

**Course code : “207”**

**Course Name : Software Engineering Theory**

Project Name : </FitnessHub>

**Final Project**

Full Name ID

* Mahmoud Hussienn 203130
* Mahmoud El-sayed 203133
* Omar Anan 203081

Nov 2020

Contents

[Team 1](#_Toc402452669)

[Document Purpose and Audience 2](#_Toc402452670)

[Introduction 3](#_Toc402452671)

[Software Purpose 3](#_Toc402452672)

[Software Scope 3](#_Toc402452673)

[Definitions, acronyms, and abbreviations 3](#_Toc402452674)

[Requirements 4](#_Toc402452675)

[Functional Requirements 4](#_Toc402452676)

[Non Functional Requirements 4](#_Toc402452677)

[System Models 5](#_Toc402452678)

[Use Case Model 5](#_Toc402452679)

[Use Case Tables 6](#_Toc402452680)

[Ownership Report 18](#_Toc402452681)

[Policy Regarding Plagiarism: 18](#_Toc402452682)

# Introduction

## Software Purpose

* **To Automate Gym & Fitness admission Process.**

## Software Scope

* **As Selecting Trainer is difficult and sometimes it’s difficult to get a desired slot timing.**
* **Being healthy is the first thing to be kept in mind as most of the time out attitude depends on how we feel.**

## Definitions, acronyms, and abbreviations

* **Functional : system or its component. It describes the functions a software must perform.**
* **Non-Functional : the quality attribute of a software system. They represent a set of standards used to judge the specific operation of a system.**
* **Electronic money transfer : Online Payment**

# Requirements

## Functional Requirements

* **Login.**
* **Logout.**
* **Manage the packages by adding cost, discount and delete the old packages.**
* **Manage the member details to add, update and delete.**
* **Can view the package details of individual a member.**
* **Can view the member’s attendance taken by the trainer.**
* **View their profiles and list of trainers.**
* **View the package and payment details.**
* **Giving feedback on the trainer.**
* **Take members attendance daily.**
* **Set the trainer Profile.**

## Non Functional Requirements

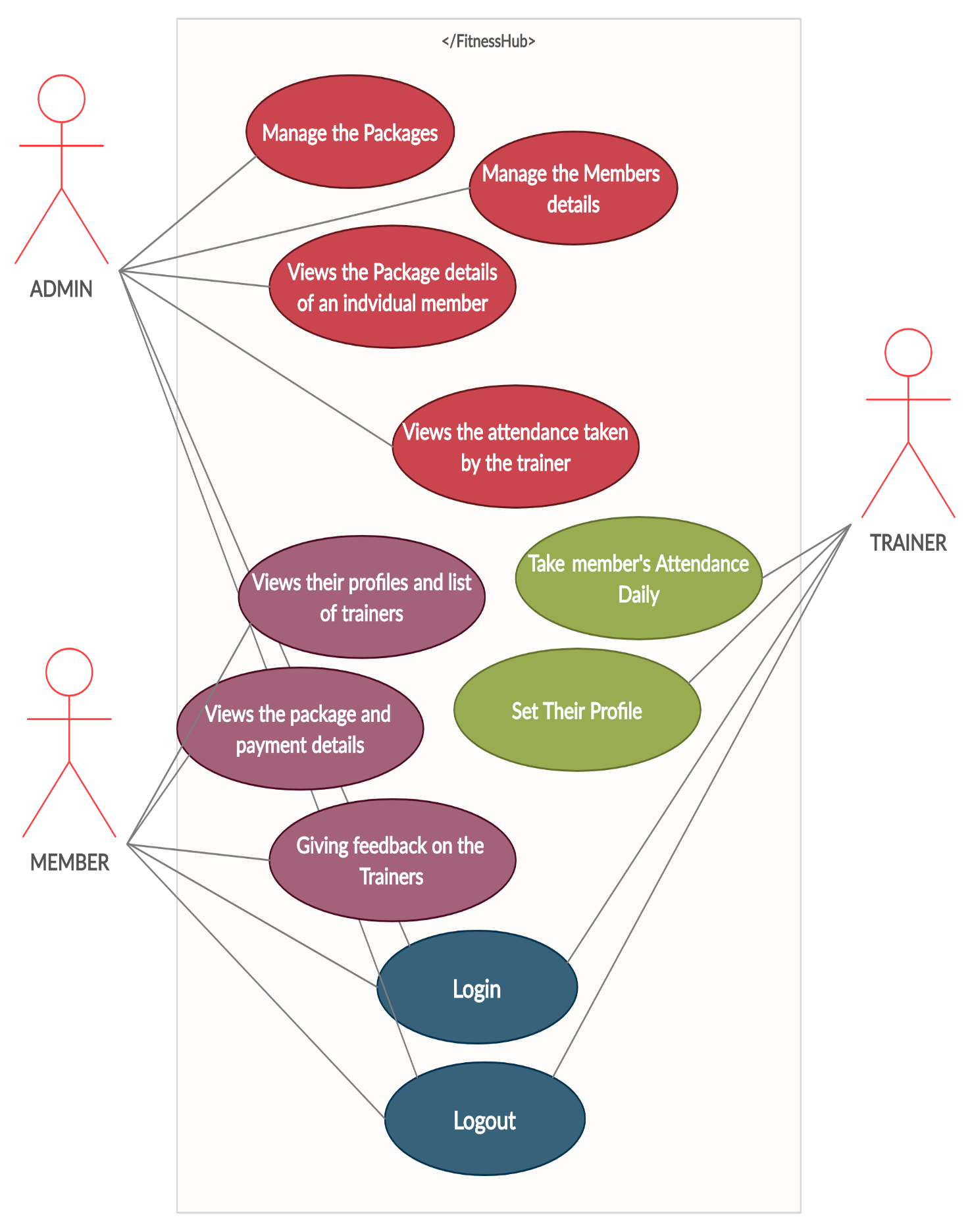
* **Software Performance**
* **Scalability**
* **Make payment use visa Card.**
* **Change password in case of security.**
* **Portability**
* **Mobile app for android and IOS.**
* **Desktop app for Win/Mac/Linux**
* **Users data Backup**

|  |  |
| --- | --- |
| **Non-Functional Requirements** | **Details** |
| **Performance** | * **Attendance operation will be done within 10 seconds and shorten the data transmission time** |
| **Scalability** | * **System should be able to support up to +1000 Member monthly** |
| **Security** | * **System should be able to Secure the users information and encrypt the info of the payment method** |
| **Portability** | * **Software should be supported by several of computing platforms** |
| **Backup** | * **Backup user’s information could be large as it’s saving a copy of all data considered worth saving** |

# 

# System Models

# Use Case Model



## Use Case Tables

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 000001 | |
| Use Case Name: | Login | |
| Actors: | Admin, Member and Trainer. | |
| Pre-conditions: | none | |
| Post-conditions: | Access the system | |
| Triggers: | Allow the users to access the system by entering valid username and password | |
| Flow of events: | **User Action** | **System Action** |
| 1- enters the login credentials |  |
|  | 2- the system checks the validity of the login details if valid the system will all function |
|
|
|
| Exceptions: | **User Action** | **System Action** |
| 1- enters the login credentials | 1- invalid login  2- Invalid Password  3- Failure in internet connection |
|
| Priority: | Must be implemented | |
| When to be available : | First increment | |
| Frequency: | 5 times a day | |
| Notes and Issues: | Can’t login without internet connection | |
| Use Case ID: | 000002 | |
| Use Case Name: | Manage The packages | |
| Actors: | Admin. | |
| Pre-conditions: | Login | |
| Post-conditions: | Add ,remove and edit the packages with the admin dashboard | |
| Triggers: | Allow the admin to edit the packages and the items | |
| Flow of events: | **User Action** | **System Action** |
| 1- enters the login credentials |  |
| 2- Manage the packages | 2- the system checks the validity of the login details if valid the system will all function |
| 3- add, remove and make discount | 3- Show list of the packages |
| 4- Save the edited package | 4- add : add new packages according to the package details  Remove : remove the old packages  Make discount : making sale on packages |
|  | 5- Edited packages saved in the database |
|
|
|
| Exceptions: | **User Action** | **System Action** |
| 1- enters the login credentials | 1- invalid login  2- Invalid Password  3- Failure in internet connection |
|
| Priority: | Should be implemented to start using the system as admin | |
| When to be available : | Second increment | |
| Frequency: | With admin privilege can edit and add or remove as long as he logged in | |
| Notes and Issues: | Can’t login without internet connection  Can’t edit the same package twice a day | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 000003 | |
| Use Case Name: | Manage the Member’s details | |
| Actors: | Admin | |
| Pre-conditions: | Login | |
| Post-conditions: | Edit the Member’s details | |
| Triggers: | Allow the Admin to edit the user’s details | |
| Flow of events: | **User Action** | **System Action** |
| 1- enters the login credentials |  |
| 2- show the admin specific details of a member | 2- the system checks the validity of the login details if valid the system will all function |
| 3- edit the details | 3- Show the member’s old details |
| 4- save the new details | 4- Save the new details in the database and remove the old ones |
|
|
|
| Exceptions: | **User Action** | **System Action** |
| 1- enters the login credentials | 1- invalid login  2- Invalid Password  3- Failure in internet connection |
|
| Priority: | Should be implemented to start using the system as admin | |
| When to be available : | Second increment | |
| Frequency: | With admin privilege can manage the members details as long as he logged in | |
| Notes and Issues: | Can’t login without internet connection  Can’t edit the same details twice a day | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 000004 | |
| Use Case Name: | View the package details of an individual member | |
| Actors: | Admin | |
| Pre-conditions: | Login | |
| Post-conditions: | view the Member’s package details | |
| Triggers: | Allow the Admin to view the user’s package details | |
| Flow of events: | **User Action** | **System Action** |
| 1- enters the login credentials |  |
| 2- view the package details of a specific member | 2- the system checks the validity of the login details if valid the system will all function |
|  | 3- Show the member’s package details |
|
|
|
| Exceptions: | **User Action** | **System Action** |
| 1- enters the login credentials | 1- invalid login  2- Invalid Password  3- Failure in internet connection |
|
| Priority: | Should be implemented to start using the system as admin | |
| When to be available : | Second increment | |
| Frequency: | With admin privilege can see the details of a specific member as long as he logged in | |
| Notes and Issues: | Can’t login without internet connection | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 000005 | |
| Use Case Name: | View the attendance | |
| Actors: | Admin | |
| Pre-conditions: | Login | |
| Post-conditions: | view the Member’s attendance taken by the trainer | |
| Triggers: | Allow the Admin to view the user’s attendance taken by the trainer | |
| Flow of events: | **User Action** | **System Action** |
| 1- enters the login credentials |  |
| 2- view the attendance of the members taken by the trainer | 2- the system checks the validity of the login details if valid the system will all function |
|  | 3- system open the attendance page |
|
|
|
| Exceptions: | **User Action** | **System Action** |
| 1- enters the login credentials | 1- invalid login  2- Invalid Password  3- Failure in internet connection |
|
| Priority: | Should be implemented to start using the system as admin | |
| When to be available : | third increment | |
| Frequency: | With admin privilege can see the attendance of the members as long as he logged in | |
| Notes and Issues: | Can’t login without internet connection | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 000006 | |
| Use Case Name: | View their profiles | |
| Actors: | Members | |
| Pre-conditions: | Login | |
| Post-conditions: | view their profiles and a list of trainers | |
| Triggers: | Allow the members to edit their profile and choose the desired trainer | |
| Flow of events: | **User Action** | **System Action** |
| 1- enters the login credentials |  |
| 2- view the profile | 2- the system checks the validity of the login details if valid the system will all function |
| 3- choose the desired trainer from a list | 3- system open the profile page |
|  | 4- the system sends a email to the trainer to inform him about the new registered member to his class |
|
|
|
| Exceptions: | **User Action** | **System Action** |
| 1- enters the login credentials | 1- invalid login  2- Invalid Password  3- Failure in internet connection |
|
| Priority: | Should be implemented to start using the system as a member | |
| When to be available : | second increment | |
| Frequency: | Can view their profile as long as they logged in | |
| Notes and Issues: | Can’t login without internet connection | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 000007 | |
| Use Case Name: | View the packages | |
| Actors: | Members | |
| Pre-conditions: | Login | |
| Post-conditions: | view the packages details and the payment information | |
| Triggers: | Allow the members to see their package and the payment details | |
| Flow of events: | **User Action** | **System Action** |
| 1- enters the login credentials |  |
| 2- view their package | 2- the system checks the validity of the login details if valid the system will all function |
| 3- view the payment details | 3- system open the packages page |
|  | 4- the system shows up the payment details |
|
|
|
| Exceptions: | **User Action** | **System Action** |
| 1- enters the login credentials | 1- invalid login  2- Invalid Password  3- Failure in internet connection |
|
| Priority: | Should be implemented to start using the system as a member | |
| When to be available : | second increment | |
| Frequency: | Can view their packages and the payment details as long as they are logged in | |
| Notes and Issues: | Can’t login without internet connection  Can’t see the whole payment details in case of security | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 000008 | |
| Use Case Name: | Giving feedback | |
| Actors: | Members | |
| Pre-conditions: | Login | |
| Post-conditions: | Giving feedback on their trainer | |
| Triggers: | Allow the members give feedback on their trainer | |
| Flow of events: | **User Action** | **System Action** |
| 1- enters the login credentials |  |
| 2- Feedback | 2- the system checks the validity of the login details if valid the system will all function |
| 3- giving the feedback | 3- system open the feedback page |
|  | 4- the system put the rate they chooses in starts shape |
|
|
|
| Exceptions: | **User Action** | **System Action** |
| 1- enters the login credentials | 1- invalid login  2- Invalid Password  3- Failure in internet connection |
|
| Priority: | Should be implemented to start using the system as a member | |
| When to be available : | Third increment | |
| Frequency: | Giving feedback once or in case of changing the trainer | |
| Notes and Issues: | Can’t login without internet connection  Can’t change the rate or the feedback once it been written | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 000009 | |
| Use Case Name: | Take member’s attendance | |
| Actors: | Trainer | |
| Pre-conditions: | Login | |
| Post-conditions: | Take the members that registered in his table attendance daily | |
| Triggers: | Allow the Trainer to take the attendance daily in the class | |
| Flow of events: | **User Action** | **System Action** |
| 1- enters the login credentials |  |
| 2- Attendance | 2- the system checks the validity of the login details if valid the system will all function |
| 3- take the attendance | 3- system open the Attendance page |
|  | 4- the system saves the attendance and sends it to the admin |
|
|
|
| Exceptions: | **User Action** | **System Action** |
| 1- enters the login credentials | 1- invalid login  2- Invalid Password  3- Failure in internet connection |
|
| Priority: | Should be implemented to start using the system as a trainer | |
| When to be available : | Second increment | |
| Frequency: | Take attendance daily | |
| Notes and Issues: | Can’t login without internet connection  Can’t edit the attendance information | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 000010 | |
| Use Case Name: | Set their profile | |
| Actors: | Trainer | |
| Pre-conditions: | Login | |
| Post-conditions: | Setting up their profiles | |
| Triggers: | Setting up the profiles to let the new members see their experience | |
| Flow of events: | **User Action** | **System Action** |
| 1- enters the login credentials |  |
| 2- Profile | 2- the system checks the validity of the login details if valid the system will all function |
| 3- Set up his profile | 3- system open the trainer Profile page |
|  | 4- the system saves the trainer details |
|
|
|
| Exceptions: | **User Action** | **System Action** |
| 1- enters the login credentials | 1- invalid login  2- Invalid Password  3- Failure in internet connection |
|
| Priority: | Should be implemented to start using the system as a trainer | |
| When to be available : | Second increment | |
| Frequency: | Set up the profile once a month | |
| Notes and Issues: | Can’t login without internet connection | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: | 000011 | |
| Use Case Name: | Logout | |
| Actors: | Admin, Member and Trainer | |
| Pre-conditions: | Login | |
| Post-conditions: | none | |
| Triggers: | Closing the system | |
| Flow of events: | **User Action** | **System Action** |
| 1- enters the login credentials |  |
| 2- explore the system | 2- the system checks the validity of the login details if valid the system will all function |
| 3- logout | 3- exploring the system without any issues |
|  | 4- the system saves the session and closes |
|
|
|
| Exceptions: | **User Action** | **System Action** |
| 1- enters the login credentials | 1- invalid login  2- Invalid Password  3- Failure in internet connection |
|
| Priority: | Must be implemented | |
| When to be available : | First increment | |
| Frequency: | Logout daily | |
| Notes and Issues: | Can’t login without internet connection can’t logout once logged in directly | |