

SOFTWARE REQUIREMENT SPECIFICATION (SRS)

FOR

STUDENT GYM MEMBERSHIP
AND SESSION BOOKING
SYSTEM (STUGYM)

BY

OLAKUNLE ADEGEYE

Table of Content

Table of Content	0
List of Tables	1
Table of Figures	3
1.0. INTRODUCTION	4
1.1 Purpose, Scope and Objectives	4
1.1.1. Purpose.....	4
1.1.2. Product Scope	4
1.1.3. Objectives:	5
1.2. Project description.....	5
1.3. Project Deliverables	6
1.4. Development Environment	6
1.5. Definition of Terms, Acronyms and Abbreviations.....	7
2.0. REQUIREMENT COLLECTION FOR OO DESIGN.....	8
2.1. Requirement collection methodology	8
2.1.1. One-on-one interview with students	8
2.1.2. One-on-one interview with gym instructors	10
2.1.3. Exploring fitness websites including	10
2.2. Functional Requirements.....	11
2.2.1. User Functional Requirement	11
2.2.2. Administrative Functional Requirements	13
2.3. Non-functional Requirements	14
3.0. SYSTEM DESIGN	15
3.1. Use cases and use case UML models.....	15
3.1.1. Use Case diagram	15
3.1.2. Activity Diagram	16
3.1.3. Activity diagram for Administrative Users.....	17
3.1.4. Use Case Description.....	18
3.2. Class diagram	25

List of Tables

One-on-one interview with students 9

Table 2: One-on-one interview with gym instructors 10

Table 3: User Functional Requirements..... 12

Table 4: Administrative Functional Requirement..... 13

Table 5: Non-Functional Requirement 14

Table 6: UC_001..... 18

Table 7: UC_002..... 19

Table 8: UC_003..... 20

Table 9: UC_004..... 21

Table 10: UC_005..... 22

Table 11: UC_006..... 23

Table 12: UC_007..... 24

Table of Figures

Fig. 1: Use Case Diagram	15
Fig. 2: Activity Diagram for Students.....	16
Fig. 3: Activity Diagram for Administrative Users	17
Fig. 4: Class Diagram	25

SRS Revision History

Name	Date	Details.	Version
Olakunle	11/11/2022	Creation of SRS file	1.0

1.0.INTRODUCTION

1.1 Purpose, Scope and Objectives

1.1.1. Purpose

The purpose of this document is to provide a comprehensive overview of the goals and requirements of the student gym membership and session booking system which will hereafter be referred to as ‘Stugym’.

The document would highlight the purpose of the system and explain the system constraints and dependencies, mode of operation, functionalities, development environment, design patterns, interface, interactions and the intended usage of the system.

1.1.2. Product Scope

Stugym would be tailored specifically for the students at all levels of the University. It would not be accessible by non-students. Stugym is a solution that is designed to solve the problems associated with the current gym usage and membership where the gym gets crowded at specific times, and the staff are unable to properly track the its usage.

The system would allow students to register every semester and choose a time within the available periods. The system would have a quota for usage within a specific period of time to ensure that the gym is not overcrowded at any time, and students have access to all equipment. The gym would operate from 7:00 am to 7:00 pm from Monday to Saturday, each session would last for two hours, and when students want to register, they immediately see the available times.

Students would only be allowed a limited number of gym usage within an academic semester, week and gym session. The students can choose to allocate their time as they wish, but within the constraints of the system. Once a session has been missed, the time is non-redeemable. This would also ensure that the admin does not need to always update the system manually.

The platform would be accessible via a web address, but only verified students of the University would have access. This means that a login information would be required to book a session.

Students would have the option of choosing between a cheaper basic package or a more expensive premium package that includes a personal trainer. The basic package would be inclusive with the students' school fees, while the premium package would cost an additional 200 pounds per session.

The administrator would have access to the backend to see student registrations and usage.

All information on the backend would be available in a database and downloadable as a csv file for further analysis on BI package.

1.1.3. Objectives:

The overarching objective is to create a gym membership system that automates the entire process of student membership and usage.

- To automate the process of student membership and expiration.
- To eliminate overcrowding of the gym space and allow for access to all equipment through an automated time allocation system.
- To eliminate administrative errors in tracking the activity log and membership of students.

1.2. Project description

Stugym is a student gym membership and session booking system website for students of the University. The solution would allow students to register for their gym sessions for each system and also employ a semi-automatic scheduling system. Semi-automatic in the sense that the students would be allowed to choose their session time, however it would be done within the constraints of the system such as the maximum allowable time per session and week, and the available time slots.

The solution would automate the membership registration and usage system, and also allow for seamless administrative monitoring and reporting.

1.3. Project Deliverables

Name	Date	Details
Initial SRS File	11/11/2022	Introduction, Requirement Collection, and System Design

1.4. Development Environment

Development Tools: Django, Visual Code Editor

Database: MySQL

Operating System: OS, Windows, Mac, Linux, Android, Web

The solution consists of a client-server architecture which is the front-end and the back-end. The front-end can be accessed via a web browser on any of the major operation systems. It would be designed using Django. The database would be designed using MySQL

1.5. Definition of Terms, Acronyms and Abbreviations

FR: Functional Requirements

NFR: Non-Functional Requirements

UC: Use Case

Stugym: The name given to the student gym membership and session booking system, developed as a portmanteau of ‘student’ and ‘gym’

2.0. REQUIREMENT COLLECTION FOR OO DESIGN

2.1. Requirement collection methodology

The following methods were used for requirement collection:

- One-on-one interview with students
- One-on-one interview with some trainers
- Exploring fitness websites including the University's sports facility

2.1.1. One-on-one interview with students

In order to gather requirements for the features that relate specifically to the students, a number of students were interviewed.

The table below shows the information regarding the questions that were posed, the answers that were provided and the requirements that were developed.

S/N	Questions	Answer	Requirements
1	Do you have any gym membership? For those who didn't have, the following follow up questions were asked: a. Have you held a membership at any fitness center? b. Do you intend to have a membership in the future?	The responses revealed that some students had memberships at a fitness center, while others either had previous memberships, or were considering getting one. Some students claimed that the process of going to register made them procrastinate. They claimed that if it was inclusive with the fees, it would have been easier for them to just go to the gym.	To include the fee for the basic gym membership subscription in the school fees. This would obviously involve liaising with the management of the University by presenting both the financial benefits to the school and health benefits to the students.
2	How many times a week do you visit the gym per week?	It was discovered that the average number of days students spent going to the gym per week was 3 days	Limit the gym session to a maximum of 3 days per week

3	What is the maximum number of hours you spend in the gym per session?	It was discovered that students spent an average of approximately 1hr 30 mins hours in the gym.	Limit the gym session to 2 hours daily
4	Do you make use of a personal trainer at the gym?	The responses showed that not all students had a personal trainer, but those who didn't have personal trainers reported that they would have subscribed for a personal trainer if the price was more affordable.	Create a monthly subscription plan that would be cheaper than paying daily.
5	What are the challenges you experience in the gym?	Some of the responses related to inaccessibility to some of the equipment due to the fact that there were usually more people than equipment at certain times.	Limit the gym session to a maximum of 2 hours per session and 3 days per week to reduce congestion

Table 1: One-on-one interview with students

2.1.2. One-on-one interview with gym instructors

S/N	Questions	Answer	Requirements
1	How many days do you work per week?	Most of the gym instructors worked 5 days a week	Automate the system to limit their working hours to 5 days a week
2	How many clients do you train per day?	As regards the number of clients trained per day, the number varied across the trainers from a minimum of 4 to a maximum of 20	Limit the number of clients per 2 - hour gym session to 2 clients and 3 days per week.
3	What are the challenges you face with your clients?	The challenges faced related to having too many clients to focus on at the same time. There were also inconsistencies of the gym members as regards their personal fitness goals.	Limit the number of clients per gym session to 2 clients and 3 days per week. Create an automatic and non-adjustable schedule system that would create order around their gym sessions and motivate the members to maintain their schedule

Table 2: One-on-one interview with gym instructors

2.1.3. Exploring fitness websites

To supplement the information gathered from the interview, I explored the internet for student gym websites, including the school's Sports facility, and general fitness websites.

2.2. Functional Requirements

2.2.1. User Functional Requirement

This section captures the requirements of the system from the students' perspective

UC ID	Use Case/Epic	User Story	Req Code	Requirement
UC_001	Membership Registration	As a student, I want to be able to register to the gym membership platform so that I can become a member	FR1	The system must allow students to register with their personal information and verified registration number
			FR2	The system must allow students to select their subscription plan by choosing between the free basic plan or the paid premium plan
			FR3	The system must allow premium members to pay for their subscriptions using their credit cards
			FR4	The system allows users to update their profile with their age, body details and profile picture.
UC_002	Session booking	As a student I want to be able to book my gym session so that I know the days I can visit the gym	FR5	System must allow students to pick their gym time from the available time slots in the system's calendar
			FR6	The system must only allow registration within the working hours of 6:00 am to 8:00 pm
			FR7	The system must only allow a maximum gym time of 2 hours daily
			FR8	The system must only allow maximum gym days of 3 days per week (6 hours)

			FR9	The system must only allow a maximum of 24 hours per month per student
			FR10	Only 5 users can be assigned to a specific gym session of 2 hours
			FR11	Premium members automatically get a trainer assignment for each 2 hour gym session
			FR12	The selected dates are not changeable once they have been selected
			FR13	Once a session as elapsed, the time is not recoverable
UC_003	Profile viewing	As a user, I want to be able to see my personal details, schedule and usage statistics when I view my profile	FR14	Users can view personal details, schedule and usage statistics in profile

Table 3: User Functional Requirements

2.2.2. Administrative Functional Requirements

This section captures the requirements of the system from the administrators' perspective

The Administrative comprises the Admin, Employees and Trainers.

UC ID	Use Case	User Story	Req Code	Requirement
UC_004	CRUD operations	As the Admin, I want to have access to all administrative privileges so that I can manage the system	FR15	Admin can create, read, update and delete profiles and information for all system users (Students, Employees, Trainers)
			FR16	Admin can update database with records of verified students
UC_005	Schedule viewing	As an admin, I want to be able to view the schedule of all students, so that I can monitor the operations	FR17	Admin has access to view students' profile and schedule
			FR18	Admin can query database to view students schedule and details
UC_006	Automatic assignment	As the Admin, I want the system to automate the assignment process so that I can reduce human errors	FR19	Only 2 students can be assigned to each trainer per gym session of 2 hours
			FR20	The system must assign a maximum of 10 hours daily to each personal trainer
UC_007	Automatic verification	As the Admin, I want the system to automate the student verification process, so that I can reduce human errors	FR21	The system must only allow verified_students to register

Table 4: Administrative Functional Requirement

2.3. Non-functional Requirements

ID	Category	NFR_Code	Requirement
001	Security	NFR1	User must not be able to update personal information and school registration number once registered The system must automatically detect if the user is a verified student of the school
		NFR2	Students and employees' password cannot be less than 10 characters
		NFR3	Only Master admin has access to all administrative privileges
		NFR4	Employees have read-only access
002	Compatibility	NFR5	The website must be accessible on all major operating systems: Windows, Linux, Unix, Android, IOS
003	Usability	NFR6	The website must be adaptive to both PC and mobile devices
		NFR7	The website should be user-friendly
004	Reliability	NFR8	Students can access the platform 95% of the time without failure
		NFR9	Up to 10000 students can registered on the system
005	Data integrity	NFR10	The system automatically database backup every 8pm to prevent data loss during system crash
		NFR11	User information is not deleted from the database when student deletes account
006	Performance	NFR12	The system should allow the creation of cookies and caching data on user's web browser to speed up loading time.

Table 5: Non-Functional Requirement

3.0. SYSTEM DESIGN

3.1. Use cases and use case UML models

3.1.1. Use Case diagram

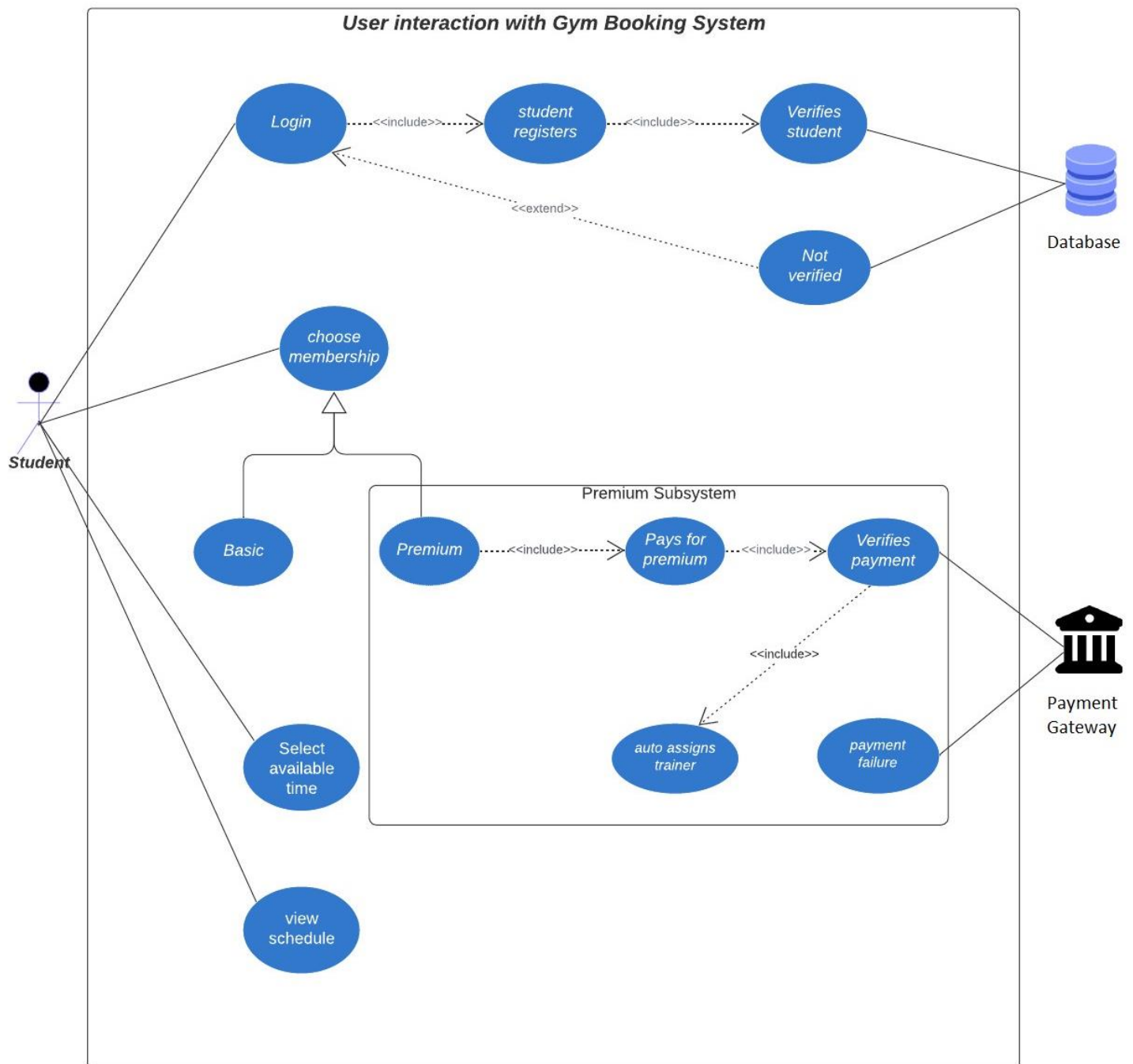


Fig 1: Use Case Diagram

3.1.2. Activity Diagram

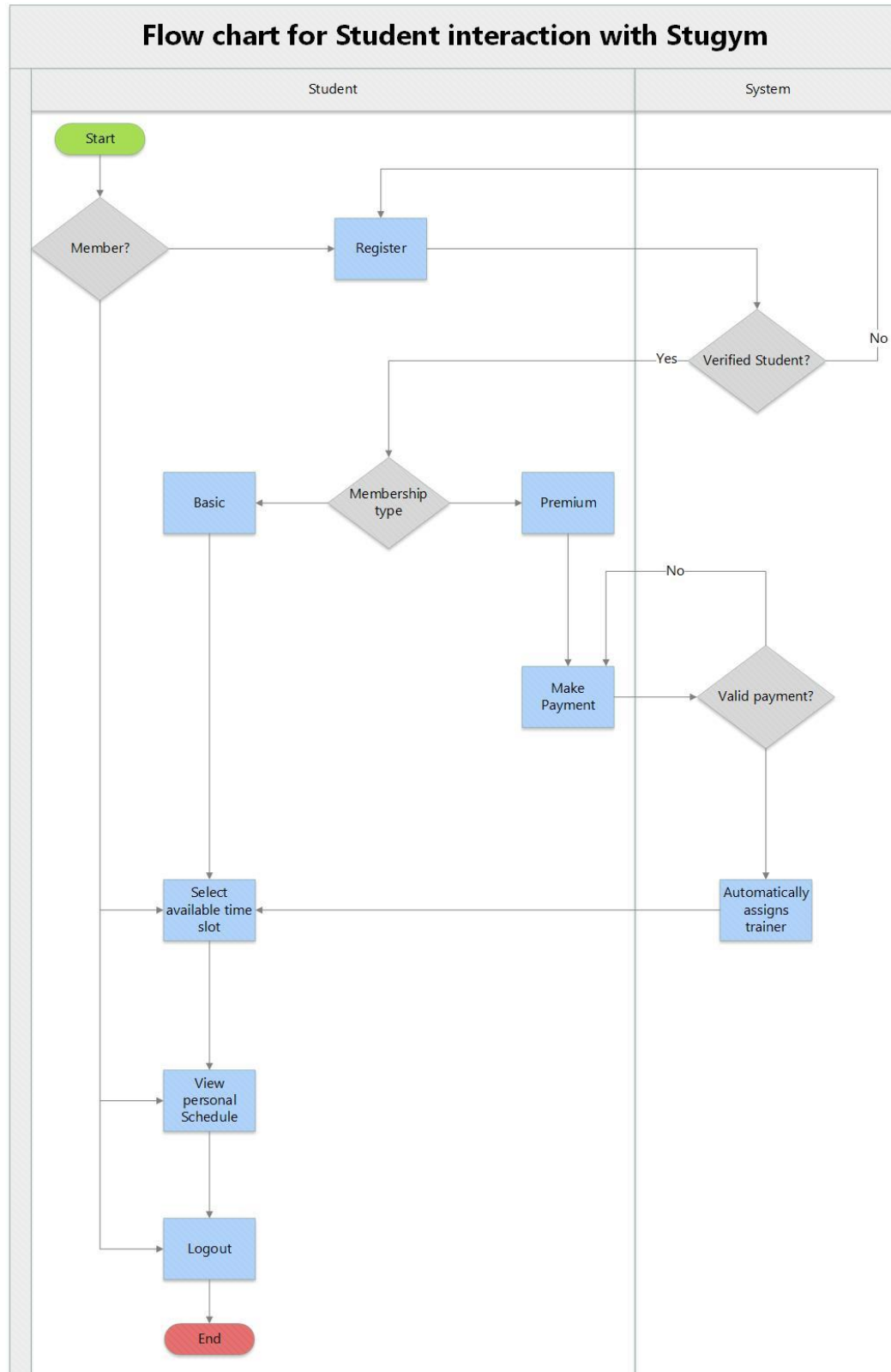


Fig 2: Activity Diagram for Students

3.1.3. Activity diagram for Administrative Users

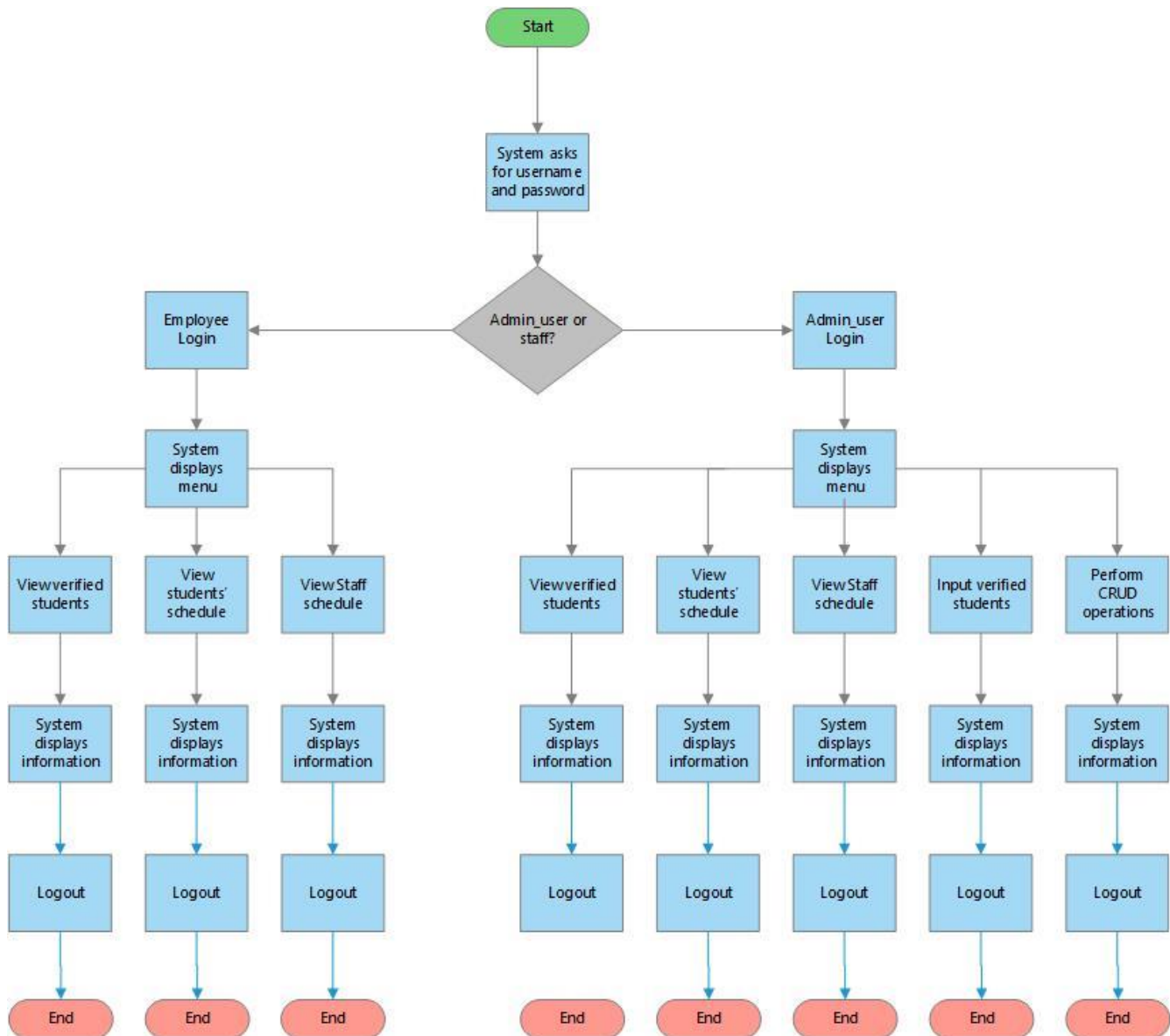


Fig 3: Activity Diagram for Administrative User

3.1.4. Use Case Description

Use Case ID:	UC_001	Use Case Name:	Membership Registration
Created by:	Olakunle	Date	11/11/2022
Actor	Student		
Goal:	The student registers for a gym membership		
Description			
This use case shows the flow of the student through the process of registering for a gym membership. The system automatically verifies and allows the student to choose between the free basic plan which comes with their studentship, or the paid premium plan which comes with a personal trainer.			
Functional Requirements	FR1, FR2, FR3, FR4		
Triggers	The student wants to register for a gym membership		
Pre-conditions	Must be a verified student of the University whose details are contained in the verified_students table		
Post-conditions	Student has a gym membership		
Basic Flow			
Step	User Activity	Response	
1	Student visits the Stugym website on PC or mobile	Adaptive website opens	
2	Student clicks 'Register'	Goes to registration page	
3	Student inputs personal information such as first_name, last_name, email, age, body details and school registration number and clicks 'next'	System verifies information and navigates to the membership selection page	
4.1	User selects basic membership	System takes students to the schedule selection page	
4.2	User selects premium membership	System takes student to payment page	
5	Student inputs valid card details	System confirms payment and takes students to schedule select page	
Alternative Flow			
Step	User Activity	Response	
3	3a. Student inputs school registration number that is not up to 10 digits or does not follow the format,	System notifies student to check the length of the registration number and the format	
5	5a Student inputs invalid card details	System takes notifies student and takes student back to payment page	
Exceptional Flow			
Step	User Activity	Response	
1	Website fails to load	System shows error code and directs student to contact admin	
3	3a. students' registration number is not in the verified_students list	system notifies student that the number is not verified, and directs the student to confirm with the school bursary.	

Table 6: UC_001

Use Case ID:	UC_002	Use Case Name:	Session Booking
Created by:	Olakunle	Date	11/11/2022
Actor	Student		
Goal:	To select gym schedule from the available time slots in order to prevent clashes and overcrowding at the gym		
Description			
This use case shows the flow of the student through the process of selecting a gym schedule from the available time slots. The system employs an automatic assignment model and would only show the time slots that are available. Premium members get personal trainer assignment alongside their time assignment.			
Functional Requirements	FR5, FR6, FR7, FR8, FR9, FR10, FR11, FR12, FR13		
Triggers	The students want to select time schedule		
Pre-conditions	Must have been verified by the system during registration		
Post-conditions	Student has an unchangeable time schedule		
Basic Flow			
Step	User Activity	Response	
1	Student visits the schedule selection page	System shows calendar of available days and available time slots	
2	Member clicks the monthly calendar	System shows available days	
3	Member clicks the available days	System shows the available time	
4.1	Basic member selects from the available time slot	System assigns the time	
4.2	Premium member selects from the available time slot	System assigns the time and personal trainer	
5	Student reaches the 24-hour maximum selection per month and clicks confirm	System records the time and locks the student from changing the time for that month.	
Alternative Flow			
Step	User Activity	Response	
2.1	4.1a. Student selects more than 2 hours	System notifies student that they can only select two hours daily.	
4.1	4.1b. Student select time slot that has been filled up	System notifies student to select from another time slot.	
4.1	4.1c. Student selects more than 6 hours a week	System notifies students that they can only select a maximum of 6 hours a week.	
4.2	4.2a Premium member selects time slot where personal trainer is not available	System asks student to select another time slot or proceed without a personal trainer	
Exceptional Flow			
Step	User Activity	Response	
4.1	All time slots are unavailable for the month	System directs student to contact admin	
4.2	All personal trainers are unavailable for the month	system notifies student to contact admin	

Table 7: UC_002

Use Case ID:	UC_003	Use Case Name:	Profile Viewing
Created by:	Olakunle	Date	11/11/2022
Actor	Student		
Goal:	To view the user’s time schedule and usage statistics		
Description			
This use case shows the flow of the student through the process of viewing their profile in order to see their gym schedule and usage statistics. The user would be able to see the hours elapsed and the time left.			
Functional Requirements	FR14		
Triggers	The student wants to view their profile and usage statistics		
Pre-conditions	The student must have selected a time schedule for the period and confirmed selection		
Post-conditions	Student sees usage statistics and schedule		
Basic Flow			
Step	User Activity	Response	
1	Student logs in with valid ID	System takes the student to Stugym home page	
2	Student clicks on Profile	System takes student to profile and shows profile information, person gym schedule and usage statistics	
Alternative Flow			
Step	User Activity	Response	
1	Student logs in with invalid information	System notifies student and asks student to choose input correct information and shows ‘reset password’ option	
Exceptional Flow			
Step	User Activity	Response	

Table 8: UC_003

Use Case ID:	UC_004	Use Case Name:	Admin CRUD operations
Created by:	Olakunle	Date	11/11/2022
Actor	Admin		
Goal:	To access the backend of the system to manage the entire operations		
Description			
This use case shows the flow of Admin through the process of managing the back-end operations of the website and performing all CRUD operations on the system.			
Functional Requirements	FR15, FR16		
Triggers	Admin wants to perform CRUD operation on a student or trainer		
Pre-conditions	Must have Admin login information		
Post-conditions	CRUD operations performed		
Basic Flow			
Step	User Activity	Response	
1	Admin visits the admin login page	System takes the Admin to Stugym's admin login page	
2	Admin enters valid username and password	System takes Admin to the back-end of the website and provides access to all CRUD operations for students and trainer	
	Admin clicks on verified_students table	System takes user to the table and shows all CRUD operations	
	Admin user inserts verified_students individually	System adds the verified_user	
Alternative Flow			
Step	User Activity	Response	
2	Admin user inserts invalid or incomplete registration number, or incomplete student name	System notifies user and tells user to check details again	
Exceptional Flow			
Step	User Activity	Response	

Table 9: UC_004

Use Case ID:	UC_005	Use Case Name:	Schedule Viewing
Created by:	Olakunle	Date	11/11/2022
Actor	Admin		
Goal:	To access the backend of the system to view the schedule of all students and staff, so that the admin can monitor operations		
Description			
This use case shows the flow of the admin through the process of viewing the individual profiles of the students and employees, and the general schedule for a time period by querying the database.			
Functional Requirements	FR17, FR18		
Triggers	Admin wants to view information and schedule of students and staff to manage operations		
Pre-conditions	Must have admin login information		
Post-conditions	Admin views back-end operations		
Basic Flow			
Step	User Activity	Response	
1	Admin visits the admin login page	System takes the Admin to Stugym's admin login page	
2	Admin enters valid username and password	System takes staff to the back-end of the website and provides view access to verified_students table and to both staff and students profiles and schedules.	
3	Admin clicks on user profile	System shows profile details	
4	Admin queries database	System returns result	
Alternative Flow			
Step	User Activity	Response	
2	Admin inserts invalid or incomplete registration number, or incomplete student name	System notifies user and tells user to check details again	
Exceptional Flow			
Step	User Activity	Response	
	User attempts to perform create, update or delete information	System does not provide access	

Table 10: UC_005

Use Case ID:	UC_006	Use Case Name:	Automatic assignment
Created by:	Olakunle	Date	11/11/2022
Actor	System		
Goal:	To automate the process of time schedule and personal trainer assignment		
Description			
This use case shows the automated process carried out by the system to automatically limit the number of assigned premium members to each trainer, and limit the working hours of each trainer to 10 hours.			
Functional Requirements	FR19, FR20		
Triggers	Students select gym schedule		
Pre-conditions	Student must be a verified member		
Post-conditions	Available slots are allocated and clashes are prevented		
Basic Flow			
Step	User Activity	Response	
1	Personal trainer is added	System automatically sets a maximum working hour of 10 hours	
2	Two Premium students picks a specific available time slot	System searches for available trainer and assigns him to them	
3	Third premium member picks the same time slot	System notices that the previous personal trainer already has two students for that time slots, and assigns a different trainer	
Alternative Flow			
Step	User Activity	Response	
Exceptional Flow			
Step	User Activity	Response	

Table 11: UC_006

Use Case ID:	UC_007	Use Case Name:	Automatic verification
Created by:	Olakunle	Date	11/11/2022
Actor	System		
Goal:	To automate the process of student verification during registration		
Description			
This use case shows the automated process carried out by the system to automatically verify students during registration. The verified students are first inputted by the Admin. During student registration, the system cross checks the first_name, last_name and registration number provided by the student against the one contained in the verified_students table before approving membership.			
Functional Requirements	FR16, FR21		
Triggers	Admin wants to view information and schedule of students and staff to manage operations		
Pre-conditions	Admin must have populated the database with verified students.		
Post-conditions	Verified students are given membership after registration		
Basic Flow			
Step	User Activity	Response	
1	Admin prepopulates the verified_students table with the first_name, last_name and valid registration number of verified students.	The system accepts the input of correctly formatted information and stores in the database	
2	Student inputs personal details during registration.	System cross-checks the information against the information contained in the verified_students table.	
3	Students information matches information in the database	System approves the students' registrations	
Alternative Flow			
Step	User Activity	Response	
2	Student inputs incomplete or incorrect details	System notifies user and tells user to check details again	
Exceptional Flow			
Step	User Activity	Response	
	Students' information is not available in the database	System directs student to contact the school bursary	

Table 12: UC_007

3.2. Class diagram

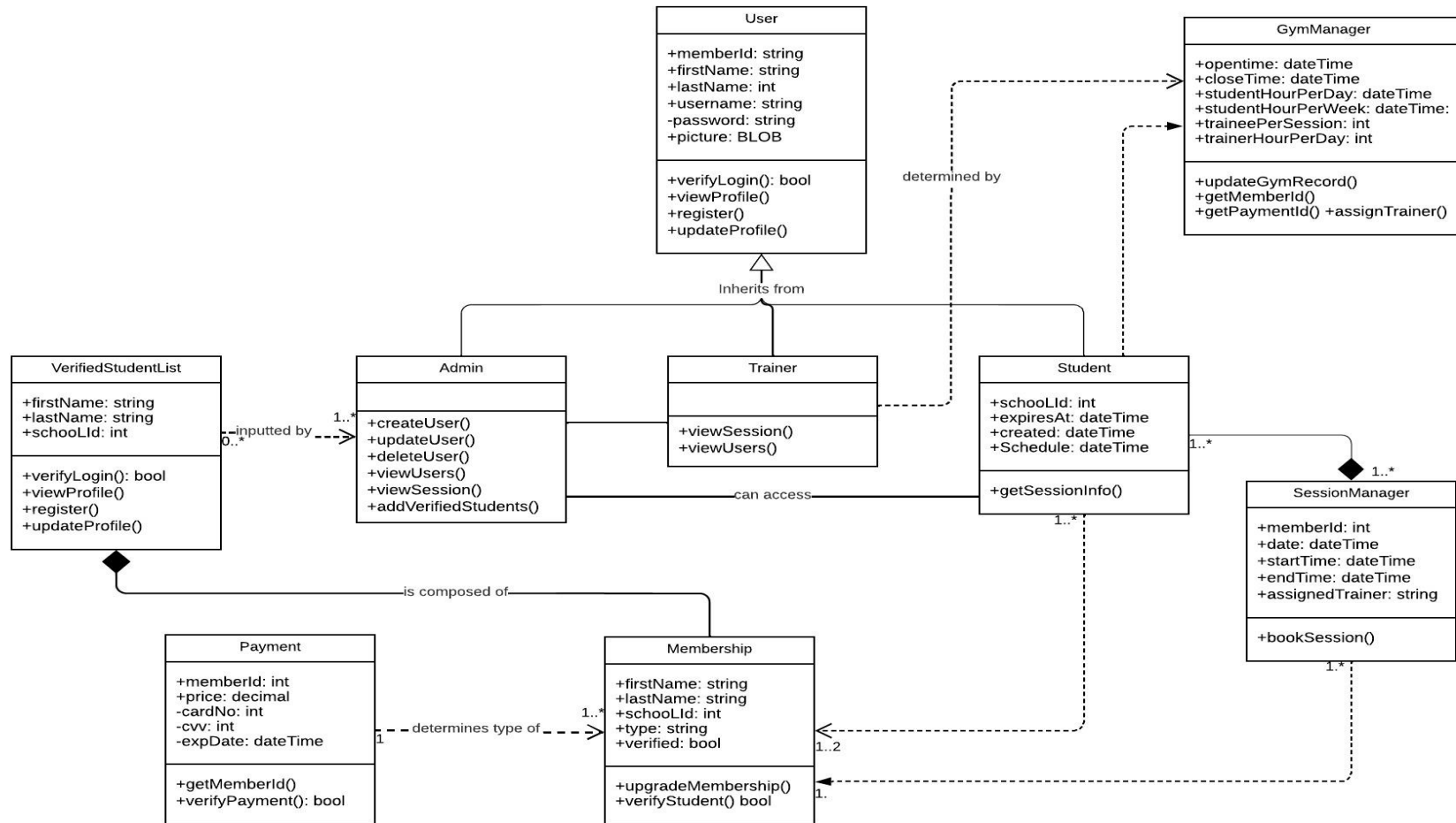


Fig 4: Class Diagram