



---

# Software Requirements Specification

for

## A5-swimiNIT

Version 1.1

Prepared by

Team Number: 05

Lenoah Chacko  
Pavithra Rajan  
Abin Gigo Joseph  
Joseph Mani Jacob Mani  
Akshay Kuttikkattuparambil Biju  
Varun Chittezhath Anilkumar

B190657CS  
B190632CS  
B190880CS  
B190529CS  
B190803CS  
B190621CS

**Project Owner:**

Ms. Lekshmy P Chandran

**Course:**

CS4096D Software Engineering Laboratory

**Date:**

17/02/2022

This template is based on the one available from GMU site by Dr Rob Pettit. Modifications specific to NITC are made and will be used for academic purposes only.

# Contents

<b>1 Introduction</b>	<b>4</b>
1.1 Document Purpose	4
1.2 Product Scope	4
1.3 Intended Audience and Document Overview	4
1.4 Definitions, Acronyms and Abbreviations	4
1.5 Document Conventions	5
1.6 References and Acknowledgments	5
<b>2 Overall Description</b>	<b>5</b>
2.1 Product Overview	5
2.2 Product Functionality	7
2.3 Design and Implementation Constraints	7
2.4 Assumptions and Dependencies	8
<b>3 Specific Requirements</b>	<b>8</b>
3.1 External Interface Requirements	8
3.1.2 Hardware Interfaces	8
3.1.3 Software Interfaces	8
3.2 Functional Requirements	8
3.3 Use Case Model	10
3.3.1 Use Case #1 (Login – U1)	11
3.3.2 Use Case #2 (Register a new swimmer – U2)	12
3.3.3 Use Case #3 (Swimmer Entry – U3)	13
3.3.4 Use Case #4 (Swimmer Exit – U4)	15
3.3.5 Use Case #5 (Search pool history by date – U5)	16
3.3.6 Use Case #6 (View swimmer details and history – U6)	17
3.3.7 Use Case #7 (Mail regarding pool updates – U7)	18
3.3.8 Use Case #8 (Edit swimmer details – U8)	19
3.3.9 Use Case #9 (Pool Status – U9)	20
3.3.10 Use Case #10 (View Dues – U10)	21
3.3.11 Use Case #11 (View all SPMs – U11)	22
3.3.12 Use Case #12 (Create SPM account – U12)	23
3.3.13 Use Case #13 (Delete SPM account – U13)	24
3.3.14 Use Case #14 (Download Quarterly Visit and Collection Report – U14)	25
3.3.15 Use Case #15 (Edit swimmer’s receipt details – U15)	26
3.3.16 Use Case #16 (Log out – U16)	27

<b>4 Other Non-functional Requirements</b>	<b>28</b>
4.1 Performance Requirements	28
4.2 Safety and Security Requirements	28
4.3 Software Quality Attributes	28
4.3.1 Reliability	28
4.3.2 Correctness	28
4.3.3 Availability	28
4.3.4 Scalability	28
4.3.5 Maintainability	28
4.3.6 Portability	28
4.3.7 Extensibility	29
4.3.8 Reusability	29
4.3.9 Usability	29
4.3.10 Robustness	29
<b>5 Other Requirements</b>	<b>29</b>
<b>Appendix A - Activity Log</b>	<b>29</b>
<b>Appendix B - User Interface</b>	<b>30</b>
<b>Appendix C</b>	<b>48</b>

## Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft-02	Lenoah Chacko Pavithra Rajan Abin Gigo Joseph Joseph Mani Jacob Mani Akshay Kuttikkattuparambil Biju Varun Chittezhath Anilkumar	Revision made based on feedback received on 12/02/2022	17/02/22
Draft-01	Lenoah Chacko Pavithra Rajan Abin Gigo Joseph Joseph Mani Jacob Mani Akshay Kuttikkattuparambil Biju Varun Chittezhath Anilkumar	Document created based on the specifications made by the client on 04/01/2022	05/02/22

---

# 1 Introduction

## 1.1 Document Purpose

The purpose of this document is to provide a detailed description of the requirements for the swimiNIT mobile application. The document contains a general description, the performance, functional requirements and non-functional requirements of the application, a list of other relevant application attributes and usage scenarios. Additionally, this document provides a detailed use case analysis with a use case diagram.

## 1.2 Product Scope

The aim of the system is to streamline the process of registration and regulate the entry of swimmers into the NITC swimming pool. Additionally, this system aims to facilitate the admin and SPMs with the retrieval of the swimmer's visit history and payment information. A mailing feature will notify the swimmers after each visit, the start and end of each quarter, and pool maintenance. The admin will have the power to add, view and delete SPMs. They will also be able to download the quarterly visit and collection reports. We expect this application to be actively integrated into the day-to-day working of the NITC swimming pool to automate record-keeping and ensure a smooth and efficient operation.

## 1.3 Intended Audience and Document Overview

This document is intended for the admin and SPMs who will manage the swimming pool registration and entry system. The SPMs include the staff members who regulate the entry into and out of the pool, while the admin will manage the SPMs.

The next chapter, the overall description section of this document, gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, the specific requirements section of this document, is written primarily for the developers and describes the details of the functionality of the product in technical terms. Both sections of the document describe the same software product in its entirety but are intended for different audiences and thus use different languages.

## 1.4 Definitions, Acronyms and Abbreviations

S.L No.	Abbreviation/Acronym	Definition
1.	SPM	Swimming Pool Manager
2.	NITC	National Institute of Technology Calicut
3.	SDLC	Software Development Life Cycle

---

## **1.5 Document Conventions**

This document follows the IEEE formatting requirements.

## **1.6 References and Acknowledgments**

IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998

# **2 Overall Description**

## **2.1 Product Overview**

The primary objective of swimiNIT is to seamlessly regulate the entry of swimmers into the NITC swimming pool. As of now, the visitors to the swimming pool are tracked using a tedious log book. This application is a self-contained product. With the provision of five free trials for the student swimmers in each quarter, it has become crucial to track the visits. The retrieval of the swimmer's visit history, dues and payment information are features of high importance.

This application enables the SPMs to register swimmers by adding the required details. They will be able to record the entry of each swimmer. In the case of student swimmers, dues will be charged if the free trials have been exhausted. For faculty members or their dependents, they will be charged if they have visited the pool without making a payment. If dues have not been cleared in the previous quarter, the system will keep track of it. The SPMs will be able to edit the payment receipt information of swimmers.

Additionally, the application aims to implement a mailing feature to inform the swimmers of each visit, beginning and ending of each quarter and remaining trials in the case of student swimmers. Subsequently, the SPMs can view the swimmers who are in the pool and their duration in the pool is recorded.

The admin will be able to add and delete SPMs, edit details of swimmers, relay information regarding pool maintenance and download quarterly visit and collection report. The admin and SPMs can view the details and visit history of each swimmer and retrieve the swimmers who visited the pool on any given day/date range.

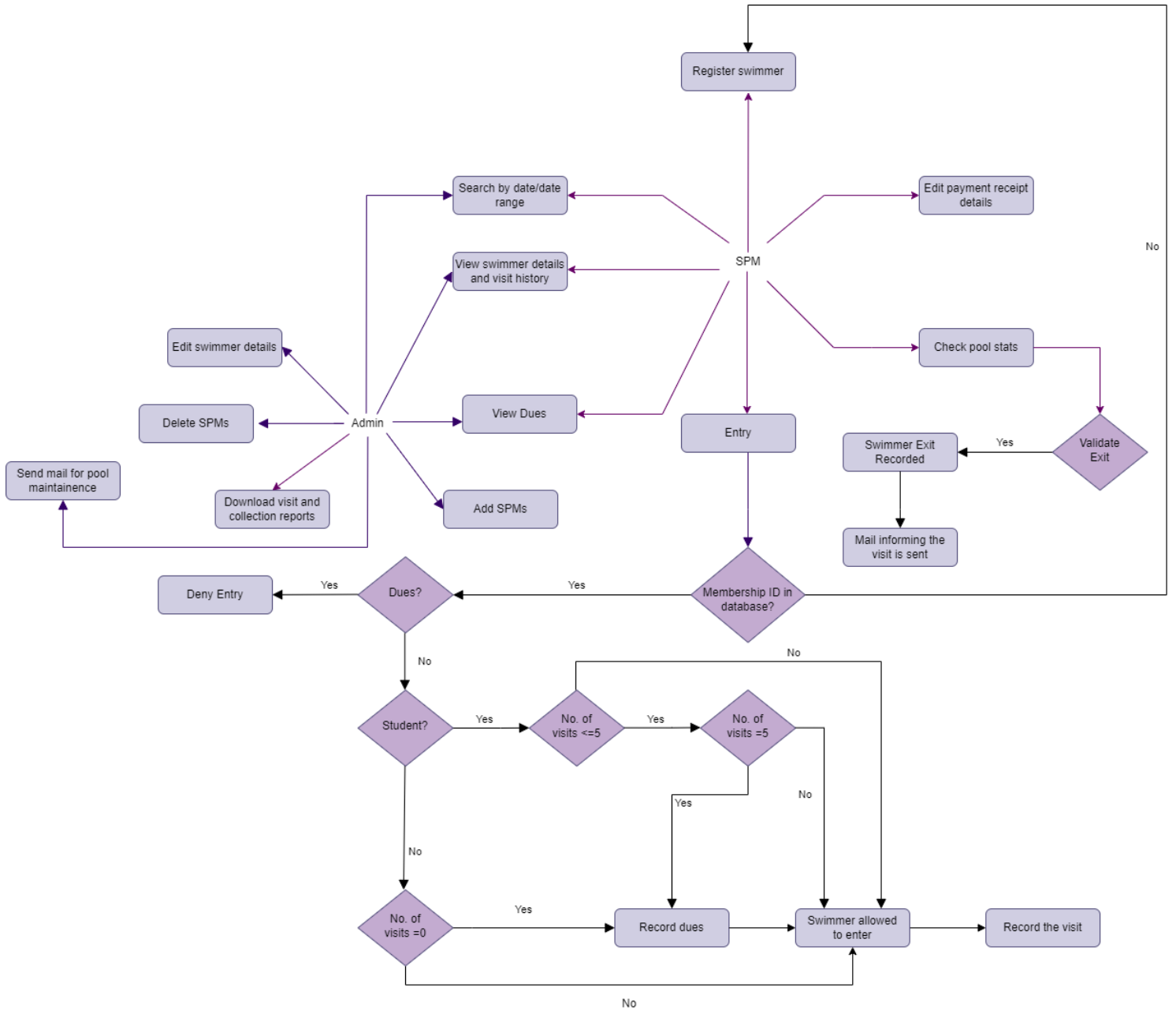


Fig. 1. High-level description of the product

---

## 2.2 Product Functionality

The swimiNIT App allows the SPMs to

- Register a swimmer by filling in the required information
- Mark the entry/exit of a swimmer into the pool
- View the number of swimmers currently in the pool, the start-time, end-time and duration they've spent there (Pool Status)
- View the history of a day (number of swimmers and time) in the swimming pool
- View all the swimmers with pending dues
- View the details of a swimmer by searching for a membership ID
- Edit the payment receipt details once a swimmer has made a payment

The app allows the admin to

- Create or delete SPM accounts
- View all SPMs
- View and download the history of a day (number of swimmers and time) in the swimming pool
- View and download the history (number of visits and date) of a swimmer in the pool along with any pending dues
- Edit the details of a swimmer
- View the details of a student by searching with the membership ID
- Send a mail to all swimmers regarding the status of the pool
- Generate and download quarterly and collection reports

## 2.3 Design and Implementation Constraints

**Database:** Cloud Firebase will be utilised to store the data. If only 1024MB of memory would be required to store the records, the free-tier of Cloud Firebase can be availed. Additionally, assuming that within the requirements, only 20,000 reads and writes are performed per day, the free-tier would suffice. If it goes past these specifications, the next tier has to be used with an additional cost of \$0.108 (approximately Rs 8.1).

**Hardware constraints:** A smartphone with Android OS version 6 or above. Minimum 1 GB RAM to ensure seamless run of the application.

**Internet:** The smartphone must be connected to the internet with adequate bandwidth to connect to the Cloud database.

**Design conventions and standards:** This application development process will follow the principles specified by the SDLC.

**Development Environment:** The development environment used for the application will be Android Studio/VS Code.

---

**Programming Standards:** Effective Dart programming styles as per official documentation will be adhered to while developing the application. Consistent naming, ordering and formatting will ensure that the application is maintainable.

## **2.4 Assumptions and Dependencies**

- The application will be provided directly to the admin and SPMs and will not be available for download on any public platform.
- There will only be one admin.
- There will be an uninterrupted internet connection.

# **3 Specific Requirements**

## **3.1 External Interface Requirements**

### **3.1.2 Hardware Interfaces**

There must be a smartphone with a minimum of 1 GB RAM to ensure a seamless run of the application.

### **3.1.3 Software Interfaces**

The smartphone's Operating system must be Android OS version 6 or above.

## **3.2 Functional Requirements**

F1: The system shall allow the admin and SPMs to log in to the system after verifying the login credentials.

F2: The system shall allow the SPMs to add the swimmer to the database after registration.

F3: The system shall provide an interface for the SPMs to validate and record the swimmer's entry to the pool by verifying their registration, payment, and/or the number of free trials.

F4: The system shall allow the SPMs to remove the swimmer from the pool status page and record the swimmer's exit from the pool.

F5: The system shall allow the admin and SPMs to search for the list of swimmer visits to the pool on a specific day or a given date range.

F6: The system shall allow the SPMs to view the pool usage history and details of a swimmer other than the contact information by searching for them using their membership ID.

F7: The system shall allow the admin to view the details of a swimmer and pool usage history by searching for them using their membership ID



---

F8: The system shall provide an interface to the SPMs that displays the entry time and time elapsed in the pool of each swimmer.

F9: The system shall mail the swimmer to update them of their daily visits, quarter and payment reminders.

F10: The system shall allow the admin to mail the swimmer regarding pool announcements.

F11: The system shall allow the SPMs to edit the swimmer's payment receipt details after searching by their membership ID.

F12: The system shall allow the admin to edit all the swimmer's details other than the membership ID after searching by their membership ID.

F13: The system shall display the pending dues of the swimmers to the admin and SPMs.

F14: The system shall allow the admin to view all the SPM accounts.

F15: The system shall allow the admin to create an SPM account.

F16: The system shall allow the admin to delete an SPM account.

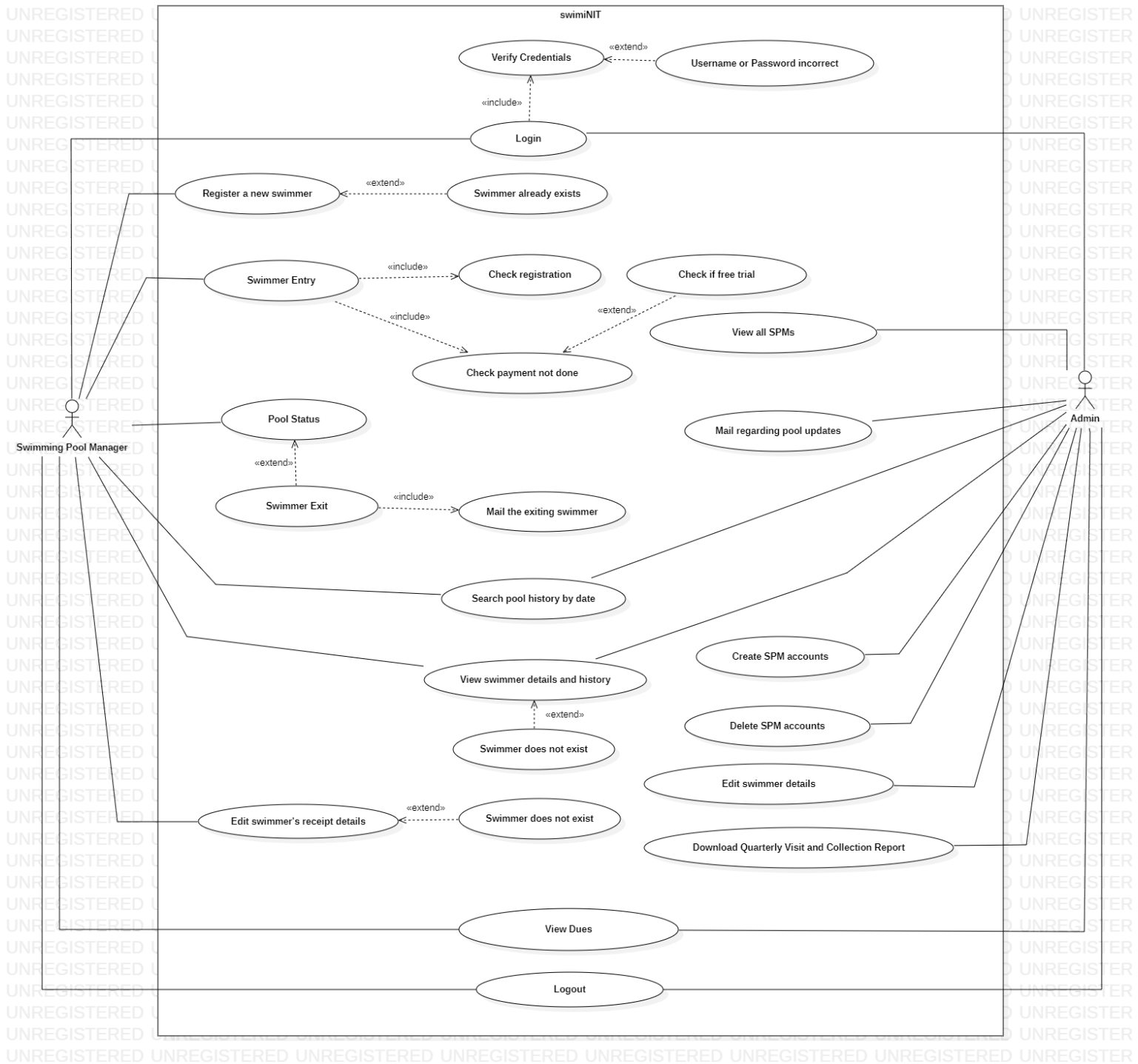
F17: The system shall allow the admin to download a list of swimmers who used the swimming pool in a specified date range.

F18: The system shall allow the admin to generate and download quarterly visit and collection reports.

F19: The system shall send a mail to the swimmer once they have been successfully registered.

F20: The system shall allow the admin or SPM to log out.

### 3.3 Use Case Model



---

### F3.3.1 Use Case #1 (Login – U1)

**Author** – Joseph Mani Jacob Mani

**Purpose** – To allow the user (admin or SPM) to log in to the system using valid login credentials.

**Requirements Traceability** – F1

**Priority** – High

**Preconditions** – None

**Post conditions** – The user will be successfully logged into the system.

**Actors** – Admin, SPM

**Extends** – None

#### Flow of Events

- **Basic Flow:**

Actor's Action	System's Response
The user enters the login credentials (The admin's credentials which are provided by the application developer and the SPM's credentials which are provided by the admin) and clicks the 'Submit' button.	The system verifies the entered login credentials with the database and logs the user into the system.

- **Alternative Flow:** None

- **Exceptions:**

Actor's Action	System's Response
The user enters incorrect login credentials and clicks the 'Submit' button.	The system displays an error message stating that the entered user ID or password is wrong.

**Includes** – None

**Notes/Issues** – None

---

### 3.3.2 Use Case #2 (Register a new swimmer – U2)

**Author** – Akshay Kuttikkattuparambil Biju

**Purpose** – To register a new swimmer into the system.

**Requirements Traceability** – F2, F19

**Priority** – High

**Preconditions** – The SPM should be logged into the system.

**Post conditions** – The swimmer will be successfully registered into the system.

**Actors** – SPM

**Extends** – None

#### Flow of Events

- **Basic Flow:**

Actor's Action	System's Response
The SPM enters the details of the swimmer and selects the 'Pay Now' option.	For student swimmers, the amount to be paid will be prefilled.  For non-student swimmers, the amount to be paid will not be prefilled.
The SPM enters the payment receipt details. In the case of non-student swimmers, the SPM enters the amount paid as well.  The SPM clicks the 'Submit' button.	The system displays a message stating that the swimmer has been successfully registered into the system.  An email is sent to the swimmer stating that they have been registered.

- **Alternative Flow:**

Actor's Action	System's Response
The SPM enters the details of the swimmer and selects the 'Pay Later' option.	The amount to be paid will not be prefilled.

The SPM clicks the 'Submit' button.	The system displays a message stating that the swimmer has been successfully registered into the system. Dues will be added for non-student swimmers.
-------------------------------------	---

- **Exceptions:**

Actor's Action	System's Response
The SPM enters the details of the swimmer and clicks the 'Submit' button.	The system displays an error message stating that the swimmer is already registered.

**Includes** – None

**Notes/Issues** – None

### 3.3.3 Use Case #3 (Swimmer Entry – U3)

**Author** – Varun Chittezhath Anilkumar

**Purpose** – To provide an interface to validate and record the swimmer's entry into the pool.

**Requirements Traceability** – F3

**Priority** – High

**Preconditions** – The SPM should be logged into the system.

**Post conditions** – The swimmer's entry into the pool will be successfully recorded.

**Actors** – SPM

**Extends** – None

**Flow of Events**

- **Basic Flow:**

Actor's Action	System's Response
----------------	-------------------

The user enters the membership ID of the swimmer and clicks the 'Search' button.	The system displays the details of the swimmer if the swimmer is registered in the system.
The user presses the 'Enter' button to record the entry of the swimmer into the pool.	<ol style="list-style-type: none"> <li>1. For student swimmers, the system records the swimmer's entry into the pool if the swimmer has no dues or has free sessions left in their name.</li> <li>2. For student swimmers, the system records the swimmer's entry into the pool and records the dues if they haven't paid and has no free sessions left.</li> <li>3. For non-student swimmers, the system records the swimmer's entry into the pool and records the dues if they haven't paid.</li> </ol>

- **Alternative Flow:** None
- **Exceptions:**

Actor's Action	System's Response
The user enters the membership ID of the swimmer and clicks the 'Search' button.	The system displays an error message stating that the swimmer does not exist if the swimmer is not registered in the system.
The user presses the 'Enter' button to record the entry of the swimmer.	<p>The system displays an error message if:</p> <ol style="list-style-type: none"> <li>1. The student swimmer has exhausted the five free sessions and has used the pool for the sixth time but hasn't cleared their dues before the seventh time of visit.</li> <li>2. The non-student swimmer has used the pool for the first time but hasn't cleared their dues before the next visit.</li> <li>3. The swimmer tries to use the pool in the next quarter without clearing the dues of the previous quarter.</li> </ol>

**Includes –** None

---

**Notes/Issues** – None

### 3.3.4 Use Case #4 (Swimmer Exit – U4)

**Author** – Varun Chittezhath Anilkumar

**Purpose** – To remove the swimmer from the pool status page and record the swimmer's exit from the pool.

**Requirements Traceability** – F4, F9

**Priority** – High

**Preconditions** –

- The SPM should be logged into the system.
- The swimmer should be registered in the system.

**Post conditions** – The swimmer's exit from the pool will be successfully recorded.

**Actors** – SPM

**Extends** – None

**Flow of Events**

- **Basic Flow:**

Actor's Action	System's Response
The user clicks the 'Exit User' button against the swimmer to be exited from the pool.	The system asks for a confirmation, 'Are you sure you want to exit the swimmer?'.
The user clicks 'Yes.'	The system records the swimmer's exit from the pool successfully, and a mail will be sent to their email ID automatically regarding their visit. This mail will also specify the dues/number of trials left, if any.

- **Alternative Flow:** None

- **Exceptions:**

Actor's Action	System's Response
----------------	-------------------

The user clicks the 'Exit User' button against the swimmer to be exited from the pool.	The system asks for a confirmation, 'Are you sure you want to exit the swimmer?'.
The user clicks 'No.'	The system redirects the user back to the pool status page.

**Includes** – None

**Notes/Issues** – None

### 3.3.5 Use Case #5 (Search pool history by date – U5)

**Author** – Pavithra Rajan

**Purpose** – To search for the list of swimmers who visited the pool on a specific date or date range.

**Requirements Traceability** – F5, F17

**Priority** – High

**Preconditions** –

- The admin and SPM should be logged into the system.
- The swimmer should be registered into the system, and each visit must be recorded.

**Post conditions** – The swimmer details related to the particular search will be displayed.

**Actors** – Admin, SPM

**Extends** – None

**Flow of Events**

- **Basic Flow:**

Actor's Action	System's Response
The user enters the 'from' and 'to' dates.	The system displays the membership ID, date of visit and start-end time of all the swimmers that visited the pool in the given date range.
The admin clicks the 'Download' button.	A report containing the details of the



	swimmers will be downloaded.
--	------------------------------

- **Alternative Flow:**

Actor's Action	System's Response
The user selects the 'same as from' option.	The system displays the membership ID, date of visit and start-end time of all the swimmers that visited the pool on that specific date.
The admin clicks the 'Download' button.	A report containing the details of the swimmers will be downloaded.

- **Exceptions:** None

**Includes** – None

**Notes/Issues** – None

### 3.3.6 Use Case #6 (View swimmer details and history – U6)

**Author** – Joseph Mani Jacob Mani

**Purpose** – To view the details of a swimmer and pool usage history by searching for them using their membership ID.

**Requirements Traceability** – F6, F7

**Priority** – Medium

**Preconditions** –

- The admin or SPM should be logged into the system.
- The swimmer should be registered into the system, and each visit must be recorded.

**Post conditions** – The swimmer's details and visit history will be displayed.

**Actors** – Admin, SPM

**Extends** – None

**Flow of Events**

- **Basic Flow:**

---

**a. For the admin:**

<b>Actor's Action</b>	<b>System's Response</b>
The admin enters the 'Membership ID' of the swimmer to be searched.	The system displays the swimmer's details such as the name, role (student or faculty), membership ID, contact no.1, contact no.2, email ID, dues, receipt ID, date of payment, the amount paid and the pool usage history details such as no. of visits, date of visit, time of entry and exit from the pool.

**b. For the SPM:**

<b>Actor's Action</b>	<b>System's Response</b>
The SPM enters the 'Membership ID' of the swimmer to be searched.	The system displays the swimmer's details such as the name, role (student or faculty), membership ID, dues, receipt ID, date of payment, the amount paid and pool usage history details such as the no. of visits, date of visit, time of entry and exit from the pool.

- **Alternative Flow:** None
- **Exceptions:**

<b>Actor's Action</b>	<b>System's Response</b>
The user enters the 'Membership ID' of the swimmer to be searched.	The system displays an error message stating that the entered membership ID is invalid.

**Includes –** None

**Notes/Issues –** None

### **3.3.7 Use Case #7 (Mail regarding pool updates – U7)**

**Author –** Abin Gigo Joseph

---

**Purpose** – To send an email to the swimmer regarding pool updates.

**Requirements Traceability** – F10

**Priority** – Low

**Preconditions** –

- The admin should be logged into the system.
- The swimmer should be registered into the system.

**Post conditions** – The swimmer will receive an email regarding pool updates.

**Actors** – Admin

**Extends** – None

**Flow of Events**

- **Basic Flow:**

Actor's Action	System's Response
The user clicks the 'Send Mail' button in the navigation bar.	The system opens the user's default email app with the 'To' textbox already prefilled with the Email IDs of all the swimmers registered in the system.

- **Alternative Flow:** None
- **Exceptions:** None

**Includes** – None

**Notes/Issues** – None

### 3.3.8 Use Case #8 (Edit swimmer details – U8)

**Author** – Akshay Kuttikkattuparambil Biju

**Purpose** – To edit swimmer details.

**Requirements Traceability** – F12

---

**Priority** – Medium

**Preconditions** – The admin should be logged into the system.

**Post conditions** – The swimmer's details will be updated.

**Actors** – Admin

**Extends** – None

**Flow of Events**

- **Basic Flow:**

Actor's Action	System's Response
The admin enters the membership ID of the swimmer whose details are to be edited.	The system displays the swimmer's details.
The admin edits the swimmer's details such as name, role, email ID, contact no.1, contact no.2 and clicks the 'Submit' button.	The system updates the swimmer's details.

- **Alternative Flow:** None

- **Exceptions:**

Actor's Action	System's Response
The admin enters the membership ID of the swimmer whose details are to be edited.	The system displays an error message stating that the swimmer does not exist.

**Includes** – None

**Notes/Issues** – None

### 3.3.9 Use Case #9 (Pool Status – U9)

**Author** – Lenoah Chacko

**Purpose** – To view the swimmers who are currently in the pool and how much time elapsed since they had checked in.

---

**Requirements Traceability – F8****Priority – High****Preconditions –**

- The SPM should be logged into the system.
- The swimmer's entry to the pool should be recorded.

**Post conditions –** The details of the swimmers in the pool and the time elapsed since their entry will be displayed to the SPM.

**Actors – SPM****Extends – U4****Flow of Events**

- **Basic Flow:**

Actor's Action	System's Response
The user clicks on 'Pool Status' in the navigation bar.	The system displays the names of all the swimmers currently in the pool and the time elapsed since their entry. The names will be displayed in descending order of the duration spent in the pool.

- **Alternative Flow:**

- **Exceptions:**

Actor's Action	System's Response
The user clicks on 'Pool Status' in the navigation bar.	The system displays that there are no swimmers in the pool if the pool is empty or closed.

**Includes – None****Notes/Issues – None**

---

### 3.3.10 Use Case #10 (View Dues – U10)

**Author** – Abin Gigo Joseph

**Purpose** – To view the dues of the swimmers.

**Requirements Traceability** – F13

**Priority** – High

**Preconditions** – The admin and SPM should be logged into the system.

**Post conditions** – The swimmer's dues will be displayed.

**Actors** – Admin, SPM

**Extends** – None

#### Flow of Events

- **Basic Flow:**

Actor's Action	System's Response
The user clicks on 'Pending Dues' in the navigation bar.	The system displays the dues of the swimmer.

- **Alternative Flow:** None

- **Exceptions:** None

**Includes** – None

**Notes/Issues** – None

### 3.3.11 Use Case #11 (View all SPMs – U11)

**Author** – Varun Chittezath Anilkumar

**Purpose** – To view the list of all the SPMs.

**Requirements Traceability** – F14

**Priority** – Medium

---

**Preconditions** – The admin should be logged into the system.

**Post conditions** – The list of all the SPMs will be displayed.

**Actors** – Admin

**Extends** – None

**Flow of Events**

- **Basic Flow:**

Actor's Action	System's Response
The user clicks on 'Pool Managers' in the navigation bar.	The system displays the list of all the SPMs.

- **Alternative Flow:** None

- **Exceptions:**

Actor's Action	System's Response
The user clicks on 'Pool Managers' in the navigation bar.	The system displays an error message stating that there are no SPMs added.

**Includes** – None

**Notes/Issues** – None

### 3.3.12 Use Case #12 (Create SPM account – U12)

**Author** – Joseph Mani Jacob Mani

**Purpose** – To create a new SPM account.

**Requirements Traceability** – F15

**Priority** – High

**Preconditions** – The admin should be logged into the system.

---

**Post conditions** – The SPM's account is created.

**Actors** – Admin

**Extends** – None

**Flow of Events**

- **Basic Flow:**

Actor's Action	System's Response
The user clicks on the 'Add' button on the 'Pool Managers' page.	The system redirects the user to a new page to enter the details and create the credentials for the new SPM.
The user enters contact no.1 of the SPM as the user ID and sets a password for the new SPM account.	The system saves the entered credentials and creates a new SPM account.

- **Alternative Flow:** None

- **Exceptions:** None

**Includes** – None

**Notes/Issues** – None

### 3.3.13 Use Case #13 (Delete SPM account – U13)

**Author** – Lenoah Chacko

**Purpose** – To delete an SPM account.

**Requirements Traceability** – F16

**Priority** – Medium

**Preconditions** – The admin should be logged into the system.

**Post conditions** – The SPM's account is deleted.

**Actors** – Admin

**Extends** – None



---

**Flow of Events**

- **Basic Flow:**

Actor's Action	System's Response
The user clicks on the 'Delete' button against the SPM account to be deleted on the 'Pool Managers' page.	The system displays a message asking the user if he/she is sure to delete the SPM account.
The user clicks on the 'Confirm' button.	The system deletes the SPM account.

- **Alternative Flow:** None
- **Exceptions:** None

**Includes** – None

**Notes/Issues** – None

### **3.3.14 Use Case #14 (Download Quarterly Visit and Collection Report – U14)**

**Author** – Pavithra Rajan

**Purpose** – To download the quarterly visit and collection report

**Requirements Traceability** – F18

**Priority** – High

**Preconditions** – The admin should be logged into the system.

**Post conditions** – The quarterly visit and collection report will be downloaded to the admin's phone.

**Actors** – Admin

**Extends** – None

#### **Flow of Events**

- **Basic Flow:**

Actor's Action	System's Response
The admin clicks on 'Reports' in the navigation bar.	The report containing the quarterly visit and collection of the swimming pool will be downloaded to the admin's phone.

- **Alternative Flow:** None
- **Exceptions:** None

**Includes** – None

**Notes/Issues** – None

### 3.3.15 Use Case #15 (Edit swimmer's receipt details – U15)

**Author** – Akshay Kuttikkattuparambil Biju

**Purpose** – To edit the receipt details.

**Requirements Traceability** – F11

**Priority** – High

**Preconditions** –

- The SPM should be logged into the system.

**Post conditions** – The swimmer's details will be updated.

**Actors** – SPM

**Extends** – None

**Flow of Events**

- **Basic Flow:**

Actor's Action	System's Response
The user enters the membership ID of the swimmer whose receipt details are to be filled/edited.	The system displays the fields to be filled/edited.
The user fills in the receipt details of the swimmer such as receipt ID, date of payment and amount paid (for non-student	The system updates the receipt details.

---

swimmers).	
------------	--

- **Alternative Flow:** None
- **Exceptions:**

Actor's Action	System's Response
The user enters the membership ID of the swimmer whose details are to be filled/edited.	The system displays an error message stating that the swimmer does not exist.

**Includes** – None

**Notes/Issues** – None

### 3.3.16 Use Case #16 (Log out – U16)

**Author** – Abin Gigo Joseph

**Purpose** – To allow the user (admin or SPM) to log out of the system.

**Requirements Traceability** – F20

**Priority** – Medium

**Preconditions** – The user should be logged into the system

**Post conditions** – The user will be successfully logged out of the system.

**Actors** – Admin, SPM

**Extends** – None

#### Flow of Events

- **Basic Flow:**

Actor's Action	System's Response
The user clicks the 'Log out' button.	The system displays a message asking the user if he/she is sure if they want to log out.
The user clicks on the 'Confirm' button.	The system logs out the user.

- 
- **Alternative Flow:** None
  - **Exceptions:** None

**Includes –** None

**Notes/Issues –** None

## 4 Other Non-functional Requirements

### 4.1 Performance Requirements

- **System:** The application will run on all devices running Android 6 (Marshmallow) or later. The application will be responsive.
- **Response Time:** The application should load and be usable within 3 seconds. It should update the interface on interaction within 2 seconds.
- **Backup:** The application will back up the dues of the swimmers every two days.

### 4.2 Safety and Security Requirements

The application will be provided directly to the admin and the SPMs and will not be available for download on any public platform. Hence, there will be no vulnerabilities.

### 4.3 Software Quality Attributes

#### 4.3.1 Reliability

The application requires Internet connectivity. For fetching any data, connectivity is indispensable to the functioning of this application. With the assumption that there is an uninterrupted connection, this application is reliable.

#### 4.3.2 Correctness

The application will be correct in terms of adhering to the functional requirements and will exhibit all the necessary functionalities.

#### 4.3.3 Availability

The application will run seamlessly. Since the data is stored on the cloud, any failure in the device used by the user will not impact the data.

#### 4.3.4 Scalability

---

Though the free-tier does limit the number of data records, a nominal payment can scale the application to handle more records.

#### **4.3.5 Maintainability**

The development team will follow the best practices and adhere to the officially documented effective Dart techniques. In addition, software modularity will be followed throughout, making the application maintainable. An extensive maintenance document will be provided for future reference.

#### **4.3.6 Portability**

The application will be made using Flutter and will run on devices running Android 6.0 or higher.

#### **4.3.7 Extensibility**

The code for the application will be available, and extensions can easily be created as and when required.

#### **4.3.8 Reusability**

The components of this application are very simple and will likely not be reused.

#### **4.3.9 Usability**

The application will have a UI that will be simple to use and explain. New users will be able to use it with ease.

#### **4.3.10 Robustness**

The application will have robust features correctly implementing the above use cases and will handle exceptions when required.

## **5 Other Requirements**

There is no other requirement needed for this project.

## **Appendix A - Activity Log**

S. No.	Date	Time
--------	------	------

1.	01-02-2022	04:00 to 05:20 PM
2.	02-02-2022	09:30 to 11:30 PM
3.	02-02-2022	12:15 to 02:05 AM
4.	03-02-2022	12:00 to 02:30 AM
5.	04-02-2022	12:00 to 01:10 AM
7.	16-02-2022	08:00 to 09:30 PM
8.	17-02-2022	12:00 to 02:00 AM
9.	17-02-2022	03:00 to 11:25 PM

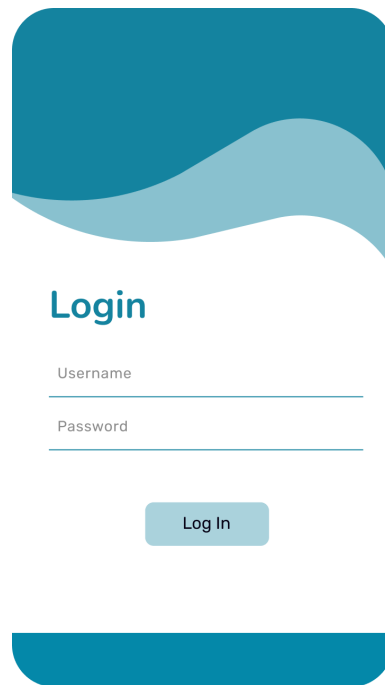
S. No.	Team Member	Contribution
1.	Lenoah Chacko	Intended Audience and Document Overview, Assumptions, Performance Requirements, Use Case #9, #13
2.	Pavithra Rajan	Product Scope, Design and Implementation constraints, Software Quality Attributes 4.3.1-4.3.5, Flow Chart, Use Case #5, #14
3.	Abin Gigo Joseph	Product Functionality, Software Quality Attributes 4.3.6-4.3.10, Use Case #7, #10, #16
4.	Joseph Mani Jacob Mani	Product Overview, Functional Requirements, Safety and Security Requirements, Use Case #1, #6, #12
5.	Akshay Kuttikkattuparambil Biju	Document Scope, Functional Requirements, Use Case #2, #8, #15
6.	Varun Chित्तेzhath Anilkumar	Hardware and Software Interfaces, Use Case #3, #4, #11

## Appendix B - User Interface

### User Interfaces for SPM and Admin

- Login Page [Corresponds to Use Case #1]

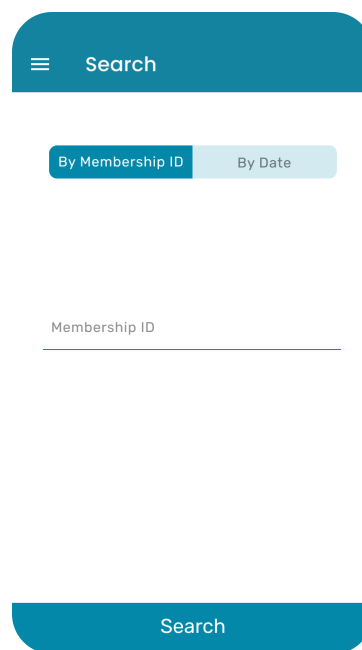
- 
- In this page, the user will enter the username and password and press the 'Log In' button to log in to the system.



The login form features a blue header with a white wavy graphic. Below the header, the word "Login" is displayed in a bold, blue font. Underneath, there are two input fields: "Username" and "Password", each with a blue underline. A blue "Log In" button is positioned below the password field. The form is set against a light blue background with a dark blue footer bar.

- **Search Page: “By Membership” active [Corresponds to Use Case #6]**


- This page is used to search the history of a specific swimmer’s visits. The user will enter the membership ID of the swimmer and pressing the ‘Search’ button will show the detailed history of the swimmer’s visits.



The search form has a blue header with a white hamburger menu icon and the word "Search". Below the header, there are two tabs: "By Membership ID" (active) and "By Date". Underneath the tabs, there is a "Membership ID" input field with a blue underline. A blue "Search" button is located at the bottom of the form. The form is set against a light blue background with a dark blue footer bar.

- **Search Page : “By Date” active [Corresponds to Use Case #5]**


- 
- This page is used to search the history of the visits on a particular day or a date range. The user will enter the 'From date' and select 'Same as from' to get the history of a particular day. To get the history of a particular date range, the user will have to enter the 'To date' as well. Pressing the 'Search' button will display the history.

 Search

By Membership ID


By Date

From date



☐ Same as from

To date



Search

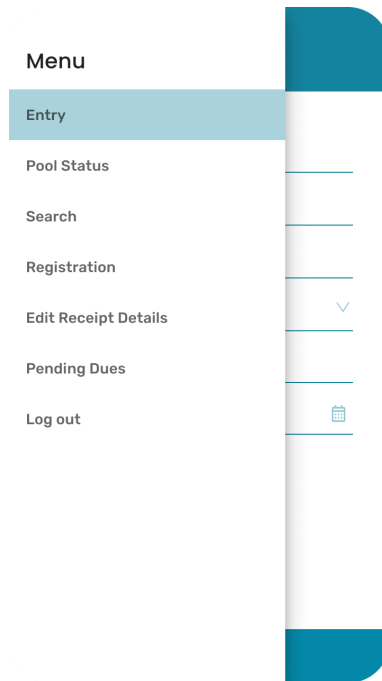


---

## User Interfaces for SPM

- **Navigation Menu**

- Navigation Menu shows seven options: Registration, Entry, Search, Status, Edit Receipt Details, Pending Dues and Log out.



- **Registration Page [Corresponds to Use Case #2]**

- The swimmers are registered in the app by the user. In this page, the user will enter the membership ID, name, email ID, role, contact number 1 and contact number 2. There will be two options displayed: Pay Now and Pay Later. If the SPM selects Pay Now, they will be prompted to fill in the receipt details such as receipt number, receipt date and payment amount. Else they can complete their registration and dues will be added for non-students directly. The swimmer will be registered once the 'Submit' button is pressed.

Registration

Membership ID

Name

Role

Email ID

Contact Number 1

Contact Number 2

Pay Now

Pay Later

Receipt ID

Payment Date

Quarterly fees:

Rs. 200

Money Paid

Submit

Registration

Membership ID

Name

Role

Email ID

Contact Number 1

Contact Number 2

Pay Now

Pay Later

Submit

- **Entry Page [Corresponds to Use Case #3]**

- This page is used to check whether the swimmer exists in the database or not by the user. The user enters the membership ID of the swimmer on entry and presses the 'Search' button.

Entry/Exit

Membership ID

Search

---

- **Entry Page for Invalid User**

- This error shows up if a particular swimmer does not exist in the database.

 Entry

Membership ID

Error! User does not exist!

Search

- **Entry Confirmation Page**

- This page confirms the entry of a swimmer into the swimming pool. Pressing the 'Enter' button by the user will record the swimmer's entry into the pool.

 Entry

Membership ID

B190657CS

Name

Lenoah Chacko

Role

Student

Email ID

B190657CS

Payment

No

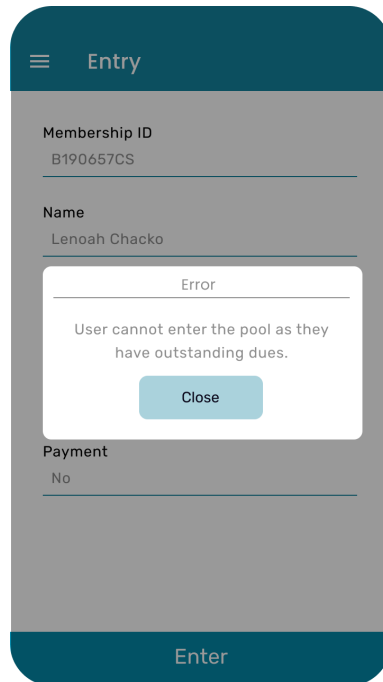
Enter

- 
- **Warning: If the swimmer has exhausted their free sessions**
    - This warning shows up if a student swimmer tries to enter for the 6th time and has not paid his/her dues yet or if a faculty swimmer tries to enter without completing payment.

The image is a screenshot of a mobile application interface. At the top, there is a dark teal header with a hamburger menu icon and the word "Entry". Below the header, the background is a light gray. There are three input fields with labels: "Membership ID" with the value "B190657CS", "Name" with the value "Lenoah Chacko", and "Payment" with the value "No". A white dialog box with a light gray border is centered on the screen. The dialog box has a title "Warning" and a message "As 5 free trials have been exhausted, dues will be incurred". At the bottom of the dialog box is a blue button labeled "Close". At the very bottom of the screen, there is a dark teal bar with the word "Enter" in white.

- **Error: If the swimmer has dues**
  - This error shows up if the swimmer already has dues

---




 Swimmer's entry has been recorded




- **History of a specific swimmer [Corresponds to Use Case #6]**

- This page shows the detailed history of the searched swimmer's visits. The details of the swimmer's visits shown are Date of visit, Time of entry and Time of exit. Pressing the 'Back' button will take the user back to the Search Page. In addition to

this, the role, membership ID, number of visits, dues and receipt details (if applicable) will be shown.

 User History

**Varun Anilkumar**  
Student  
B190621CS

7 visits  
Dues: Rs 0


Receipt ID: R-043657839    Date: 24-01-2022  
Amount Paid: Rs. 200

Date of visit	Time of entry	Time of exit
01-01-2022	16:00:36	16:53:20
02-01-2022	16:32:12	17:05:02
03-01-2022	17:02:21	18:01:22
04-01-2022	16:02:45	16:59:53
05-01-2022	18:04:33	18:56:46

Back

- **History of a specific day [Corresponds to Use Case #5]**

- This page displays the detailed history of the visits on a specific day. The details of the visits shown are Membership ID, Date of visit and Start-End time. Pressing the 'Back' button will take the user back to the Search Page.

 Results

Membership ID	Date of visit	Start-End time
B190621CS	01-01-2022	16:00:36-16:53:20
B190632CS	01-01-2022	17:20:17-17:43:29
EMP4321CS	01-01-2022	17:28:19-18:13:47

Back

- **History of a specific date range [Corresponds to Use Case #5]**

- This page displays the detailed history of the visits on a specific date range. The details of the visits shown are Membership ID, Date of visit and Start-End time. Pressing the 'Back' button will take the user back to the Search Page.

## Results

Membership ID	Date of visit	Start-End time
B190621CS	01-01-2022	16:00:36 16:53:20
B190632CS	01-01-2022	17:20:17 17:43:29
EMP4321CS	01-01-2022	17:28:19 18:13:47
DEP432101	02-01-2022	15:39:32 14:33:30
M200103CA	02-01-2022	16:00:36 16:53:20
P180140CS	03-01-2022	17:20:17 17:43:29
EMP4321CS	03-01-2022	17:28:19 18:13:47

Back

- **Pool Status Page [Corresponds to Use Case #9]**

- This page displays the status of all the swimmers currently in the pool. The details shown are Name, Membership ID, the time of entry and the time elapsed.

## Status

VA

**Varun Anilkumar**  
B190621CS

Entered at 16:36  
01:00:10

LC

**Lenoah Chacko**  
B190657CS

Entered at 16:44  
00:52:10

JM

**Joseph Mani**  
B190529CS

Entered at 17:04  
00:32:10

Exit

- **Swimmer Details Edit Page [Corresponds to Use Case #15]**

- This page allows the SPM to edit the receipt details of a particular swimmer. First, the SPM has to search for the swimmer using their Membership ID. If the swimmer doesn't exist, an error message will be displayed on the screen. Otherwise, the details of the swimmer such as the Name, Role, Email ID will be displayed for viewing and the receipt details such as the Receipt Number, Payment Date and Amount will be made available for editing.

≡ Edit receipt details

Membership ID

Membership ID

Search

≡ Edit receipt details

Membership ID  
B190621CS

Name  
Lenoah Chacko

Role  
Student

Email ID  
lenoah\_b190657cs@nitc.ac.in

Receipt ID  
R-043657839

Date of payment  
03-01-2022

Quarterly fees: Rs. 200

Money paid: Rs. 200

Submit

Back

## User Interfaces for Admin

- **Navigation Menu**

- Navigation Menu shows eight options: Pool Managers, Pool Status, Pending Dues, Search, Edit Swimmer Details, Send Mail, Reports and Log out.



---

## Menu

Pool Managers

Pool Status

Pending Dues

Search

Edit Swimmer Details

Send Mail

Reports

Log out

- **View all pool managers [Corresponds to Use Case #11, #13]**

- After clicking the 'Pool Manager', the admin can view all the SPMs that are present. If the admin clicks the add button, they will be redirected to a new screen to add the details of the SPMs. If the user clicks on the SPM, the widget expands to show a 'Delete' button. The admin has to click the button to delete the SPM. They will be asked for confirmation prior to this as well.

## Pool Managers

JD

John Doe

7902267026

JW

John Wick

8374612372

JC

John Cena


3737273721

Delete

Add


- **Edit Swimmer Details [Corresponds to Use Case #8]**

- After searching for a student, the admin will be able to edit the details of a student excluding the fees. On pressing the submit button, the swimmer's details will be updated.

 Edit details

Membership ID

Search

 Edit details

Membership ID  
B190621CS

Name  
Lenoah Chacko

Role  
Student

Email ID  
lenoah\_b190657cs@nitc.ac.in

Contact 1  
7902260071

Contact 2  
7902260071

Receipt ID  
R-043657839

Date of payment  
03-01-2022

Quarterly fees: Rs. 200

Money paid: Rs. 200

Submit

Back

- **View and download user history [Corresponds to Use Case #6]**

- After searching for a student, the admin will be able to view and download the details of all the student's visits.

≡

User History

VA

Varun Anilkumar

Student

B190621CS

varun\_b190621cs@nitc.ac.in

6285435321

7283412846

7 visits

Dues: Rs 0

Receipt ID:

R-043657839

Date:

24-01-2022

Amount Paid:

Rs. 200

Date of visit	Time of entry	Time of exit
01-01-2022	16:00:36	16:53:20
02-01-2022	16:32:12	17:05:02
03-01-2022	17:02:21	18:01:22
04-01-2022	16:02:45	16:59:53
05-01-2022	18:04:33	18:56:46

Download

Back

- **History of a specific day [Corresponds to Use Case #5]**

- This page displays the detailed history of the visits on a specific day. The details of the visits shown are Membership ID, Date of visit and Start-End time. Pressing the 'Back' button will take the admin back to the Search Page. The admin can download this result as well.

Results

Membership ID	Date of visit	Start-End time
B190621CS	01-01-2022	16:00:36-16:53:20
B190632CS	01-01-2022	17:20:17-17:43:29
EMP4321CS	01-01-2022	17:28:19-18:13:47

Download

Back

- History of a date range [Corresponds to Use Case #5]**
  - This page displays the detailed history of the visits on a specific date range. The details of the visits shown are Membership ID, Date of visit and Start-End time. Pressing the 'Back' button will take the admin back to the Search Page. The admin can download these results as well.

Results


Membership ID	Date of visit	Start-End time
B190621CS	01-01-2022	16:00:36-16:53:20
B190632CS	01-01-2022	17:20:17-17:43:29
EMP4321CS	01-01-2022	17:28:19-18:13:47
DEP432101	02-01-2022	15:39:32-14:33:30
M200103CA	02-01-2022	16:00:36-16:53:20
P180140CS	03-01-2022	17:20:17-17:43:29
EMP4321CS	03-01-2022	17:28:19-18:13:47

Download

Back

- View the pending dues [Corresponds to Use Case #10]**

The admin can view the pending dues of all the swimmers and download the results


 Pending Dues


Membership ID	Name	Dues
B190657CS	Lenoah Chacko	200
B190529CS	Joseph Mani	200
B190632CS	Pavithra Rajan	200
B190880CS	Abin Gigo	200
EMP1001D	Jimmy Jose	500


Download

Back

- **Generate and download quarterly visit and collection reports [Corresponds to Use Case #14]**
  - After clicking on the Reports option, the admin can generate a quarterly visit report or a collections report.

 Reports

  
Quarterly visit report

  
Collections report

Search

- **Add new pool managers [Corresponds to Use Case #12]**
  - After clicking on the add pool manager option, the admin can enter the details of the pool manager to be added. On clicking the Add button, the pool manager will be added to the app.

---

☰ Add Pool manager

Name

Contact 1

Contact 2

Password

Add

☰ Pool Manager



SPM has been added

Back

---

## Appendix C

The changes made are:

1. The **Product Overview** (Section 2.1) is made more descriptive encompassing the scope of the whole application.
2. The high-level description diagram was modified to incorporate the new user (the admin) of the app. Their functions have been added as well. This diagram accounts for the dues charged for the swimmers which were not mentioned in the earlier document. These have been added to the **Product Functionality** (Section 2.2) as well.
3. Since it was suggested that the copy of the receipt is not required, the requirement was removed from the **Design and Implementation Constraints** (Section 2.3)
4. **Hardware Interfaces** (Section 3.1.2): Since the feature of scanning QR code for renewal and entry and the upload of the copy of the receipt is removed, there is no camera requirement that was earlier specified.
5. **Functional Requirement** (Section 3.2) changes:
  - The exit of the swimmer will be done via the pool status page.
  - There is a new actor (admin) who can add/view/delete SPMs.
  - The admin will be able to get a quarterly visit and collection report.
  - Both the admin and SPMs can get the details of the swimmers that visited on a particular day/date range. Additionally, the admins can download this.
  - The admin can view and edit the details of the swimmer and the SPMs can only edit the payment receipt details and view all the details that the admin can except the contact numbers.
  - The admin will send the mail regarding pool updates instead of SPMs.
  - The admin and SPMs can log out of the system.
6. The **Use Case Model** (Section 3.3) has been modified to account for the aforementioned changes.
7. Use-Cases #10, #11, #12, #13, #14, #15, 16 have been added and other use cases have been modified in the cases where the new actor (admin) is involved.
8. Backup has been added in **Performance Requirements** (Section 4.1).