |
| 1.2.6 | Enter new details to access level | To update the access level with the new details entered | When an edit is required to an existing access level | AccessLevel\_ID”  “AccessLevel\_Name”  “AccessLevel\_Description”  AccessLevel Entity | Provided by the employee |
| 1.2.9 | Store updated details | To update the access level with the new details entered | When an edit is required to an existing access level | AccessLevel\_ID”  “AccessLevel\_Name”  “AccessLevel\_Description”  **AccessLevel Entity** | Selected by the employee |
| 1.3.1 | Request to search for an access level | To trigger the process of searching for an access level | When certain details are needed about an access level |  | Selected by the employee |
| 1.3.4 | Enter search details | To Know which details are to be searched for | When certain details are needed about an access level | AccessLevel\_ID”  “AccessLevel\_Name”  **AccessLevel Entity** | Provided by the employee |
| 1.3.7 | Display search results | So that the employee can see the details they were searching for | When certain details are needed about an access level | AccessLevel\_ID”  “AccessLevel\_Name”  “AccessLevel\_Description”  AccessLevel Entity | Provided by the employee |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UseCase** | **Flow Line Description** | **Purpose** | **When Used** | **Entities and Attributes** | **Logical Layout** |
| 2.1.1 | Request to register an employee | To trigger the process of adding an employee | Adding a new employee to the system |  | Provided by the employee |
| 2.1.4 | Enter employee details | To keep details about the new employee | Adding a new employee to the system | “EMPID”  “EMP\_Name”  “EMP\_Surname”  “AuditID”  “EMP\_TypeID”  “EMP\_EmailAddress”  “EMP\_Contact”  “EMP\_Password”  “EMP\_IDNum”  “TitleID”  “GenderID”  **Employee Entity** | Provided by the employee |
| 2.1.7 | Store employee details | To keep the new employee in the system | Adding a new employee to the system | “EMPID”  “EMP\_Name”  “EMP\_Surname”  “AuditID”  “EMP\_TypeID”  “EMP\_EmailAddress”  “EMP\_Contact”  “EMP\_Password”  “EMP\_IDNum”  “TitleID”  “GenderID”  **Employee Entity** | Selected by the employee |
| 2.2.1 | Request to update employee | To trigger the process of updating an employee | When an edit is required to an existing employee |  | Selected by the employee |
| 2.2.6 | Enter new details to employee | To update the employee with the new details entered | When an edit is required to an existing employee | “EMPID”  “EMP\_Name”  “EMP\_Surname”  “AuditID”  “EMP\_TypeID”  “EMP\_EmailAddress”  “EMP\_Contact”  “EMP\_Password”  “EMP\_IDNum”  “TitleID”  “GenderID”  **Employee Entity** | Selected by the employee |
| 2.2.9 | Store updated employee details | To update the employee with the new details entered | When an edit is required to an existing employee | “EMPID”  “EMP\_Name”  “EMP\_Surname”  “AuditID”  “EMP\_TypeID”  “EMP\_EmailAddress”  “EMP\_Contact”  “EMP\_Password”  “EMP\_IDNum”  “TitleID”  “GenderID”  **Employee Entity** | Selected by the employee |
| 2.3.1 | Request to search for an employee | To trigger the process of searching for an employee | When certain details are needed about an employee |  | Selected by the employee |
| 2.3.4 | Enter employee search details | To Know which details are to be searched for | When certain details are needed about an employee | “EMP\_Name”  “EMP\_Surname”  “EMP\_TypeID”  **Employee entity** |  |
| 2.3.7 | Display search results | So that the employee can see the details they were searching for | When certain details are needed about an employee | “EMP\_Name”  “EMP\_Surname”  “EMP\_TypeID”  **Employee entity** | Selected by the employee |

| **UseCase** | **Flow Line Description** | **Purpose** | **When Used** | **Entities and Attributes** | **Logical Layout** |
| --- | --- | --- | --- | --- | --- |
| 3.1.1 | Request to register an employee type | To trigger the process of adding an employee type | Adding a new employee type to the system |  | Selected by the employee |
| 3.1.4 | Enter employee type details | To keep details about the new employee type | Adding a new employee type to the system | “EMP\_TypeID”  “EMP\_TypeName”  “EMP\_TypeDes”  **Employee type Entity** | Provided by the employee |
| 3.1.7 | Store employee type details | To keep the new employee type in the system | Adding a new employee type to the system | “EMP\_TypeID”  “EMP\_TypeName”  “EMP\_TypeDes”  **Employee type Entity** | Provided by the employee |
| 3.2.1 | Request to update employee type | To trigger the process of updating an employee type | When an edit is required to an existing employee type |  | Provided by the employee |
| 3.2.6 | Enter new details to employee type | To update the employee type with the new details entered | When an edit is required to an existing employee type | “EMP\_TypeName”  “EMP\_TypeDes”  **Employee type Entity** | Provided by the employee |
| 3.2.9 | Store updated employee type details | To update the employee type with the new details entered | When an edit is required to an existing employee type | “EMP\_TypeName”  “EMP\_TypeDes”  **Employee type Entity** | Selected by the employee |
| 3.3.1 | Request to search for an employee type | To trigger the process of searching for an employee type | When certain details are needed about an employee type |  | Provided by the employee |
| 3.3.4 | Enter employee type search details | To Know which details are to be searched for | When certain details are needed about an employee type | “EMP\_TypeName”  “EMP\_TypeDes”  **Employee type Entity** | Provided by the employee |
| 3.3.7 | Display search results | So that the employee can see the details they were searching for | When certain details are needed about an employee type | “EMP\_TypeID”  “EMP\_TypeName”  “EMP\_TypeDes”  **Employee type Entity** | Selected by the employee |
| 3.3.1 | Request to remove employee type | To trigger the process of removing an employee type | So that the employee type can no longer be used anymore |  | Provided by the employee |
| 3.3.5 | Enter the employee type details | To know which employee type is to be removed | So that the employee type can no longer be used anymore | “EMP\_TypeID”  “EMP\_TypeName”  “EMP\_TypeDes”  **Employee type Entity** | Provided by the employee |
| 3.3.8 | Disable the employee type | To prevent further use of the employee type | So that the employee type can no longer be used anymore | “EMP\_TypeID”  “EMP\_TypeName”  “EMP\_TypeDes”  **Employee type Entity** | Selected by the employee |
|  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Use Case No.** | **Flow Line** | **Description** | **Purpose** | **When it will be Used** | **Entities and Attributes** | **Logical Layout** |
| 4.1 Search driver | 4.1a | Search driver prompt | Triggering input to begin the search driver process | When the booking consultant wants to search a driver | - | Requested to search by booking consultant |
|  | 4.1b | Character information | To enter character information | When the booking consultant wants to enter characters of the driver name they are searching for | **DRIVER**   * Driver\_ID * Driver\_Name | Entered by booking consultant |
| 4.2 Check driver availability | 4.2a | Check driver availability prompt | To trigger the system to start the check availability process | When the booking consultant wants to check available drivers | **SLOT**   * Slot\_ID * Slot\_Date * Slot\_Time | Requested to check by booking consultant |
| 4.3 Assign driver to trip | 4.3a | Assign driver availability prompt | To trigger the process that assigns a vehicle to a trip | When the booking wants to confirm a booked trip | **SLOT**   * Slot\_ID * Slot\_Date * Slot\_Time   **DRIVER**   * Driver\_ID * Driver\_Name | Selected by booking consultant |
| 4.4 Outsource driver | 4.4a | Outsource driver output | To trigger the outsource a driver process | When there are no internal drivers available and the booking consultant wants to assign a driver to a trip | **SLOT**   * Slot\_ID * Slot\_Date * Slot\_Time   **OutsourceDriver**   * OutsourceDriver\_ID * OutsourceDriver\_Name | Selected by the booking consultant |
| 5.1 Add vehicle | 5.1 a | Add vehicle prompt | To begin the process that adds another vehicle to the system | When there is a new vehicle to be added to the system | **VEHICLE**   * Vehicle\_ID * Vehicle\_Model * Vehicle\_Colour\_Description   **VEHICLE\_MAKE**   * Vehicle\_Make\_Name, * Vehicle\_Make\_Description * VehicleMake\_ID   **VEHICLE MAINTENANCE**   * VehicleMaintenance\_ID * ServiceProvider * Vehicle\_ID | Requested by the booking consultant |
| 5.2 Search vehicle | 5.2a | Search vehicle prompt | Triggers the system to begin the process that searches a vehicle | When the booking consultant wants to search for a vehicle | **VEHICLE**   * Vehicle\_ID * Vehicle\_Model * VehicleMake\_ID * VehicleMaintenance\_ID * VehicleColour\_ID * VehicleLicencePlates | Requested by the booking consultant |
| 5.3 Confirm trip | 5.3a | Confirm trip prompt | This starts the system process which confirms a booking | When the booking consultant wants to send confirmation messages of trip to the driver and the passenger | **SLOT**   * BookingReference\_ID * DeparturePoint * DestinationPoint * Slot\_Time   **PASSENGER**   * Passenger\_ID * Passenger\_Name * Passenger\_Contact   **DRIVER**   * Driver\_ID * Driver\_Name * Driver\_Contact | Requested by the booking consultant |
| 5.4 Check vehicle availability | 5.4a | Check driver availability prompt | This triggers the process that checks the drivers that are available | When a list of available vehicles | **BOOKING**   * BookingReference\_ID * Passenger\_ID * Driver\_ID * DeparturePoint * DestinationPoint * Time   **VEHICLE GROUP**   * Number\_of\_Passengers   **SLOT**   * Slot\_Date * Slot\_Time | Selected by the bookng consultant |
| 5.5 Assign vehicle to trip | 5.5a | Assign vehicle prompt | This begins the process that assigns a vehicle to a trip | When the booking consultant wants to confirm a booking and needs to assign a vehicle to a trip | **SLOT**   * Slot\_ID * Slot\_Date * Slot\_Time | Requested by the booking consultant |
|  | 5.5b | Selected vehicle information | This triggers the process the captures the details of a chosen vehicle | When the booking consultant selects a vehicle they want to assign to a trip | **SLOT**   * Slot\_ID * Slot\_Date * Slot\_Time | Requested by the booking consultant |
| 5.6 Outsource vehicle | 5.6a | Outsource vehicle prompt | Triggers the process that assigns an outsourced driver to a trip | When the booking consultant wants to confirm a booking and assigns an outsourced driver | **SLOT**   * Slot\_ID * Slot\_Date * Slot\_Time   **OUTSOURCE\_VEHICLE**   * OutsourceVehicle\_ID * OutSource\_Vehicle\_Model * OutSource\_Make | Requested by the booking consultant |
| 6.1 Create vehicle group activity | 6.1a | Add new vehicle group prompt | This begins the process of adding a new vehicle group in the system | When the booking consultant wants to add a new vehicle group into the system | **VEHICLE GROUP**   * Vehicle group ID, * VehicleGroup\_Name * VehicleGroup\_Description | Requested by the booking consultant |
|  | 6.1b | New vehicle group information | This begins the process that captures the new vehicle group information | When a new vehicle group is to be added by the booking consultant | **VEHICLE GROUP**   * Vehicle group ID, * VehicleGroup\_Name * VehicleGroup\_Description | Requested by the booking consultant |
| 6.2 Search vehicle group | 6.2a | Search vehicle group prompt | This triggers the process that searches for a group activity | When the booking consultant wants to search for a vehicle group | **VEHICLE GROUP**   * Vehicle group ID, * VehicleGroup\_Name * VehicleGroup\_Description | Requested by the booking consultant |

| Use case | Description | Purpose | When will it be used | Entities and Attributes | Logical Layout |
| --- | --- | --- | --- | --- | --- |
| 7.1 Add Client | Request to add new client | Contains the details used to initiate the Add Client process | When the booking consultant adds the new Client to the system |  | Provided by the client |
| New Client Details | Contains new client details | When the Booking consultant captures the details of the new client | Entity:Client  Attributes  Client\_Name  Client\_Email  Client\_Tel  Client\_Reference  Client\_Address  Client\_Type | Provided by the client |
| 7.2 Search Client | Request to Search a client | Contains details used to initiate client search | When the Booking consultant wants to search a client | None | Selected by the Booking consultant |
| Client Name | To retrieve a client being searched | The booking consultant enters the name of the client | Entity: Client  Client\_Name  Client\_Email  Client\_Tel  Client\_Reference  Client\_Address  Client\_Type | Provided by the Booking consultant |
| Select Specific client | The option select the specific client they want | When the booking consultant wants to select the specific client and view details of the client |  | Selected by the Booking consultant |
| 7.3 Update Client | Request to update client details | Contains all the details needed to initiate the update client details | When the Booking consultant wants to update the client details |  | Provided by the client |
|  | The Updated Client details | Contains all the updated client details | When the Booking consultant enters the Updated client details |  | Selected by the booking consultant |
|  | Confirm Updated Details | The Booking consultant selects confirm the changes made | When the Booking Consultant wants to confirm the changes made | Entity: Client  Client\_Name  Client\_Email  Client\_Tel  Client\_Reference  Client\_Address  Client\_Type | Selected by the booking consultant |
| 8.1 Add ClientType | Request to add new clientType | Contains the details used to initiate the Add ClientType process | When the Operation Manager adds the new ClientType to the system |  | Operation Manager Provides the details |
| New ClientType Details | Contains new clientType details | When the Operations Manager captures the details of the new clientType | Entity:Client\_Type  Attributes  **Client\_Type\_Name**  **Client\_Type\_Description** | Operations Manager Provides the details |
| 8.2 Search ClientType | Request to Search a clientType | Contains details used to initiate clientType search | When the Booking consultant wants to search a clientType | None | Provided by the Booking consultant |
| ClientType Name | To retrieve clientType being searched | The booking consultant enters the name of the clientType | Entity:Client\_Type  Attributes  **Client\_Type\_Name**  **Client\_Type\_Description** | Provided by the Booking consultant |
| Select Specific clientType | The option selects the specific clientType they want | When the booking consultant wants to select the specific clientType and view details of the clientType |  | Selected by the Booking consultant |
|  |  |  |  |  |  |

| Use case | Description | Purpose | When will it be used | Entities and Attributes | Logical Layout |
| --- | --- | --- | --- | --- | --- |
| 8.3 Update ClientType | Request to update clientType details | Contains all the details needed to initiate the update clientType details | When the Booking consultant wants to update the clientType details |  | Selected by Booking consultant |
|  | The Updated ClientType details | Contains all the updated clientType details | When the Booking consultant enters the Updated clientType details | Entity:Client\_Type  Attributes  **Client\_Type\_Name**  **Client\_Type\_Description** | Provided by the Booking consultant |
|  | Confirm Updated Details | The Booking consultant selects confirm the changes made | When the Booking Consultant wants to confirm the changes made | Entity:Client\_Type  Attributes  **Client\_Type\_Name**  **Client\_Type\_Description** | Selected by the Booking consultant |
|  |  |  |  |  |  |

| Use case | Description Flow Line | Purpose | When will it be used | Entities and Attributes | Logical Layout |
| --- | --- | --- | --- | --- | --- |
| 9.1 Create Booking | Request to Create a booking | Contains all the details needed to initiate Create Booking Process | When the Booking Consultant wants to create a Booking |  | Provided by client |
|  | Client Name/Reference | Contains details of the Client making a booking | When the consultant wants to retrieve details of the client |  | Provided by client |
|  | [ALT] Add new Client | Contains all the details of the new Client | When the booking consultant wants to create new Client |  | Selected by the Booking consultant |
|  | The Pickup and dropOff details Details | Contains all the required Pickup and dropOff details | When the Booking consultant wants to generate a quote for booking | Entity:Client\_Type  Attributes  **Client\_Type\_Name**  **Client\_Type\_Description** | Provided by client |
|  | Passenger Details | Contains all the required to create a new Passenger | When the Booking consultant wants to enter details of the Passenger | **Entity: Passenger**  **Passenger Name**  **Passenger Surname**  **Passenger PhoneNo** | Provided by client |
|  | PickUp Instruction | Contains all the required details needed for instruction | When the manager wants to enter the instructions of PickUp |  | Provided by client |
|  | Compiled Booking Details | Contains all the details required to create a new Booking | When the manager wants to enter the Booking details | **Entity: Booking\_Trip**  **Date\_of\_PickUp**  **Date\_of\_Arrival**  **Time\_Of\_PickUp**  **Time\_Of\_Arrival**  **Booking\_Status**  **Pickup\_Location\_ID**  **DropOff\_Location\_ID**  **Number\_Of\_Passengers**  **PickUp\_Instruction**  **Entity : Invoice Invoice\_Number**  **Invoice\_Date**  **Payment\_Status**  **Booking\_Reference**  **Booking\_Cost** | System generated a compiled booking details |
| 9.2 Search Booking | Request to Search a Booking | Contains details used to initiate Search Booking | When the Booking consultant wants to search a Booking | None | Selected by Booking Consultant |
| Booking Reference | To retrieve Booking being searched | The booking consultant enters the Booking reference | Entity:Booking\_Trip  Attributes  **Booking\_Reference** | Provided by Booking consultant |
| Select Specific Booking | The option selects the specific Booking they want | When the booking consultant wants to select the specific Booking and view details |  | Selected by Booking consultant |
| 9.3 Update Booking | Request to update Booking details | Contains all the details needed to initiate the update Booking details | When the Booking consultant wants to update the Booking details |  | Provided by the client |
| The Updated Booking details | Contains all the updated Booking details | When the Booking consultant enters the Updated Booking details |  | Provided by the client |
|  | Confirm Updated Details | The Booking consultant selects confirm the changes made | When the Booking Consultant wants to confirm the changes made | **Entity: Booking\_Trip**  **Date\_of\_PickUp**  **Date\_of\_Arrival**  **Time\_Of\_PickUp**  **Time\_Of\_Arrival**  **Booking\_Status**  **Pickup\_Location\_ID**  **DropOff\_Location\_ID**  **Number\_Of\_Passengers**  **PickUp\_Instruction**  **Entity : Invoice Invoice\_Number**  **Invoice\_Date**  **Payment\_Status**  **Booking\_Reference**  **Booking\_Cost** | System confirms the updated details |
| 9.4 Cancel Booking | Request to Cancel a Booking | The Booking consultant selects the option to cancel the Booking | When the Booking consultant wants to cancel a Booking |  | Provided by the client |
|  | Booking Reference | Contains the details of the Booking that needs to be canceled | When the Booking consultant wants to retrieve the Booking to cancel | Entity: Booking\_Trip  Attributes:  Booking\_Reference | Provided by the client |
|  | Cancellation Reasons | Contains the details of cancelling the Booking | When the Booking consultant want to enter the Cancellation reasons |  | Provided by the client |
| 9.5 Confirm Booking | Request to Confirm a Booking | The Booking consultant selects to confirm a Booking | When the Booking consultant wants to confirm a Booking |  | Email |
|  | Booking Reference | Contains the details of the Booking that needs to be Updated | When the Booking consultant wants to retrieve the Booking to Updated |  | Retrieved from email |
|  | Select a Vehicle | Contains details needed to select a Vehicle | When the Booking consultant wants to assign a vehicle to a Booking | Entity:Vehicle  Attributes  Vehicle\_ID  Entity: Slot  Slot\_Date  Slote\_Time | Selected by Booking consultant |
|  | Select a Driver | Contains details needed to select a Driver | When the Booking consultant wants to assign a Driver to a Booking | Entity:Driver  Attributes  Driver  Entity: Slot  Slot\_Date  Slote\_Time | Selected by Booking consultant |

| UseCase | Flow Line | Description | Purpose | When Used | Entities and Attributes | Logical Layout |
| --- | --- | --- | --- | --- | --- | --- |
| 10.1 | Add to Schedule | Notification of Scheduling | Notifies the manager of any maintenances or maintenance appointments that are to the schedule. | When the booking consultant wants to add a booking to the schedule | N/A | Notification the system |
|  |  | ALT - Invalid Appointment/Maintenance Details Error Message | Notifies the manager that invalid appointment or maintenance details are inserted when adding to the schedule. | When invalid booking details are inserted when adding to the schedule. | N/A | A message that notifies the manager that invalid Booking details were received |
| 10.2 | Update Schedule | Updatable Schedule Details Edit Fields | Displays the updatable schedule details edit fields. | When a manager updates the schedule. | N/A | Displayed on the system. |
|  |  | ALT - Schedule Details Validation Error Message | Notifies the manager that there is a validation error with the schedule details. | When invalid details are inserted when maintaining a schedule. | N/A | A message that notifies the consultant that invalid details were inserted when maintaining the schedule. |
|  |  | ALT - Request Confirmation of Schedule Removal | Prompts the consultanat for confirmation on removing a schedule. | When the consultant removes a schedule. | N/A | Prompts displayed on the system. |
|  |  | Notification of Successful Schedule Details Update or Removal | Notifies the consultant that schedule details were successfully updated or removed. | When schedule details are successfully updated or removed. | N/A | Notification on the system. |
| 10.3 | View Schedule | Displayed List of Schedule Search Results | Displays a list of schedules that were searched for by the consultant. | When the booking consultant wants to search for a booking on the schedule | **Slot** table:   * **BookingRefernce** * **Client\_ID** * **Driver\_ID** * **Date** * **Time** * **ContactPerson\_ID** * **DestinationLocation** * **TripDuration** | Displayed on the system. |
|  |  | Displayed Specific Search Result Details | Displays specific search result details that the consultanat searched for. | When the consultant wants to view the details of a specific Booking in the schedule | **Slot** table:   * **BookingRefernce** * **Client\_ID** * **Driver\_ID** * **Date** * **Time** * **ContactPerson\_ID** * **DestinationLocation** * **TripDuration** | Displayed on the system. |
|  |  | ALT - Invalid Schedule Search Criteria Error Message | Notifies the consultant that invalid search criteria were entered when viewing the schedule. | When the consultant enters invalid data when viewing a schedule. | **N/A** | A message that notifies the consultant that invalid details were inserted viewing the schedule. |
| 11.1 | Booking Report |  |  |  |  |  |
| 11.2 | Driver-Trip Report |  |  |  |  |  |
| 11.3 | Mileage Report |  |  |  |  |  |
| 12.1 | Schedule Vehicle Maintenance Schedule | Notification of Scheduling | Notifies the manager of any maintenances that are to the schedule. | When a maintenance is added to the schedule. | None | Notification on the system. |
|  |  | ALT – Invalid Maintenance Details Error Message | Notifies the manager that invalid maintenance details are inserted when adding to the schedule. | When invalid maintenance details are inserted when adding to the schedule. | None | A message that notifies the manager that invalid maintenance details were received. |
| 12.2 | Update Vehicle Maintenance Schedule | Updatable Schedule Details Edit Fields | Displays the updatable schedule details edit fields. | When an operational manager maintains the schedule. | None | Displayed on the system. |
|  |  | ALT - Schedule Details Validation Error Message | Notifies the operational manager that there is a validation error with the schedule details. | When invalid details are inserted when maintaining a schedule. | None | A message that notifies the operational manager that invalid details were inserted when maintaining the schedule. |
|  |  | ALT - Request Confirmation of Schedule Removal | Prompts the operational manager for confirmation on removing a schedule. | When the operational manager removes a schedule. | None | Prompts displayed on the system. |
|  |  | Notification of Successful Schedule Details Update or Removal | Notifies the operational manager that schedule details were successfully updated or removed. | When schedule details are successfully updated or removed. | None | Notification on the system. |
| 12.3 | View Vehicle Maintenance Schedule | Displayed List of Schedule Search Results | Displays a list of schedules that were searched for by the operational manager. | When the operational manager searches for schedules. |  | Displayed on the system. |
|  |  | Displayed Specific Search Result Details | Displays specific search result details that the operational manager searched for. | When the operational manager searches for a specific schedule. |  | Displayed on the system. |
|  |  | ALT - Invalid Schedule Search Criteria Error Message | Notifies the operational manager that invalid search criteria were entered when viewing the schedule. | When the operational manager enters invalid data when viewing a schedule. | N/A | A message that notifies the operational manager that invalid details were inserted viewing the schedule. |

| UseCase  Number | Flow  Line | Description | Purpose | When it will be  Used | Entities and Attributes | Logical  Layout |
| --- | --- | --- | --- | --- | --- | --- |
| 13.1 Add Location | **13.1.1** | Request Add Location Details | Triggering input to start the adding Location  process. | When a new Location is to be added on the  system. | Booking Consultant (Location)   * Location name   [**Location\_name**]   * Province name   [**Province\_name**]   * City   [**City\_Name**]   * Suburb   [**Suburb\_name**]   * Street   **[Street\_name**]   * LocationType   [**Locationtype]** | Requested by Booking Consultant . |
| **13.1.2** | Location Details | Details of new Location to be used to add new Location . | When a new Location is to be added on the system. | * Location name   [**Location\_name**]   * Province name   [**Province\_name**]   * City   [**City\_Name**]   * Suburb   [**Suburb\_name**]   * Street   **[Street\_name**]   * LocationType   [**Locationtype]** | Entered by Booking Consultant . |
| **13.1.3** | Save Location Details | The Booking Consultant indicates selection to save details for new  Location . | When a new Location is to be added on the system. | Booking Consultant (Location)   * Location name   [**Location\_name**]   * Province name   [**Province\_name**]   * City   [**City\_Name**]   * Suburb   [**Suburb\_name**]   * Street   **[Street\_name**]   * LocationType   [**Locationtype]** | Selected by Booking Consultant . |
| **13.1.4** | Confirm Location Details | The Booking Consultant indicates confirmation of  adding new Location . | When the system prompts the Booking Consultant to confirm new  Location details. | Booking Consultant (Location) | Selected by Booking Consultant . |
| **13.1 [ALT]** | Revoked Confirmation Details | The Booking Consultant revokes confirmation of adding new Location . | When the system prompts the Booking Consultant to confirm new  Location details. | Booking Consultant (Location) | Selected by Booking Consultant . |
| 13.2 Search Location | **13.2.1** | Request Search Location Details. | Triggering input to start the Location Viewing process. | When a specific Location is to be viewed on the  system. | Booking Consultant (Location)   * Location name   [**Location\_name**]   * Province name   [**Province\_name**]   * City   [**City\_Name**]   * Suburb   [**Suburb\_name**]   * Street   **[Street\_name**]   * LocationType   [**Locationtype]** | Requested by Booking Consultant . |
|  | **13.2.2** | Request to Search Specific Location Details. | The Booking Consultant selects the Location to be viewed on the system. | When a specific Location is to be viewed on the system. | Booking Consultant (Location)   * Location name   [**Location\_name**]   * Province name   [**Province\_name**]   * City   [**City\_Name**]   * Suburb   [**Suburb\_name**]   * Street   **[Street\_name**]   * LocationType   [**Locationtype]** | Selected by Booking Consultant . |
| 13.13 Update Location | **13.3.1** | Request Update  Location Details. | Triggering input to start the updating Location process. | When the Location details need to change on the system. | Booking Consultant (Location) | Requested by Booking Consultant . |
|  | **13.3.2** | Update Location Details. | The Booking Consultant selects which Location to update details | When the Location details need to change on the system. | Booking Consultant (Location)   * Location name   [**Location\_name**]   * Province name   [**Province\_name**]   * City   [**City\_Name**]   * Suburb   [**Suburb\_name**]   * Street   **[Street\_name**]   * LocationType   [**Locationtype]** | Selected by Booking Consultant . |
|  | **13.3.3** | Updated Location Details. | Details of Location to be used for updating details of Location . | When the Location details need to change on the system. | * Location name   [**Location\_name**]   * Province name   [**Province\_name**]   * City   [**City\_Name**]   * Suburb   [**Suburb\_name**]   * Street   **[Street\_name**]   * LocationType   [**Locationtype]** | Entered by Booking Consultant . |
|  | 13.3.4 | Save Updated Location Details. | The Booking Consultant indicates selection to save details to be updated on the system. | When the Location details need to change on the system. | Booking Consultant(Location).   * Location name   [Location\_name]   * Province name   [Province\_name]   * City   [City\_Name]   * Suburb   [Suburb\_name]   * Street   [Street\_name]   * LocationType   [Locationtype] | Selected by Booking Consultant . |

| Use Case  Number | Flow  Line | Description | Purpose | When it will be  Used | Entities and Attributes | Logical  Layout |
| --- | --- | --- | --- | --- | --- | --- |
| 14.1 Add Zone | **14.1.1** | Request Add Zone Details | Triggering input to start the adding Zone  process. | When a new Zone is to be added on the  system. | Booking Consultant (Zone)   * Zone Id   **[Zone\_ID]**   * Zone name   **[Zone\_name]**   * Zone range   **[zone\_range]** | Requested by Booking Consultant . |
| **14.1.2** | Enter Zone Details | Details of new Zone to be used to add new Zone . | When a new Zone is to be added on the system. | * Zone Id   **[Zone\_ID]**   * Zone name   **[Zone\_name]**   * Zone range   **[zone\_range]** | Entered by Booking Consultant . |
| **14.1.3** | Save Zone Details | The Booking Consultant indicates selection to save details for new  Zone . | When a new Zone is to be added on the system. | Booking Consultant (Zone)   * Zone Id   **[Zone\_ID]**   * Zone name   **[Zone\_name]**   * Zone range   **[zone\_range]** | Selected by Booking Consultant . |
| **14.1.4** | Confirm Zone Details | The Booking Consultant indicates confirmation of  adding new Zone . | When the system prompts the Booking Consultant to confirm new  Zone details. | Booking Consultant (Zone ) | Selected by Booking Consultant . |
| **14.1 [ALT]** | Revoked Confirmation Details | The Booking Consultant revokes confirmation of adding new Zone . | When the system prompts the Booking Consultant to confirm new  Zone details. | Booking Consultant (Zone ) | Selected by Booking Consultant . |
| 14.2 Search Zone | **14.2.1** | Request Search Zone Details. | Triggering input to start the Zone Viewing process. | When a specific Zone is to be viewed on the  system. | Booking Consultant (Zone)   * Zone Type   [**Zone type]** | Requested by Booking Consultant . |
|  | **14.2.2** | Request to Search Specific Zone Details. | The Booking Consultant selects the Zone to be viewed on the system. | When a specific Zone is to be viewed on the system. | Booking Consultant (Zone)   * Zone Id   **[Zone\_ID]**   * Zone name   **[Zone\_name]**   * Zone range   **[zone\_range]** | Selected by Booking Consultant . |
| 13.3 Update Zone | **14.3.1** | Request Update  Zone Details. | Triggering input to start the updating Zone process. | When the Zone details need to change on the system. | Booking Consultant (Zone ) | Requested by Booking Consultant . |
|  | **14.3.2** | Update Zone Details. | The Booking Consultant selects which Zone to update details | When the Zone details need to change on the system. | Booking Consultant (Zone)   * Zone Id   **[Zone\_ID]**   * Zone name   **[Zone\_name]**   * Zone range   **[zone\_range]** | Selected by Booking Consultant . |
|  | **14.3.3** | Updated Zone Details. | Details of Zone to be used for updating details of Zone . | When the Zone details need to change on the system. | * Zone Id   **[Zone\_ID]**   * Zone name   **[Zone\_name]**   * Zone range   **[zone\_range]** | Entered by Booking Consultant . |
|  | 14.3.4 | Save Updated Zone Details. | The Booking Consultant indicates selection to save details to be updated on the system. | When the Zone details need to change on the system. | Booking Consultant (**Zone)**.   * Zone Id   **[Zone\_ID]**   * Zone name   **[Zone\_name]**   * Zone range   **[zone\_range]** | Selected by Booking Consultant . |
|  | **14.3.5** | Confirm Zone Details. | The Booking Consultant indicates confirmation of  updating Zone . | When the system prompts the Booking Consultant to confirm  updated details. | Zone Booking Consultant | Selected by Zone Booking Consultant . |
| **14.3 [ALT]** | Revoke Confirmation Details. | The Booking Consultant revokes confirmation of  updating Zone . | When the system prompts the Booking Consultant to confirm  updated details. | Zone Booking Consultant | Selected by Zone Booking Consultant . |
| 13.4 Remove Zone | **14.4.1** | Request Removal of Zone Details | Triggering input to start the Zone Removal process. | When a needs to be removed from  the system. | Zone Booking Consultant | Requested by Zone Booking Consultant . |
| **14.4.2** | Request Removal of Selected Zone . | The Booking Consultant selects which Zone to be removed from the  system. | When a Zone needs to be removed from the system. | Zone Booking Consultant   * Zone Id   **[Zone\_ID]**   * Zone name   **[Zone\_name]**   * Zone range   **[zone\_range]** | Selected by Zone Booking Consultant . |
| **14.4.3** | Confirmation to Remove Zone Details. | The Booking Consultant indicates confirmation of  removal of Zone . | When the system prompts the Booking Consultant to confirm  removal of Zone . | Zone Booking Consultant | Selected by Zone Booking Consultant . |
| **14.4 [ALT]** | Revoke Confirmation of Removal Details. | The Booking Consultant revokes confirmation of  removal of Zone . | When the system prompts the Booking Consultant to confirm  removal of Zone . | Zone Booking Consultant | Selected by Zone Booking Consultant . |

| **Use Case**  **Number** | **Flow**  **Line** | **Description** | **Purpose** | **When it will**  **be Used** | **Entities and Attributes** | **Logical Layout** |
| --- | --- | --- | --- | --- | --- | --- |
| **15.1.1Generate Invoice** | **15.1.1** | Request to Generate Invoice . | Triggering input to start the Invoice ing process. | When the Booking Consultant wants the  Invoice to be generated. | - | Requested by Booking Consultant. |
| **15.1.2** | Selected Booking . | Booking Consultant selects  the Booking for the Invoice to be based on. | When the Booking  for the Invoice needs to chosen. | Booking   * Booking Name | Selected by booking consultant |
| **15.1.3** | Select to Generate Invoice. | Booking Consultant selects the option to generate the Invoice. | When the Booking Consultant wants the Invoice to be  generated. | -Invoice   * Invoice \_ID * Booking\_Refference * Invoice \_PaymentStatus * Invoice\_date * Invoice\_Quote | Selected by Booking Consultant. |

## conclusion

This section outlined in detail the system interfaces and input data depicted in the context diagram. 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 as well as the logical layout.

|  |
| --- |
| 7. reports and other outputs |

## Introduction

The following section construes each and every single output generated by our system. A table is created describe the purpose of every output, when will 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.

## Outputs

| **UseCase** | **Description** | **Purpose** | **When Used** | **Entities and Attributes** | **Logical Layout** |
| --- | --- | --- | --- | --- | --- |
| 1.1.3 | Requests access level details | To trigger the process of adding an access level | Adding a new access level to the system |  | Selected by the employee |
| 1.1.2 [Alt] | Notification failed message | To give notification message to the employee | Adding a new access level to the system |  | Selected by the employee |
| 1.1.5 [Alt] | Notification failed verification message | To give notification message to the employee | Adding a new access level to the system |  | Selected by the employee |
| 1.1.7 | Notification success message | To give notification message to the employee | Adding a new access level to the system | AccessLevel\_ID | Selected by the employee |
| 1.2.3 | Display access levels | To all the employee to select an access level | When an edit is required to an existing access level | AccessLevel\_ID”  “AccessLevel\_Name”  “AccessLevel\_Description”  AccessLevel Entity | Selected by the employee |
| 1.2.8 [Alt] | Notification failed verification message | To give notification message to the employee | When an edit is required to an existing access level |  | Selected by the employee |
| 1.2.10 | Notification success message | To give notification message to the employee | When an edit is required to an existing access level | AccessLevel\_ID”  “ | Selected by the employee |
| 1.2.8 [Alt] | Notification failed verification message | To give notification message to the employee | When an edit is required to an existing access level |  | Selected by the employee |
| 1.3.3 | Request search details | To Know which details are to be searched for | When certain details are needed about an access level | “AccessLevel\_Name”  **AccessLevel Entity** | Selected by the employee |
| 1.3.7 | Display search results | So that the employee can see the details they were searching for | When certain details are needed about an access level | AccessLevel\_ID”  “AccessLevel\_Name”  “AccessLevel\_Description”  AccessLevel Entity | Selected by the employee |

| **UseCase** | **Description** | **Purpose** | **When Used** | **Entities and Attributes** | **Logical Layout** |
| --- | --- | --- | --- | --- | --- |
| 2.1.4 | Request employee details | To keep details about the new employee | Adding a new employee to the system | “EMPID”  “EMP\_Name”  “EMP\_Surname”  “AuditID”  “EMP\_TypeID”  “EMP\_EmailAddress”  “EMP\_Contact”  “EMP\_Password”  “EMP\_IDNum”  “TitleID”  “GenderID”  **Employee Entity** | Selected by the employee |
| 2.1.8 | Display success notification | To notify the employee of the success | Adding a new employee to the system | “EMPID”  “EMP\_Name”  **Employee Entity** | Selected by the employee |
| 2.1.2 [Alt} | Notification message of no authority | To notify the employee that they don’t have the authority | Adding a new employee to the system |  | Selected by the employee |
| 2.1.6 [Alt] | Notification of failure to verify message | To notify the employee that the details entered aren’t correct | Adding a new employee to the system |  | Selected by the employee |
| 2.2.5 | Display update employee details | To show the employee the details that can be updated | When an edit is required to an existing employee | “EMPID”  “EMP\_Name”  “EMP\_Surname”  “AuditID”  “EMP\_TypeID”  “EMP\_EmailAddress”  “EMP\_Contact”  “EMP\_Password”  “EMP\_IDNum”  “TitleID”  “GenderID”  **Employee Entity** | Selected by the employee |
| 2.2.10 | Display success message of updated employee | To notify the employee of the success | When an edit is required to an existing employee | “EMPID”  “EMP\_Name”  “EMP\_Surname”  **Employee Entity** | Selected by the employee |
| 2.2.2 [Alt] | Notification message of no authority | To notify the employee that they don’t have the authority | When an edit is required to an existing employee |  | Selected by the employee |
| 2.2.8 [Alt] | Notification of failure to verify message | To notify the employee that the details entered aren’t correct | When an edit is required to an existing employee |  | Selected by the employee |
| 2.3.3 | Request the employee search details | To Know which details are to be searched for | When certain details are needed about an employee | “EMP\_Name”  “EMP\_Surname”  “EMP\_TypeID”  **Employee entity** | Selected by the employee |
| 2.3.7 | Display search results | So that the employee can see the details they were searching for | When certain details are needed about an employee | “EMP\_Name”  “EMP\_Surname”  “EMP\_TypeID”  **Employee entity** | Selected by the employee |
| 2.3.2 [Alt] | Notification message of no authority | To notify the employee that they don’t have the authority | When an edit is required to an existing employee |  | Selected by the employee |
| 2.3.5 [Alt[ | Notification of failure to verify message | To notify the employee that the details entered aren’t correct | When an edit is required to an existing employee |  | Selected by the employee |
| 2.3.7 [Alt] | Notification of failure to find a matching employee | To notify the employee that the details of the search weren’t found | When an edit is required to an existing employee |  | Selected by the employee |

| **UseCase** | **Description** | **Purpose** | **When Used** | **Entities and Attributes** | **Logical Layout** |
| --- | --- | --- | --- | --- | --- |
| 3.1.3 | Request employee type details | To keep details about the new employee type | Adding a new employee type to the system | “EMP\_TypeID”  “EMP\_TypeName”  “EMP\_TypeDes”  **Employee type Entity** | Selected by the employee |
| 3.1.8 | Display success message | To let the employee that the employee type has been added | Adding a new employee type to the system | “EMP\_TypeID”  “EMP\_TypeName”  “EMP\_TypeDes”  **Employee type Entity** | Selected by the employee |
| 3.1.2 [Alt] | Notification message of no authority | To notify the employee that they don’t have the authority | Adding a new employee type to the system |  | Selected by the employee |
| 3.2.5 | Request new details to employee type | To update the employee type with the new details entered | When an edit is required to an existing employee type | “EMP\_TypeName”  “EMP\_TypeDes”  **Employee type Entity** | Selected by the employee |
| 3.2.10 | Display success message | To let the employee that the employee type has been updated | When an edit is required to an existing employee type | “EMP\_TypeName”  “EMP\_TypeDes”  **Employee type Entity** | Selected by the employee |
| 3.2.2 | Notification message of no authority | To notify the employee that they don’t have the authority | When certain details are needed about an employee type |  | Selected by the employee |
| 3.3.3 | Request employee type search details | To Know which details are to be searched for | When certain details are needed about an employee type | “EMP\_TypeName”  “EMP\_TypeDes”  **Employee type Entity** | Selected by the employee |
| 3.3.7 | Display search results | So that the employee can see the details they were searching for | When certain details are needed about an employee type | “EMP\_TypeID”  “EMP\_TypeName”  “EMP\_TypeDes”  **Employee type Entity** | Selected by the employee |
| 3.3.2 [Alt] | Notification message of no authority | To notify the employee that they don’t have the authority | When certain details are needed about an employee type |  | Selected by the employee |
| 3.4.4 | Request the employee type details | To know which employee type is to be removed | So that the employee type can no longer be used anymore | “EMP\_TypeID”  “EMP\_TypeName”  “EMP\_TypeDes”  **Employee type Entity** | Selected by the employee |
| 3.4.6 | Request confirmation of removal | To ensure that the correct employee type is disabled | So that the employee type can no longer be used anymore |  | Selected by the employee |
| 3.4.9 | Display success notification message | So that the employee can see the details they are removing | So that the employee type can no longer be used anymore | “EMP\_TypeID”  “EMP\_TypeName”  “EMP\_TypeDes”  **Employee type Entity** | Selected by the employee |
| 3.4.2 [Alt] | Notification message of no authority | To notify the employee that they don’t have the authority | So that the employee type can no longer be used anymore |  | Selected by the employee |
| 3.4.8 [Alt] | Notification message of failure removal | To notify the employee that removal has failed | So that the employee can know the results |  | Selected by the employee |

| **Use Case Number** | | **Flow Line** | **Description** | | | **Purpose** | **When it will be used** | **Entities and attributes** | **Logical Layout** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4. Driver | |  |  | | |  |  |  |  |
| 4.1 Search driver | | 4.1.2[ALT] | Display error message | | | This is to trigger the process that displays an error message to the booking consultant | When the character information entered is not valid | - | Information being displayed to booking consultant |
|  | | 4.1.6[ALT] | Alert message | | | This triggers the process that displays an alert message to the booking consultant | When the system cannot find any characters that match the input | - | An alert is being sent to the booking consultant |
|  | | 4.1 | Display driver details | | | This is the output which displays the details of a driver | When the search result returns a match | DRIVER   * Driver\_ID * Driver\_Name, * Driver\_Surname, * Driver\_LicenceNumber, * Driver\_LincenceType, * EMPID | The information of a driver is displayed to the booking consultant |
| 4.2 Check driver availability | | 4.2 | List of available drivers | | | This is the available drivers output that is displayed | When checking driver availability | **SLOT**   * Slot\_ID * Slot\_Date * Slot\_Time | The list of drivers that is displayed to the booking consultant |
|  | | 4.2.4[ALT] | Display message | | | This is the error message to error displayed to the booking consultant | When the driver list cannot be retrieved | - | An error message is displayed to the booking consultant |
| 4.3 Assign driver to trip | | 4.3 | Assignment confirmation | | | This is to display a message of successful assignment | When the system successfully assign a driver to a trip | - | Confirmation of assignment displayed to the booking consultant |
|  | | 4.3.1 [ALT] | Driver not available message | | | This displays a message to the booking consultant that a driver is not available | When the system does not return a list of the available drivers | - | A message is displayed to the booking consultant |
| 4.4 Outsource driver | | 4.4 | Outsource driver assignment message | | | This is to display a message of a successful outsource driver assignment | When the system successfully assigns an outsourced driver to a trip | - | Confirmation of assignment displayed to the booking consultant |
| 5. Vehicle | |  |  | | |  |  |  |  |
| 5.1 Add vehicle | | 5.1 | Added vehicle confirmation | | | This is the process that displays an a confirmation of a vehicle addition | When the system has added a new vehicle to the system | - | Alerts booking consultant of successful addition of vehicle |
|  | | 5.1.3 [ALT] | Display error message | | | This is to inform the booking consultant of an error in the input | When information provided fails the system verification | - | A message is displayed to the booking consultant |
| 5.2 Search vehicle | | 5.2 | Vehicle details | | | This is the output resulting from a vehicle being searched | When the system retrieves the information of a vehicle | **VEHICLE**   * Vehicle\_ID * Vehicle\_Model * VehicleMake\_ID * VehicleMaintenance\_ID * VehicleColour\_ID * VehicleLicencePlates | Vehicle information displayed to the booking consultant |
|  | | 5.2.3 [ALT] | Error in characters message | | | To inform the booking consultant of an error in the input | When the characters entered do not fit the system requirements | - | An error message is displayed to the booking consultant |
|  | | 5.2.5 [ALT] | Match not found message | | | Display a message to the booking consultant | When the system cannot find a match for entered characters | - | An error message is displayed to the booking consultant |
| 5.3 Confirm trip | | 5.3a | Confirmation message | | | This is the process that displays a confirmation message to the booking consultant | When the system has successfully sent trip confirmation details | - | Confirmation message displayed to booking consultant |
|  | | 5.3b | Passenger trip details | | | This is the output that is sent to the passenger | When the system retrieves trip information | **SLOT**   * BookingReference\_ID * DeparturePoint * DestinationPoint * Slot\_Time   **PASSENGER**   * Passenger\_ID * Passenger\_Name * Passenger\_Contact | Trip details sent to the passenger |
|  | | 5.3c | Driver Trip Details | | | This is the output that is sent to the driver | When the system retrieves trip information and accumulates the driver trips of the day | **SLOT**   * BookingReference\_ID * DeparturePoint * DestinationPoint * Slot\_Time   **DRIVER**   * Driver\_ID * Driver\_Name * Driver\_Contact | Trip details sent to the driver |
| 5.4 Check vehicle availability | | 5.4 | List of available vehicles information | | | This is the results from the process that searches for available drivers | When the system retrieves a list of the vehicle that are available | **BOOKING**   * BookingReference\_ID * Passenger\_ID * Driver\_ID * DeparturePoint * DestinationPoint * Time   **VEHICLE GROUP**   * Number\_of\_Passengers   **SLOT**   * Slot\_Date * Slot\_Time | List of available vehicles is displayed to the booking consultant |
|  | | 5.4.3 [ALT] | Display no vehicles available message | | | Inform the booking consultant that there are no cars available | When the system cannot find available vehicles on the system | **-** |  |
| 5.5 Assign vehicle to trip | | 5.5 | Vehicle assignment confirmation | | | This process displays the information of a vehicle being assigned | When the system has assigned a driver to the trip | **SLOT**   * Slot\_ID * Slot\_Date * Slot\_Time | Cofirmation of assignment message displayed to the booking consultant |
| 5.6 Outsource vehicle | | 5.6 | Outsource vehicle assignment confirmation | | | This is the output of a process that assigns an outsourced vehicle to a trip | When the system successfully add the outsource vehicle information | **SLOT**   * Slot\_ID * Slot\_Date * Slot\_Time   **OUTSOURCE\_VEHICLE**   * OutsourceVehicle\_ID * OutSource\_Vehicle\_Model * OutSource\_Make | A message of confirmation is displayed to the booking consultant |
| 6. Vehicle group | |  |  | | |  |  |  |  |
| 6.1 Create vehicle group | | 6.1 | New vehicle confirmation message | | | This is the result of a vehicle group being added on the system | When they system adds a new vehicle group | **VEHICLE GROUP**   * Vehicle group ID, * VehicleGroup\_Name * VehicleGroup\_Description | A confirmation message is displayed to the booking consultant |
|  | | 6.1.3 [ALT] | Display error message | | | To inform the booking consultant of error in input | When there is an error in input | - | A message is displayed to the booking consultant |
| 6.2 Search vehicle group | | 6.2 | Vehicle group information | | | This is the process that displays the vehicle group | When a match as been found for the search criteria by the system | **VEHICLE GROUP**   * Vehicle group ID, * VehicleGroup\_Name * VehicleGroup\_Description | Vehicle group information is displayed to the booking consultant |
|  | | 6.2.2[ALT] | Error in character input information | | | This displays a message to the booking consultant that there is an error in the input | When the input does not meet system criteria | **-** | A message is displayed to the booking consultant |
|  | | 6.2.5[ALT] | Match not found information | | | This displays a message to the booking consultant that a match has not been found | When the system cannot find a match to the characters entered | **-** | A match not found message is displayed to the booking consultant |
| 6.3 Update vehicle group | | 6.3 | Updated details message | | | This is the process that displays a message confirming update | When the system has updated the vehicle group information | **VEHICLE GROUP**   * VehicleGroup\_Name * VehicleGroup\_Description | Confirmation message is sent to the booking consultant |
|  | | 6.3.5[ALT] | Input error message | | | This displays a message to the booking consultant about an error in the input | When the input does not meet the system requirements | - | A message is displayed to the booking consultant |
| Use case | Description | | Purpose | When will it be used | | Entities and Attributes | | Output | |
| 7.1 Add Client | [ALT] Check if the client details are in the correct format | | Inform the Booking consultant that data is incorrect format | When the system validates if the input is in the correct  format | |  | | Message to notify that incorrect input was received in the input fields | |
| Confirmation of Successfully adding a new client | | Inform the Booking consultant that Client is successfully added | Adding a new client to the Client table | | Client table  Client\_Name  Client\_Email  Client\_Tel  Client\_Reference  Client\_Address  Client\_Type | | Notification on the system | |
| 7.2 Search Client | list of all available clients in the Client table | | Booking consultant could select the specific client | When searching for a client | | Entity : Client  Attributes: Client\_Name  Client\_Reference  Client\_PhoneNumber  Client\_Email | | Displayed on the screen | |
|  | selected Client | | To view the details of the client that is being searched for | When the specific client is found | | Entity : Client  Attributes:  Client\_Name  Client\_Reference  Client\_PhoneNumber  Client\_Email  Client\_Address  Client\_type | | Displayed on the screen | |
| [ALT] Invalid format details | | Booking consultant being informed of incorrect input format | When client’s reference is in an incorrect format | |  | | Message to notify that incorrect input was received in the input fields | |
| [ALT] Client not found | | Booking consultant being notified that the Client could not be retrieved in the Client table | When the system finds no match for the searched client | |  | | Message that the client could not be retrieved | |
| 7.3 Update Client | [ALT]Client to update not found | | When the client the Booking consultant wants to update is not found | When the Booking consultant searches for the client to update | |  | | Message that the client could not be found | |
| [ALT] Incorrect input format | | When the Booking consultant is informed of the incorrect input | When the input is not in a correct format | |  | | Message to notify that there is incorrect input | |
| Updated Client Details | | When the Booking consultant is notified of the updating of Client details | Client details has been updated | |  | | Notification on the system | |
| 8.1 Add Client Type | Confirmation of successfully adding a new Client Type | | When the booking consultant is notified that the Client type is added | Client type is added to the system | | Entity :Client\_Type  Attributes  ClientType\_Name  ClientType\_Description | | Notification on the system | |
| Duplicate Client types | | When the booking consultant is notified that the client type exists | Adding a similar client type | | Entity:Client\_Type  Attributes  ClientType\_Name  ClientType\_Description | | Message to notify that the clientType already exists | |
| Invalid input format | | When the booking consultant is notified of incorrect input | When the input is not in a correct format | |  | | Message to notify that there is incorrect input | |
| 8.2 Search ClientType | Client type retrieved from client table | | When the Booking consultant is notified with the client type details | When the client type details are retrieved from the Client table | | Entity : Client\_Type  Attributes  ClientType\_Name  ClientType\_Description | | Displayed on the screen | |
| Client type not retrieved | | When the Booking consultant is notified that the Client type not retrieved | The client type name does not match any Client type | |  | | Message to notify that the client type could not be retrieved | |
| Invalid Input format | | When the booking consultant is notified of incorrect input | When the input is not in a correct format | |  | | Message to notify that there is incorrect input | |
| 8.3 Update ClientType | [ALT]ClientType to update not found | | When the clientType the Booking consultant wants to update is not found | When the Booking consultant searches for the clientType to update | |  | | Message to notify that the client type to update could not be found | |
| [ALT] Incorrect input format | | When the Booking consultant is informed of the incorrect input | When the input is not in a correct format | |  | | Message to notify that there is incorrect input | |
| Updated ClientType Details | | When the Booking consultant is notified of the updating of ClientType details | ClientType details has been updated | | Entity : Client\_Type  Attributes  ClientType\_Name  ClientType\_Description | | Notificaiton on the system | |

| Use case | Description | Purpose | When will it be used | Entities and Attributes | Output |
| --- | --- | --- | --- | --- | --- |
| 7.1 Add Client | [ALT] Check if the client details are in the correct format | Inform the Booking consultant that data is incorrect format | When the system validates if the input is in the correct  format |  | Message to notify that incorrect input was received in the input fields |
| Confirmation of Successfully adding a new client | Inform the Booking consultant that Client is successfully added | Adding a new client to the Client table | Client table  Client\_Name  Client\_Email  Client\_Tel  Client\_Reference  Client\_Address  Client\_Type | Notification on the system |
| 7.2 Search Client | list of all available clients in the Client table | Booking consultant could select the specific client | When searching for a client | Entity : Client  Attributes: Client\_Name  Client\_Reference  Client\_PhoneNumber  Client\_Email | Displayed on the screen |
| selected Client | To view the details of the client that is being searched for | When the specific client is found | Entity : Client  Attributes:  Client\_Name  Client\_Reference  Client\_PhoneNumber  Client\_Email  Client\_Address  Client\_type | Displayed on the screen |
| [ALT] Invalid format details | Booking consultant being informed of incorrect input format | When client’s reference is in an incorrect format |  | Message to notify that incorrect input was received in the input fields |
| [ALT] Client not found | Booking consultant being notified that the Client could not be retrieved in the Client table | When the system finds no match for the searched client |  | Message that the client could not be retrieved |
| 7.3 Update Client | [ALT]Client to update not found | When the client the Booking consultant wants to update is not found | When the Booking consultant searches for the client to update |  | Message that the client could not be found |
| [ALT] Incorrect input format | When the Booking consultant is informed of the incorrect input | When the input is not in a correct format |  | Message to notify that there is incorrect input |
| Updated Client Details | When the Booking consultant is notified of the updating of Client details | Client details has been updated |  | Notification on the system |
| 8.1 Add Client Type | Confirmation of successfully adding a new Client Type | When the booking consultant is notified that the Client type is added | Client type is added to the system | Entity :Client\_Type  Attributes  ClientType\_Name  ClientType\_Description | Notification on the system |
| Duplicate Client types | When the booking consultant is notified that the client type exists | Adding a similar client type | Entity:Client\_Type  Attributes  ClientType\_Name  ClientType\_Description | Message to notify that the clientType already exists |
| Invalid input format | When the booking consultant is notified of incorrect input | When the input is not in a correct format |  | Message to notify that there is incorrect input |
| 8.2 Search ClientType | Client type retrieved from client table | When the Booking consultant is notified with the client type details | When the client type details are retrieved from the Client table | Entity : Client\_Type  Attributes  ClientType\_Name  ClientType\_Description | Displayed on the screen |
| Client type not retrieved | When the Booking consultant is notified that the Client type not retrieved | The client type name does not match any Client type |  | Message to notify that the client type could not be retrieved |
| Invalid Input format | When the booking consultant is notified of incorrect input | When the input is not in a correct format |  | Message to notify that there is incorrect input |
| 8.3 Update ClientType | [ALT]ClientType to update not found | When the clientType the Booking consultant wants to update is not found | When the Booking consultant searches for the clientType to update |  | Message to notify that the client type to update could not be found |
| [ALT] Incorrect input format | When the Booking consultant is informed of the incorrect input | When the input is not in a correct format |  | Message to notify that there is incorrect input |
| Updated ClientType Details | When the Booking consultant is notified of the updating of ClientType details | ClientType details has been updated | Entity : Client\_Type  Attributes  ClientType\_Name  ClientType\_Description | Notificaiton on the system |

| Use case | Description | | Purpose | When will it be used | Entities and Attributes | | Logical Layout |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 9.1 Create Booking | Generated Booking Reference | Notify the Booking consultant of the generated Booking reference | | When a booking is made successfully | Entity: Booking\_Trip  Attributes  Booking\_Reference | Notification on the system | |
| [ALT] Incorrect input format | When the Booking consultant is informed of the incorrect input | | When the input is not in a correct format |  | Message to notify the Booking consultant that incorrect input | |
| [ALT] Duplicate Booking | When the Booking consultant is notified that the booking already exists | | Adding a Booking with the same details |  | Message to notify the Booking consultant that the Booking already exists | |
| Generated Trip Quote | When the Booking consultant wants to retrieve the estimated cost of the Trip | | When retrieving the total Cost of the trip |  | Message to Notify of the Cost of the Trip | |
|  |  | |  |  |  | |
| 9.3 Update Booking | [ALT]Booking to update not found | When the Booking to be updated is not found | | When the Booking consultant searches for the Booking to update | Booking\_Trip | Message to notify that the booking doesn’t exists | |
| [ALT] Incorrect input format | When the Booking consultant is informed of the incorrect input | | When the input is not in a correct format |  | Message to notify the Booking consultant that incorrect input | |
| Updated Booking Details | When the Booking consultant is notified of the updating of Booking details | | Booking details has been updated |  | Message to notify that the booking is successful | |
| 9.2 Search Booking | list of all available Bookings in the Booking\_Trip table | Booking consultant could select the specific Booking | | When searching for a Booking | Booking\_Trip | System displays a list of the Available Bookings | |
| selected Booking | To view the details of the Booking that is being searched for | | When the specific Booking is found | Booking\_Trip  Passenger  Invoice | Displayed on the screen | |
| [ALT] Invalid format details | Booking consultant being informed of incorrect input format | | When client’s reference is in an incorrect format |  | Message to notify the Booking consultant that there is incorrect input | |
| Booking Not Found | When the booking searched for is not found | | Booking reference does not match any Booking |  | Message to notify that the booking is not found | |
| 9.4 Cancel Booking | [ALT] Booking not found | When the booking searched for is not retrieved | | The Booking reference does not match any Booking in the Booking\_Trip | Entity: Booking\_Trip  Attributes  Booking\_Reference | Message to notify that the booking is not found | |
|  | Charge on the Payment status | When the trip is already due for dispatch | | The Booking consultant canceling trip |  | Message that 50% will be charged due to trip already due for dispatch | |
| Successful Cancellation of Booking | When the Booking cancellation succeeded | | When the Booking is removed from the Booking table |  | Notification on the system | |
|  |  | |  |  |  | |
| 9.5 Confirm Booking | [ALT] The booking could not be retrieved | When searching for the booking to confirm | | The booking reference does not match any booking in the Booking\_Trip |  | Notification on the system | |
|  | Available Vehicles | Inform the Booking consultant of the available vehicles | | When checking which vehicle is available for a trip |  | Notification on the system | |
|  | Available Drivers | Inform the Booking consultant of the available Drivers | | When checking which Driver is available for a trip |  | Notification on the system | |
|  | Booking Successfully Confirmed | To inform the booking consultant that the Booking is successfully confirmed | | When the Booking consultant finalizes a trip |  | Notification on the system | |

| **Use Case**  **Number** | Flow  Line | | Description | | Purpose | | When it will  be Produced | | Entities and Attributes | | Logical Layout |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **13.1 Add Location** | | **13.1 [ALT]** | | Failed Validation of Location Details. | | To inform the Booking Consultant that details for Location has been entered  incorrectly. | | When the Booking Consultant entered details for Location in the wrong format. | | (Location)   * Location name   [**Location\_name**]   * Province name   [**Province\_name**]   * City   [**City\_Name**]   * Suburb   [**Suburb\_name**]   * Street   **[Street\_name**]   * LocationType   [**Locationtype]** | Error message displayed to Booking Consultant . |
|  | | **13.1 [ALT]** | | Duplicate Location Detected Details. | | To inform the Booking Consultant that duplicate Location has been  detected on the system. | | When the Location being added has the details as existing Location on the system. | | (Location)   * Location name   [**Location\_name**]   * Province name   [**Province\_name**]   * City   [**City\_Name**]   * Suburb   [**Suburb\_name**]   * Street   **[Street\_name**] | Notification displayed to Booking Consultant . |
|  | **13.1.5** | | Notification of Successfully added Location information. | | To inform the Booking Consultant of successful addition of Location to  system. | | When the Booking Consultant has successfully added Location details to the  system. | | (Location)   * Location name   [**Location\_name**]   * Province name   [**Province\_name**]   * City   [**City\_Name**]   * Suburb   [**Suburb\_name**]   * Street   **[Street\_name**] | | Notification displayed to the Booking Consultant . |
| **13.2 Search Location** | **13.2.3** | | Selected Location with Relevant Location Land and Location Land Field information. | | To display all the relevant information about a Location with its lands  and fields. | | When the User has requested to view Location s on the system. | | (Location)   * Location name   [**Location\_name**]   * Province name   [**Province\_name**]   * City   [**City\_Name**]   * Suburb   [**Suburb\_name**]   * Street   **[Street\_name**] | | Information displayed on the screen. |
| **13.3 Update Location** | **13.3 [ALT]** | | Failed Validation of Location Details. | | To inform the Booking Consultant that details have been  entered incorrectly. | | When the Booking Consultant has entered details in the wrong format. | | - | | Error message displayed to Booking Consultant . |
| 13.3.6 | | Notification of Successful Update information. | | To inform the successful update of  Location details on the system. | | When the Booking Consultant has successfully  updated details of Location . | | (Location)   * Location name   [Location\_name]   * Province name   [Province\_name]   * City   [City\_Name]   * Suburb   [Suburb\_name]   * Street   [Street\_name] | | Notification message displayed to Booking Consultant . |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Use Case  Number | Flow  Line | Description | | | | Purpose | When it will  be Produced | | | Entities and Attributes | | Logical Layout | |
| 14.1 Add Zone | **14.1 [ALT]** | Failed Validation of Zone Details. | | | | To inform the Booking Consultant that details for Zone has been entered  incorrectly. | When the Booking Consultant entered details for Zone in the wrong format. | | | * Zone Id   **[Zone\_ID]**   * Zone name   **[Zone\_name]**   * Zone range   **[zone\_range]** | | Error message displayed to Booking Consultant . | |
|  | **14.1.5** | | Notification of Successfully added Zone information. | | To inform the Booking Consultant of successful addition of Zone to  system. | | | When the Booking Consultant has successfully added Zone details to the  system. | * Zone Id   **[Zone\_ID]**   * Zone name   **[Zone\_name]**   * Zone range   **[zone\_range]** | | Notification displayed to the Booking Consultant . | |
| 14.2 Search Zone | **14.2.3** | | | Selected Zone with Relevant Zone Range and Zone information. | To display all the relevant information about a Zone with Range | | | When the User has requested to view Zone s on the system. | * Zone Id   **[Zone\_ID]**   * Zone name   **[Zone\_name]**   * Zone range   **[zone\_range]**  [**Suburb\_name**] | | Information displayed on the screen. | |
| 14.3 Update Zone | 14.3 [ALT] | | | Failed Validation of Zone Details. | To inform the Booking Consultant that details have been  entered incorrectly. | | | When the Booking Consultant has entered details in the wrong format. | - | | Error message displayed to Booking Consultant . | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case | Flow  Line | Description | Purpose | When it will  be Produced | Entities and Attributes | Logical Layout |
| 14.4 Remove Zone | **14.4 [ALT]** | There is a Trip linked to Zone . | To inform Booking Consultant that the selected Zone is linked to a  Trip | When the system detects Zone is linked to a Trip and cannot proceed  to remove. | * Zone Id   **[Zone\_ID]**   * Zone name   **[Zone\_name]**   * Zone range   **[zone\_range]** | Notification message displayed to User. |
| 14.4.4 | Notification of Successfully Removed Zone information. | To inform the Booking Consultant that Zone has successfully been  removed. | When the Booking Consultant has selected to remove a specific Zone . | * Zone Id   [Zone\_ID]   * Zone name   [Zone\_name]   * Zone range   [zone\_range] | Notification message displayed to Booking Consultant . |

| **Use Case**  **Number** | **Flow**  **Line** | **Description** | **Purpose** | **When it will**  **be Produced** | **Entities and Attributes** | **Logical Layout** |
| --- | --- | --- | --- | --- | --- | --- |
| **15.5Generate Invoice .** | **15.5[ALT]** | Booking Has Not Been Selected. | To inform the Booking Consultant that Booking details have not been specified. | When the system tries to generate Invoice but detects that Booking details have not been  chosen. | - | Error message displayed to Booking Consultant . |
| **15.5[ALT]** | Unable to Access Database to Retrieve Data | To inform the Booking Consultant that Invoice cannot be generated because data required cannot be  retrieved. | When the system tries to retrieve data required for report, it encounters problems. | - | Error Notification displayed to Booking Consultant . |
|  | **15.6** | Generated  and End Dates  Failed  Validation  Checks | To generate an Invoice | When the  Consultant requests  to generate a  Invoice | Invoice   * Invoice \_ID * Booking\_Refference * Invoice \_PaymentStatus * Invoice\_date * Invoice\_Quote | Invoice  (List ): List  of all completed trips on  An invoice with all its  associated  details. |

## conclusion

This section assisted us in grasping all the system outputs in more detail. It helped us to understand the outputs purpose, when it gets produced, entities and attributes involved as well as the logical layout all contained in a tabular document.

|  |
| --- |
| 8. validation |

## Introduction

In this section, the validation of each functional specification, process and entities is provided to check if they correspond to the business requirements.

## validation

| **Subsystem** | **Requirement** | **Use Case** | **Process(DFD)** | **Entities(ERD)** |
| --- | --- | --- | --- | --- |
| 4. Driver | 4.1 Search driver | 4.1 Search driver | 4.1.1  4.1.2  4.1.3  4.1.4  4.1.5  4.1.6  4.1.7 | -  -  -  Driver  -  Driver  - |
|  | 4.2 Check driver availability | 4.2 Check driver availability | 4.2.1  4.2.2  4.2.3  4.2.4 | -  Slot  -  - |
|  | 4.3 Assign driver to trip | 4.3 Assign driver to trip | 4.3.1  4.3.2  4.3.3  4.3.4  4.3.5 | -  -  -  Slot  - |
|  | 4.4 Outsource driver | 4.4 Outsource driver | 4.5.1  4.5.2  4.5.3  4.5.4 | Outsourced\_Driver  -  Slot  - |
| 5. Vehicle | 5.1 Add vehicle | 5.1 Add vehicle | 5.1.1  5.1.2  5.1.3  5.1.4  5.1.5  5.1.6 | -  -  -  Vehicle, VehicleMake, VehicleMaintenance  Vehicle, VehicleMake, VehicleMaintenance  - |
|  | 5.2 Search vehicle | 5.2 Search vehicle | 5.2.1  5.2.2  5.2.3  5.2.4  5.2.5  5.2.6  5.2.7 | -  -  -  Vehicle  -  Vehicle  - |
|  | 5.3 Confirm trip | 5.3 Confirm trip | 5.3.1  5.3.2  5.3.3  5.3.4  5.3.5  5.3.6  5.3.7 | -  Slot  Passenger, Driver  -  -  -  - |
|  | 5.4 Check vehicle availability | 5.4 Check vehicle availability | 5.4.1  5.4.2  5.4.3  5.4.4 | Booking  Vehicle\_Group, Slot  -  - |
|  | 5.5 Assign vehicle to trip | 5.5 Assign vehicle to trip | 5.5.1  5.5.2  5.5.3  5.5.4 | -  -  Slot  - |
|  | 5.6 Outsource vehicle | 5.6 Outsource vehicle | 5.6.1  5.6.2  5.6.3  5.6.4 | Outsource\_Vehicle  -  Slot  - |
| 6. Vehicle Group | 6.1 Create vehicle group | 6.1 Create vehicle group | 6.1.1  6.1.2  6.1.3  6.1.4  6.1.5  6.1.6 | -  -  -  Vehicle\_Group  Vehicle\_Group  - |
|  | 6.2 Search vehicle group | 6.2 Search vehicle group | 6.2.1  6.2.2  6.2.3  6.2.4  6.2.5  6.2.6  6.2.7 | -  -  -  Vehicle\_Group  -  -  Vehicle\_Group |
|  | 6.3 Update vehicle group | 6.3 Update vehicle group | 6.3.1  6.3.2  6.3.3  6.3.4  6.3.5  6.3.6  6.3.7 | -  -  -  -  -  Vehicle\_Group  - |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SubSystem | Requirement | User Case | Process(DFD) | Entities(ERD) |
| CLIENT | 7.1 Add Client | 7.1 Add Client | 7.1.1 Capture new Client Details |  |
|  |  | 7.1.2 Validate the Client Details |  |
|  |  | 7.1.3 Validate for duplicates | Client |
|  |  | 7.1.4 Save the new Client | Client |
|  |  | 7.1.5 Generate Confirmation Message |  |
|  |  |  |  |
| 7.2 Search Client |  | 7.2.1 Display All Clients | Client |
|  |  | 7.2.2 Capture and validate Client Reference |  |
|  |  | 7.2.3 Narrow the Client details |  |
|  |  | 7.2.4 Display Selected Client Details | Client |
|  |  |  |  |
| 7.3 Update Client |  | 7.3.1 Retrieve Client to Update | Client |
|  |  | 7.3.2 Capture Client Details to Update |  |
|  |  | 7.3.3 Validate Updated Client details |  |
|  |  | 7.3.4 Save the update client details | Client |
|  |  | 7.3.5 Generate Confirmation of Updating Client |  |
|  |  |  |  |
| CLIENT\_TYPE | 8.1 Add Client\_Type |  | 8.1.1 Capture Client Type Details |  |
|  |  | 8.1.2 Validates the Client Type Details |  |
|  |  | 8.1.3 Validates Existing Client types | Client\_Type |
|  |  | 8.1.4 Save the Client Type | Client\_Type |
|  |  | 8.1.5 Generate Confirmation Message |  |
|  |  |  |  |
| 8.2 Search Client\_Type |  | 8.2.1 Display All Clients\_Type | Client\_Type |
|  |  |  | 8.2.2 Capture and validate Client Type name |  |
|  |  | 8.2.3 Narrow the Client Type details |  |
|  |  |  | 8.2.4 Display Selected Client Type Details | Client\_Type |
|  |  |  |  |  |
|  |  |  |  |
| 8.3 Update Client Type |  | 8.3.1 Retrieve Client Type to Update | Client\_Type |
|  |  | 8.3.2 Capture Client type Details to Update |  |
|  |  | 8.3.3 Validate New Client type details |  |
|  |  | 8.3.4 Save The update Client Type details | Client\_Type |
|  |  | 8.3.5 Generate Confirmation of Updating Client Type Details |  |
|  |  |  |  |
| 9 BOOKING | 9.1 Create Booking |  | 9.1.1 Request Client Name and validate the format |  |
|  |  | 9.1.2 Validate Client Existence | Client |
|  |  | 9.1.3 Confirm Client Details and save |  |
|  |  | 9.1.4 Capture Pickup and DropOff Details |  |
|  |  | 9.1.5 Validates Input Format |  |
|  |  | 9.1.6 Check for Booking Duplicates | Booking\_Trip |
|  |  | 9.1.7 Generate Quote and Booking Reference |  |
|  |  | 9.1.8 Capture Passenger Details |  |
|  |  | 9.1.9 Validates input Format |  |
|  |  | 9.1.10 Checks Duplicate Passenger | Passenger |
|  |  | 9.1.11 Saves Passenger | Passenger |
|  |  | 9.1.12 Capture Pickup Instructions |  |
|  |  | 9.1.13 Communicates Booking Details |  |
|  |  | 9.1.14 Generate Invoice Number | Invoice |
| BOOKING |  |  | 9.1.15 Save Booking Details and save Invoice | Booking\_Trip  Invoice |
|  |  | 9.1.16 Confirm Booking and Notify about Booking Reference |  |
|  |  |  |  |
| 9.2 Search Booking |  | 9.2.1 Display All Bookings | Booking\_Trip |
|  |  | 9.2.2 Capture and Validated Booking Reference |  |
|  |  | 9.2.3 Narrow the bookings displayed |  |
|  |  | 9.2.4 Display Selected Booking Details | Booking\_Trip  Passenger  Invoice |
|  |  |  |  |
| 9.3 Update Booking |  | 9.3.1 Retrieve a Booking to be updated | Booking\_Trip  Passenger  Invoice |
|  |  | 9.3.2 Capture updated Booking details |  |
|  |  | 9.3.3 Validate Updated Booking Details |  |
|  |  | 9.3.4 Save the Updated Booking Details | Booking\_Trip  Passenger  Invoice |
|  |  | 9.3.5 Generate Confirmation of Updating |  |
|  |  |  |  |
|  |  |  |  |
| BOOKING | Cancel Booking | 9.4 Cancel Booking | 9.4.1 Retrieve Booking to cancel | Booking\_Reference |
|  |  | 9.4.2 Check if Booking is due for Dispatch |  |
|  |  | 9.4.3 Capture Reasons for Cancellation |  |
|  |  | 9.4.4 Update Vehicle and Driver Availability | Slot |
|  |  | 9.4.5 Update Schedule |  |
|  |  | 9.4.6 Update Payment\_Status | Invoice |
|  |  | 9.4.7 Update Booking Status | Booking\_Trip |
|  |  | 9.4.8 Generate Confirmation of Booking Cancellation |  |
|  |  |  |  |
| Booking | Confirm Booking | 9.5 Confirm Booking | 9.5.1 Retrieve Booking to be Confirmed | Booking\_Trip |
|  |  | 9.5.2 Capture Assigned Vehicle Details | Slot  Vehicle\_Group |
|  |  | 9.5.3 Capture Assigned Driver Details | Slot  Driver |
|  |  | 9.5.4 Assign Driver and Vehicle | Slot |
|  |  | 9.5.5 Create Schedule and Update Booking Status | Booking\_Trip |

| **Subsystem** | **Requirement** | **Use Case** | **Process(DFD)** | **Entities(ERD)** |
| --- | --- | --- | --- | --- |
| 10 Schedule | 10.1 Create Schedule | Create Schedule | 10.1.1 Read Booking details | N/A |
| 10.1.2 Validate Format of the Booking Details | **Schedule Table**   * **BookingReference** * **Client\_ID** * **Driver\_ID** * **Date** * **Time** * **Destination Time** * **DestinationLocation** * **TripDuration (Which is calculated using Time and Destination Time)** |
| 10.1.3 Retrieve the last Booking Unique Number | N/A |
| 10.1.4 Store Booking Details | **Schedule Table**   * **BookingReference** * **Client\_ID** * **Driver\_ID** * **Date** * **Time** * **Destination Time** * **DestinationLocation** * **TripDuration (Which is calculated using Time and Destination Time)** |
| 10.1.5 Notify Booking Consultant with confirmation Message | N/A |
| ALT 10.1.2 Display Invalid Error Message | N/A |
| 10.2 Update schedule | Update schedule | 10.2.1Request to Update Scheduled booking | N/A |
| 10.2.2 Display Details of the booking to be updated | N/A |
| 10.2.3 Read Updated Booking Details | N/A |
| 10.2.4 Validate Format of the updated booking details | N/A |
| 10.2.5Stored updated Booking details | N/A |
| 10.2.6 Display updated booking details | N/A |
| 10.2.7 Notify of successful Update | N/A |
| ALT 10.2.4 Display Error message for incorrect format | N/A |
| 10.3 View schedule | View schedule | 10.3.1 Request to view schedule | N/A |
| 10.3.2 Read user entered Search criteria | **Schedule** table:   * **BookingRefernce** * **Client\_ID** * **Driver\_ID** * **Date** * **Time** * **DestinationLocation** * **TripDuration** |
| 10.3.3 Perform schedule search | N/A |
| 10.3.4 Display list of all bookings from the chosen date | **Schedule** table:   * **BookingRefernce** * **Client\_ID** * **Driver\_ID** * **Date** * **Time** * **DestinationLocation** * **TripDuration** |
| 10.3.5 Display chosen booking details | **Schedule** table:   * **BookingRefernce** * **Client\_ID** * **Driver\_ID** * **Date** * **Time** * **DestinationLocation** * **TripDuration** |
| 10.3.6 Read User Selected Search Result | N/A |
| 10.3.7 Display User Selected Search Result Details | N/A |
| ALT 10.3.3 Display Schedule Search Criteria Error Message | N/A |
| 11 Reporting | 11.1 Booking Report | Booking Report | 11.1.1 Display schedule generation | N/A |
| 11.1.2 Requests booking search criteria | N/A |
| 11.1.3 Reads entered report criteria details | Schedule Table:   * **Vehicle\_ID** * **Client\_ID** * **Driver\_ID** * **Booking\_Date** * **Pickup-Location** * **DropOff-location** * **Date** * **Time** |
| 11.1.4 Generate report of the entered schedule details | Schedule Table:   * **Vehicle\_ID** * **Client\_ID** * **Driver\_ID** * **Booking\_Date** * **Pickup-Location** * **DropOff-location** * **Date** * **Time** |
| 11.1.5 Display complete report | Schedule Table:   * **Vehicle\_ID** * **Client\_ID** * **Driver\_ID** * **Booking\_Date** * **Pickup-Location** * **DropOff-location** * **Date** * **Time** |
| ALT 11.1.4 Displays invalid schedule not found error message | N/A |
| 11.2 Driver-Trip Report | Driver-Trip Report | 11.2.1 Display schedule generation | N/A |
| 11.2.2 Requests booking search criteria | N/A |
| 11.2.3 Reads entered report criteria details | Schedule Table:   * **Vehicle\_ID** * **Client\_ID** * **Driver\_ID** * **Booking\_Date** * **Pickup-Location** * **DropOff-location** * **Date** * **Time** |
| 11.2.4 Generate report of the entered schedule details | Schedule Table:   * **Vehicle\_ID** * **Client\_ID** * **Driver\_ID** * **Booking\_Date** * **Pickup-Location** * **DropOff-location** * **Date** * **Time** |
| 11.2.5 Display complete report | Schedule Table:   * **Vehicle\_ID** * **Client\_ID** * **Driver\_ID** * **Booking\_Date** * **Pickup-Location** * **DropOff-location** * **Date** * **Time** |
| ALT 11.2.4 Displays invalid schedule not found error message | N/A |
| 12 Vehicle Maintenance | 12.1 Schedule Vehicle Maintenance schedule | Schedule Vehicle Maintenance schedule | 12.1.1 Request Maintenance Details | N/A |
| 12.1.2 Read Maintenance Details | **Vehicle Table**   * Vehicle\_ID * Vehicle\_Description * Vehicle\_Name * Vehicle\_LicseneceNumber * Duration * DateTime |
| 12.1.3 Validate Format of Maintenance Details | N/A |
| 12.1.4 Retrieve Last Maintenance unique number | **Vehicle maintenance table**   * **Vehicle\_MaintenanceID** |
| 12.1.5 Store Maintenance Details | **Vehicle maintenance table**   * Vehicle\_ID * Vehicle\_Description * Vehicle\_Name * Vehicle\_LicseneceNumber * Duration * DateTime |
| 12.1.6 Notify Operational Manger of scheduling | N/A |
| 12.2 Update Vehicle Maintenance schedule | Update Vehicle Maintenance schedule | 12.2.1 Request to Update schedule | N/A |
| 12.2.2 Display Details of schedule list that can be Updated | **Vehicle maintenance table**   * Vehicle\_ID * Vehicle\_Description * Vehicle\_Name * Vehicle\_LicseneceNumber * Duration * DateTime |
| 12.2.3 Read updated schedule details | N/A |
| 12.2.4 Validate format of the updated schedule details | N/A |
| 12.2.5 Store updated schedule details | **Vehicle maintenance table**   * Vehicle\_ID * Vehicle\_Description * Vehicle\_Name * Vehicle\_LicseneceNumber * Duration * DateTime |
| 12.2.6 Display updated schedule details | **Vehicle maintenance table**   * Vehicle\_ID * Vehicle\_Description * Vehicle\_Name * Vehicle\_LicseneceNumber * Duration * DateTime |
| 12.2.7 Notify manager of successful update | N/A |
| 12.3View Vehicle Maintenance schedule | View Vehicle Maintenance schedule | 12.3.1 Request search Criteria | N/A |
| 12.3.2 Read User Entered Search Criteria | N/A |
| 12.3.3 Perform Schedule Search | N/A |
| 13.3.4 Display List of all Maintenances scheduled that match search results | **Vehicle maintenance table**   * Vehicle\_ID * Vehicle\_Description * Vehicle\_Name * Vehicle\_LicseneceNumber * Duration * DateTime |
| 12.3.5 Read User selected search result | N/A |
| 12.3.6 Display User selected Search result details | **Vehicle maintenance table**   * Vehicle\_ID * Vehicle\_Description * Vehicle\_Name * Vehicle\_LicseneceNumber * Duration * DateTime |
| ALT 12.3.4 Display schedule search criteria error message | N/A |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 13. Location | 13.1 Add Location | 13.1 Add Location | 13.1.1 | * Location * City * Suburb * Location Type |
|  |  |  | 13.1.2 | - |
| 13.1.3 | - |
| 13.1.4 | * Location * City * Suburb * Location Type |
| 13.1.5 | * Location * City * Suburb * Location Type |
| 13.1.6 | * Location * City * Suburb * Location Type |
| 13.1.7 | * Audit Type * Audit |
| 13.1.8 | - |
| 13.2 View Location | 13.2 View Location | 13.1.9 | * Location * City * Suburb * Location Type |
| 13.2.1 | - |
| 13.2.2 | * Location * City * Suburb * Location Type |
| 13.2.3 | * Location * City * Suburb * Location Type |
| 13.2.4 | * Audit Type * Audit |
| 13.3 Update Location | 13.3 Update Location | 13.2.5 | - |
| 13.3.1 | - |
| 13.3.2 | - |
| 13.3.3 | - |
| 13.3.4 | * Location * City * Suburb * Location Type |
| 13.3.5 | Audit Type, Audit |
| 13.3.6 | - |
|

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 14. Zone | 14.1 Add Zone | 14.1 Add Zone | 14.1.1 | * Zone |
|  |  | 14.1.2 | - |
| 14.1.3 | - |
| 14.1.4 | * Zone |
| 14.1.5 | * Zone |
| 14.1.6 | * Zone |
| 14.1.7 | * Audit Type * Audit |
| 14.1.8 | - |
| 14.2 View Zone | 14.2 View Zone | 14.1.9 | * Zone |
| 14.2.1 | - |
| 14.2.2 | * Zone |
| 14.2.3 | * Zone |
| 14.2.4 | * Audit Type * Audit |
| 14.3 Update Zone | 14.3 Update Zone | 14.2.5 | - |
| 14.3.1 | - |
| 14.3.2 | - |
| 14.3.3 | - |
| 14.3.4 | * Zone |
| 14.3.5 | Audit Type, Audit |
| 14.3.6 | - |

## 

| **Subsystem** | **Requirement** | **Use Case** | **Processes**  **(DFD)** | **Entities (ERD)** |
| --- | --- | --- | --- | --- |
|  | 15.1Generate Invoice | 15.1Generate Invoice | 15.1.1 | Booking |
| 15.1.2 | - |
| 15.1.3 | - |
| 15.1.4 | * Invoice \_ID * Booking\_Refference * Invoice\_PaymentStats * Invoice\_date * Invoice\_Quote |
| 15.1.5 | AuditType, Audit |
| 15.1.6 | - |

## conclusion

The section above contains 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 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.

## conclusion

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

|  |
| --- |
| 10. general |

## Introduction

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

## 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 and also indicated the total marks we will achieve.

|  |
| --- |
| 11. SIGN OFF |

## Introduction

See

## conclusion

See