National Institute of Technology Calicut

swimiNIT

An android application for NITC's swimming pool.

Name	Roll number
Lenoah Chacko	B190657CS
Pavithra Rajan	B190632CS
Abin Gigo Joseph	B190880CS
Joseph Mani Jacob Mani	B190529CS
Akshay Kuttikkattuparambil Biju	B190803CS
Varun Chittezhath Anilkumar	B190621CS

Module Owner: Ms. Lekshmy P Chandran

Team: 05

Table of contents

Problem Statement and Feasibility Study

for

swimiNIT

Version 1.0

Prepared by

Team Number: 05

Lenoah Chacko	**B190657CS**
Pavithra Rajan	**B190632CS**
Abin Gigo Joseph	**B190880CS**
Joseph Mani Jacob Mani	**B190529CS**
Akshay Kuttikkattuparambil Biju	**B190803CS**
Varun Chittezhath Anilkumar	**B190621CS**

Project Owner: **Ms. Lekshmy P Chandran**

Course: **CS4096D Software Engineering Laboratory**

Date: **16/01/2022**

1 Problem Definition

This section will render a brief overview of the problem statement and its relevance. In addition to this, the present system that has been adopted has also been included.

1.1 Problem Statement

The client seeks a system that can regulate the entry of swimmers into the NITC swimming pool. With the provision of five free trials for the students each quarter, a framework to keep track on the trials availed becomes crucial. Additionally, retrieval of visit and payment information is expected.

1.2 Relevance

With the introduction of a new swimming pool system, some amount of automation is required to make things run smoothly. The new system requires that we keep a record of the number of times that a student swimmer has visited the swimming pool, as the first 5 visits for them are free for a particular quarter. Maintaining a record of this is very tedious. This is what our application aims to rectify.

1.3 Background and Existing System

The details of the swimmer such as the Membership ID, name and time of visit is recorded manually in a log book and the swimmer is notified at the end of the session upon verification of the duration in the pool. Entry to the pool is restricted on the days of pool maintenance. Since there were no free trials offered to the students, there was no necessity to keep track of it. There was no mailing feature available to notify the swimmers of each visit, the start and end of each quarter and pool maintenance.

2 Project Objectives

On completion of this project, the following objectives will be achieved:

- Develop a system to track the number of visits of each swimmer
- Store all the required information of the swimmer which can be viewed upon entering the Membership ID or scanning the QR code from their Membership ID card.
- Retrieval of visit and payment history of every swimmer
- Acquire visit history for a given date or range of dates
- Prevent a swimmer from swimming past the duration limit of 1 hour or an extra visit on a given day
- Updates sent to swimmers via mail after successful registration, end of free trials (applicable to students only), every visit to the pool, start and end of each quarter and pool maintenance

3 Feasibility Study

3.1.1 Proposed Solution

We plan to develop a mobile application for the client that will have the following features:

1. Admin Dashboard

The admin has an interactive dashboard that enables them to view relevant information at a single glance. The admin dashboard has the following sections-

a. Swimmer Registration

i. First Visit

Upon submission of the swimming pool membership form (for the student) or the payment receipt (for the faculty/staff and their dependents), the admin can register the swimmer by entering the relevant information.

ii. Subsequent Quarters

Upon completing the payment and submission of the receipt, the admin can **update** the **copy of the receipt** and set the '**payment'** to indicate that the swimmer has paid**.**

b. Record Visit

The staff verifies the Membership ID as per the diagram given below. Once the swimmer's details have been verified, the date and time of entry are recorded. When a swimmer leaves the pool, the time of exit is recorded.

c. View history of a specific swimmer

The user can check the detailed history of a particular swimmer's visits.
The details available will be the Membership ID, date and time and the total number of visits.
The user can also verify the payment status with the receipt if any.

d. View visits made to the pool on a day/date range.

The user can check the detailed history of the number of visits on a particular day or a date range.
The details presented will be the Membership IDs, date and time of the visits.

e. Additional Feature: Current Pool Stats

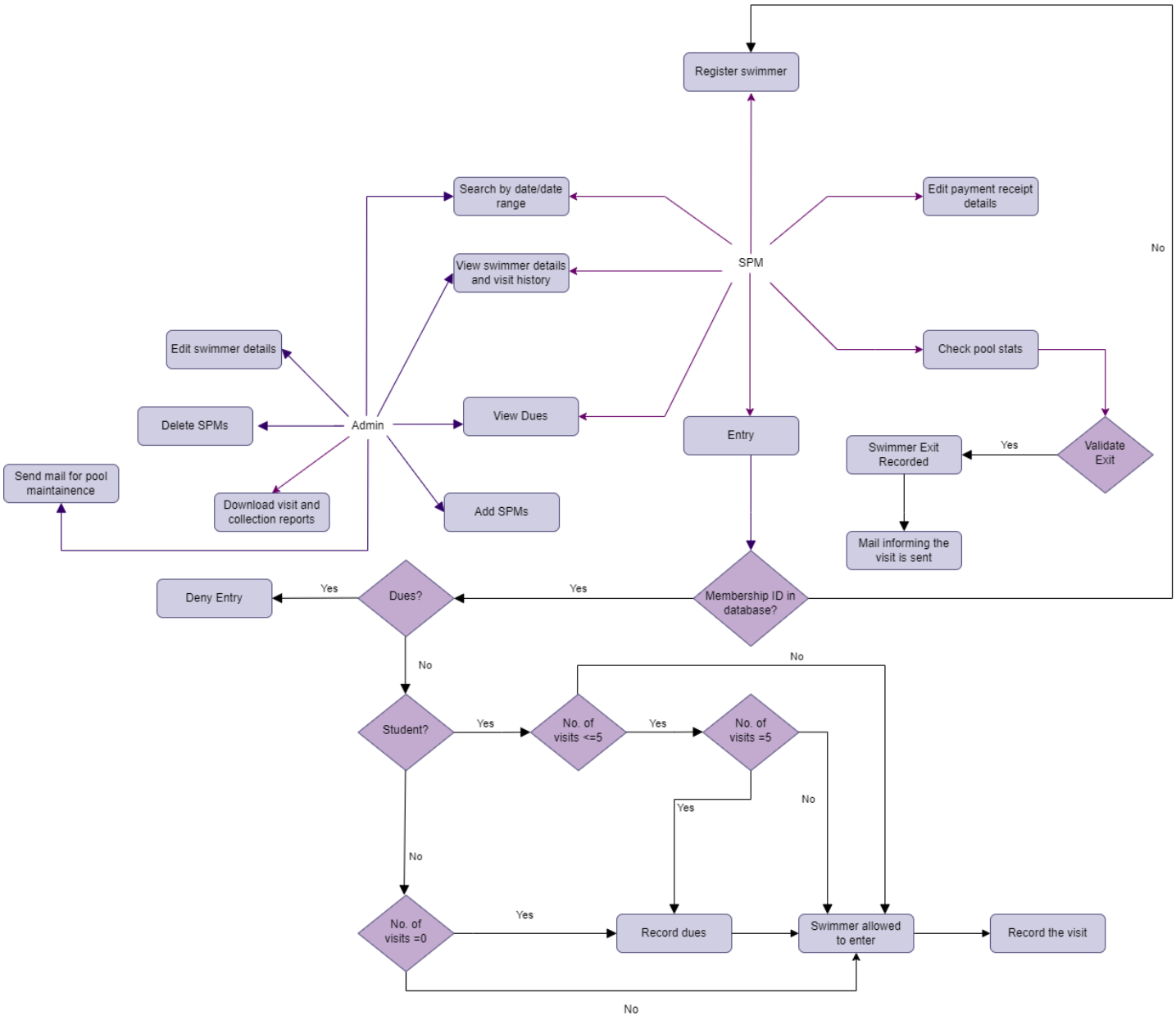
There will be a live dashboard that shows all the swimmers currently paying a visit to the pool.

The dashboard shows details such as the **membership ID, name, the time of entry, the time remaining** for a swimmer *(duration: 1 hour)* and a visual indicator showing when it's time for a swimmer to leave.
The indicator will be **green** initially. It will turn **yellow** when there are 10 minutes left for the swimmer's session to end and **red** when the session is over.
A notification will be issued when one or more swimmers' sessions are running out. There will be a single notification for multiple swimmers whose sessions are ending at the same time to prevent spamming of notifications.

2. Mailing Feature:

A mail will be sent out to the swimmers when the following events happen:

- Successful registration
- End of 5 trial attempts *(for the student)*
- Visit to the pool
- End of quarter reminder *(1 week before)*
- Start of quarter reminder *(when the quarter ends)*
- Pool maintenance



3.1.2 Economic Feasibility

As this will be a mobile application which has an exclusive user (the swimming pool staff), no cost for hosting it in Play Store will be incurred as the application file can be directly given to them. As the only multimedia data storage will be for storing the copy of the swimmer receipt during registration, the bandwidth requirement for the functioning of the system is minimal.

Since Cloud Firebase will be utilised to store the data, the free tier offers 1GiB data storage along with 20,000 data reads and writes per day which falls within the requirements. This is considering that only about 512 members register every quarter with each record (copy of receipt + relevant attributes) being under 2 MB. For every additional GiB an additional cost of $0.108 (approximately Rs. 8.1) will be charged.

Following freeware software standards, there will be no cost incurred from the user for the usage. This system will eliminate the extra effort in auditing the visits and payments of swimmers along with the creation of reports for each quarter as all of this will be automated.

Hence, it is evident that this project is economically feasible.

3.1.3 Technical Feasibility

The main technologies and tools that are associated with this project are:

- Flutter
- Android Studio
- Firebase
- Flask

Tools

- Git
- Trello
- app.diagrams.net
- Figma
- Postman

Each of the aforementioned technologies and tools are freely available and the technical skills required are achievable. Thereby, assuring that the product will be delivered within the stipulated time frame.

3.1.4 Risk Assessment

There is a risk of the users finding the application difficult to use when transitioning to the new system. A tutorial will be made to make this transition easier.

4 Conclusion

We can conclude that the project is feasible. We can go ahead while keeping storage control measures and ease of use in mind.

Appendix A - Activity Log

S. No.	Activity	Date
1.	Project Proposal discussion 1 - Feature discussion	06-01-2022, 12:26
2.	Project Proposal discussion 2 - Making the document	08-01-2022, 11:25

3.	App prototyping in Figma	08-01-2022, 20:40
4.	Application feature discussion	11-01-2022, 20:05
5.	Feasibility Report discussion 1 - Feasibility discussion	14-01-2022, 14:50
6.	Feasibility Report discussion 2 - Making the document	14-01-2022, 23:15

S. No.	Team Member	Contribution
1.	Lenoah Chacko	Relevance, Risk Assessment, Conclusion
2.	Pavithra Rajan	Record Visit Diagram, Economic Feasibility
3.	Abin Gigo Joseph	Problem Statement, Admin Dashboard (a-b)
4.	Joseph Mani Jacob Mani	Background and Existing System, Mailing Feature
5.	Akshay Kuttikkattuparambil Biju	Admin Dashboard (c-e)
6.	Varun Chittezhath Anilkumar	Project Objectives, Technical Feasibility

Software Requirements Specification

for

A5-swimiNIT

Version 1.1

Prepared by

Team Number: 05

Lenoah Chacko	**B190657CS**
Pavithra Rajan	**B190632CS**
Abin Gigo Joseph	**B190880CS**
Joseph Mani Jacob Mani	**B190529CS**
Akshay Kuttikkattuparambil Biju	**B190803CS**
Varun Chittezhath Anilkumar	**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.

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 reports. 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.

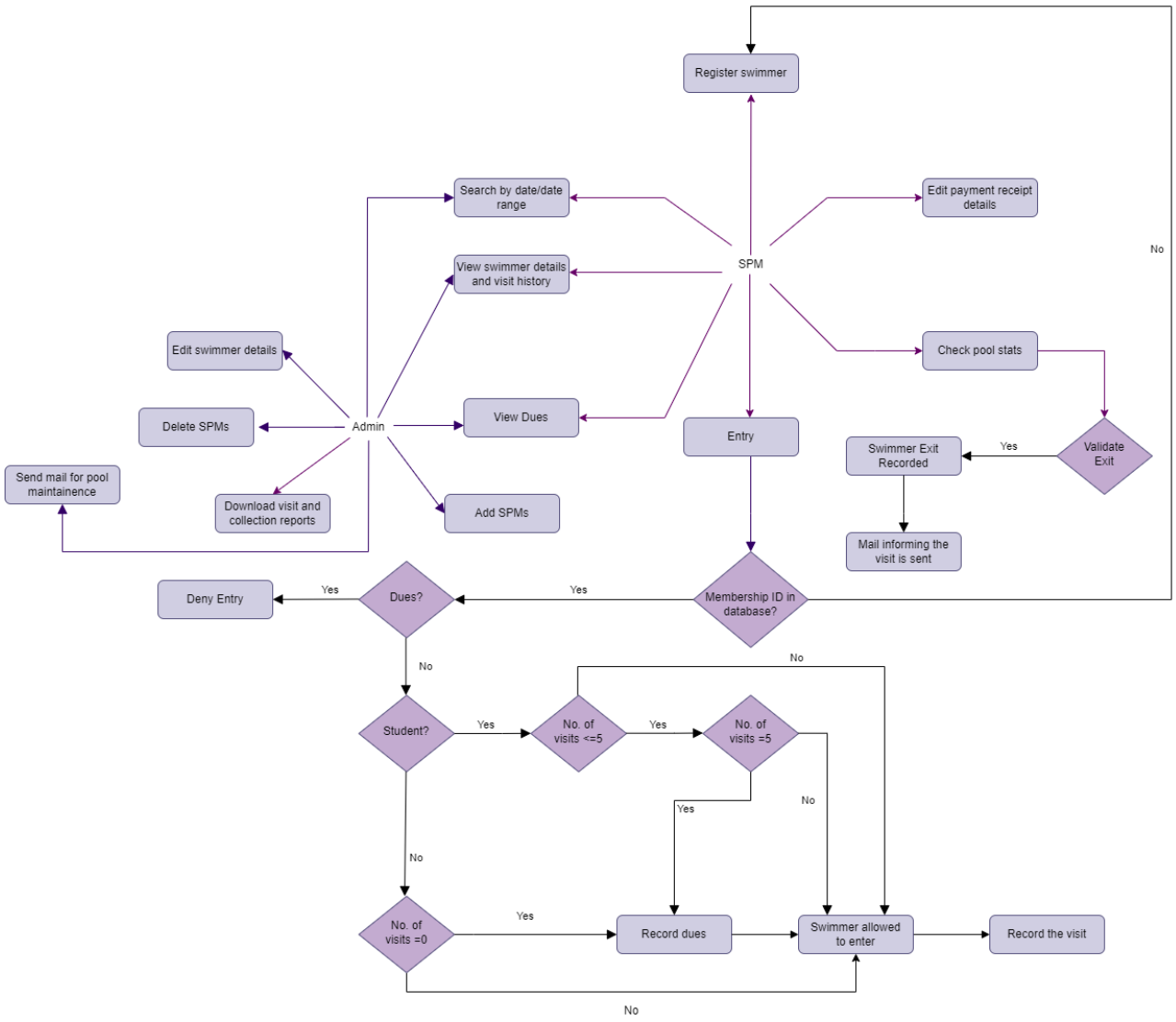


Fig. 1. High-level description of the product

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

3.1.1. Hardware Interfaces

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

3.1.2. 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.

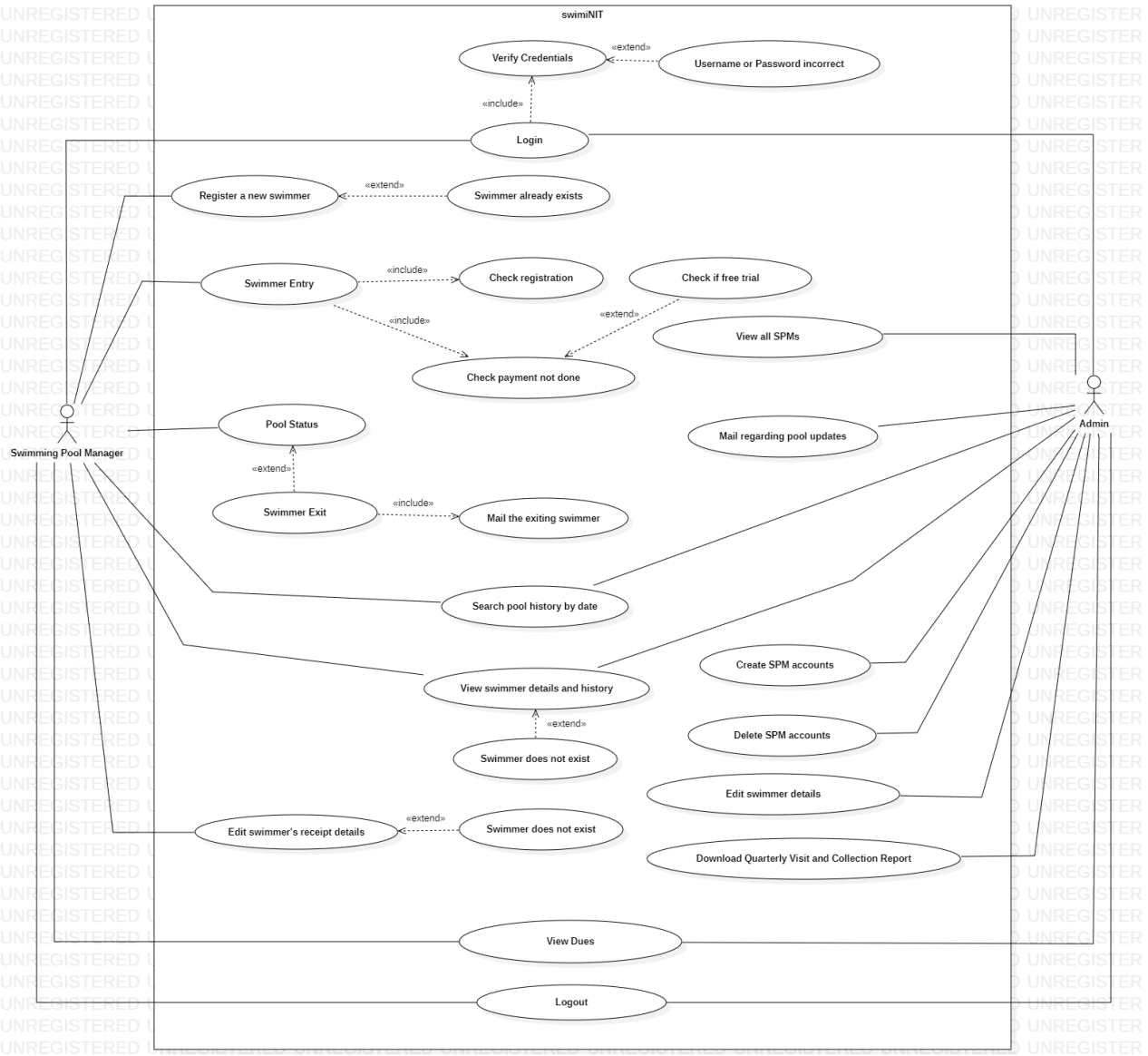
F17: The system shall allow the admin to download a list of swimmers who used the swimming pool in a specified date range.

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



3.3.1. 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.	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. 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. 3. For non-student swimmers, the system records the swimmer's entry into the pool and records the dues if they haven't paid.

- **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.	The system displays an error message if: 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. 2. The non-student swimmer has used the pool for the first time but hasn't cleared their dues before the next visit. 3. The swimmer tries to use the pool in the next quarter without clearing the dues of the previous quarter.

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 admin clicks the ‘Download' button.	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. 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:

Actor's Action	System's Response
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:

Actor's Action	System's Response
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:**

Actor's Action	System's Response
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:** None
- **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 Chittezhath 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 swimmers).	The system updates the receipt details.

- **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

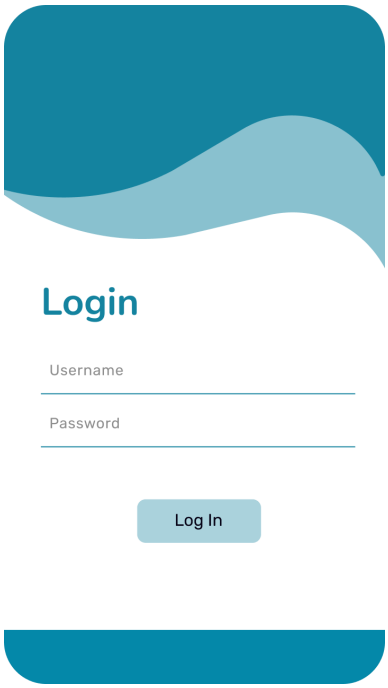
S. No.	Team Member	Contribution
1.	Lenoah Chacko	Intended Audience and Document Overview, Assumptions, Performance Requirements, Use Case #9, #13
2.	Pavithra Rajan	Product Scope, Design and Implementation constraints, Software Quality Attributes 4.3.1-4.3.5,

		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 Chittezhath Anilkumar	Hardware and Software Interfaces, Use Case #3, #4, #11

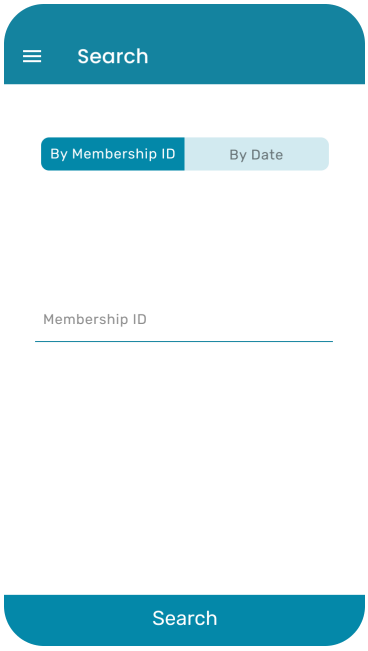
Appendix B - User Interface

User Interfaces for SPM and Admin

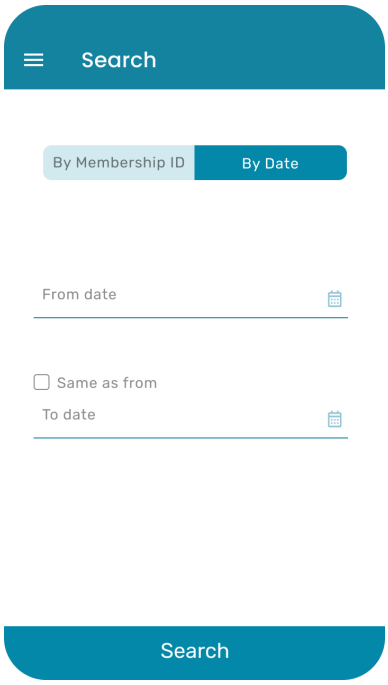
- **Login Page [Corresponds to Use Case #1]**
  In this page, the user will enter the username and password and press the 'Log In' button to log in to the system.



- **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.

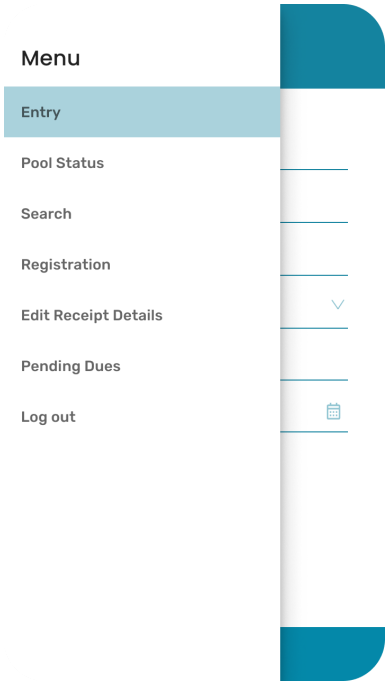


- **Search Page : “By Date” active [Corresponds to Use Case #5]**
  This page is used to search the history of the visits on a particular day or a date range. The user will enter the ‘From date’ and select ‘Same as from’ to get the history of a particular day. To get the history of a particular date range, the user will have to enter the ‘To date’ as well. Pressing the ‘Search’ button will display the history.

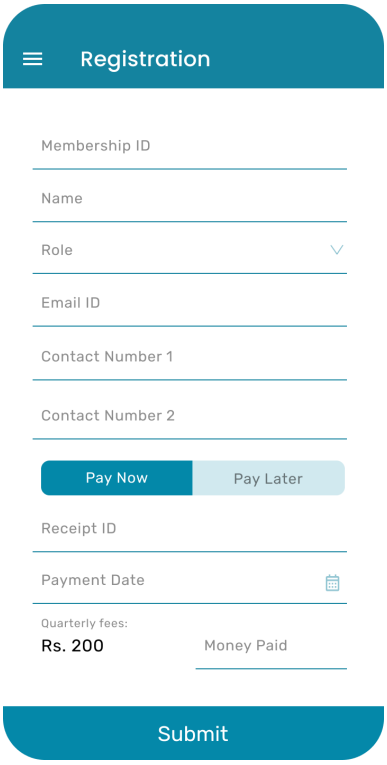


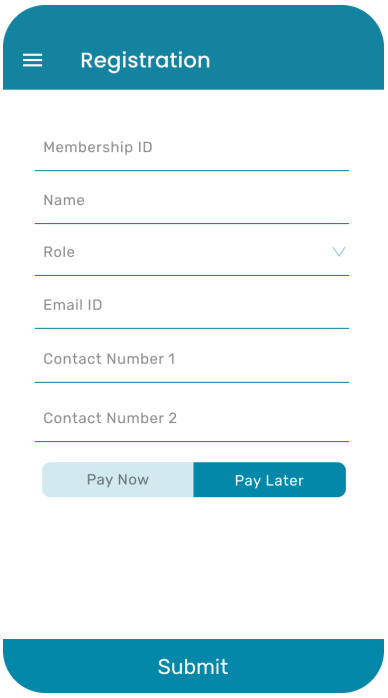
User Interfaces for SPM

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

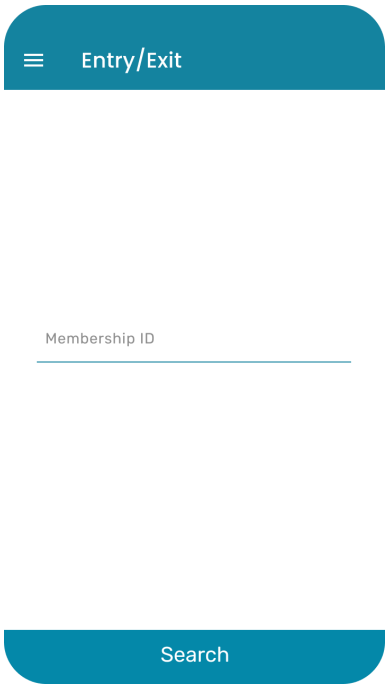


- **Registration Page [Corresponds to Use Case #2]**
  The swimmers are registered in the app by the user. In this page, the user will enter the membership ID, name, email ID, role, contact number 1 and contact number 2. There will be two options displayed: Pay Now and Pay Later. If the SPM selects Pay 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.

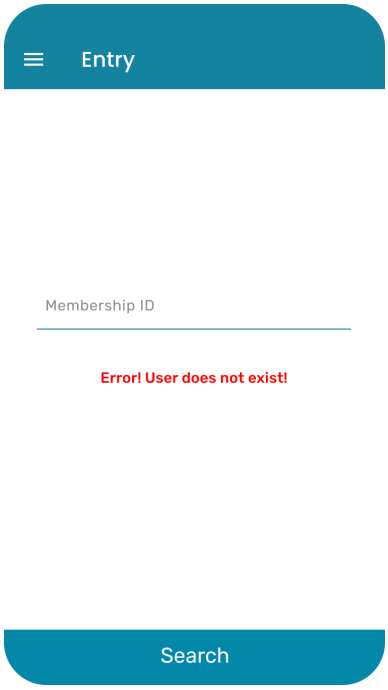




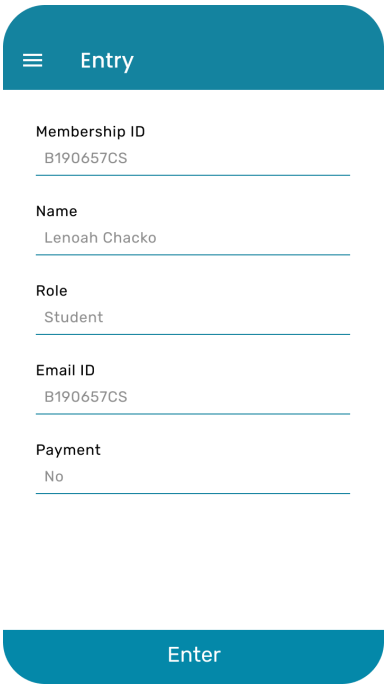
- **Entry Page [Corresponds to Use Case #3]**
  This page is used to check whether the swimmer exists in the database or not by the user. The user enters the membership ID of the swimmer on entry and presses the 'Search' button.



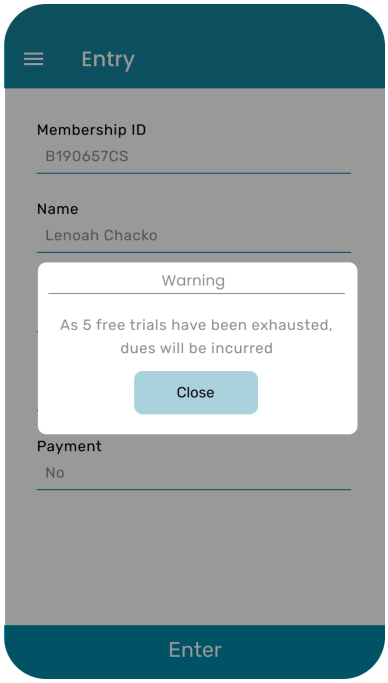
- **Entry Page for Invalid User**
  This error shows up if a particular swimmer does not exist in the database.



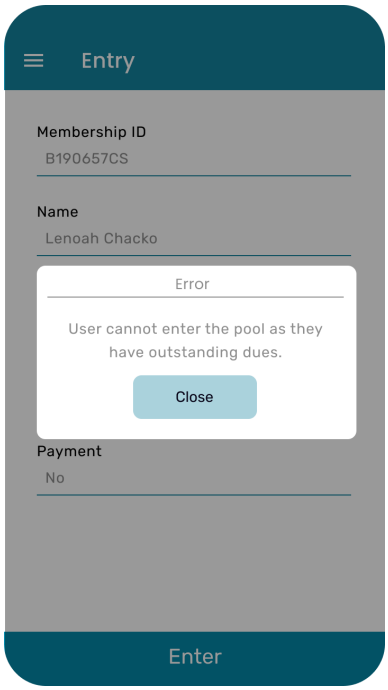
- **Entry Confirmation Page**
  This page confirms the entry of a swimmer into the swimming pool. Pressing the ‘Enter’ button by the user will record the swimmer’s entry into the pool.

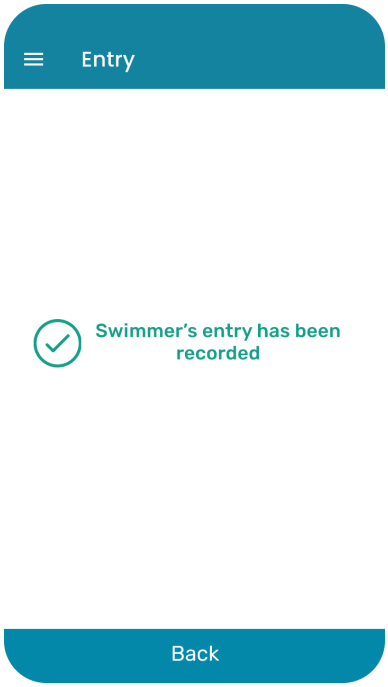


- **Warning: If the swimmer has exhausted their free sessions**
  This warning shows up if a student swimmer tries to enter for the 6th time and has not paid his/her dues yet or if a faculty swimmer tries to enter without completing payment.

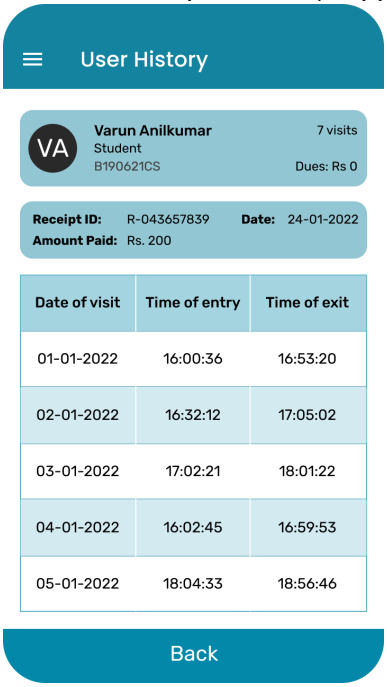


- **Error: If the swimmer has dues**
  This error shows up if the swimmer already has dues



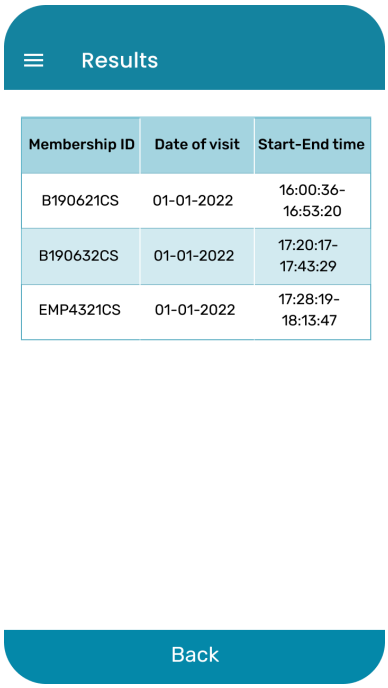


- **History of a specific swimmer [Corresponds to Use Case #6]**
  This page shows the detailed history of the searched swimmer's visits. The details of the swimmer's visits shown are Date of visit, Time of entry and Time of exit. Pressing the 'Back' button will take the user back to the Search Page. In addition to this, the role, membership ID, number of visits, dues and receipt details (if applicable) will be shown.



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

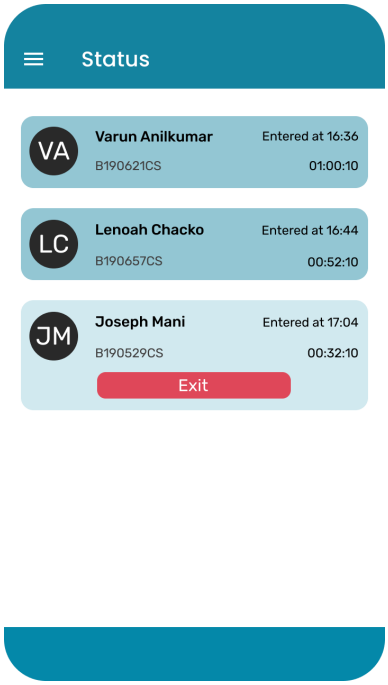
This page displays the detailed history of the visits on a specific day. The details of the visits shown are Membership ID, Date of visit and Start-End time. Pressing the ‘Back’ button will take the user back to the Search Page.



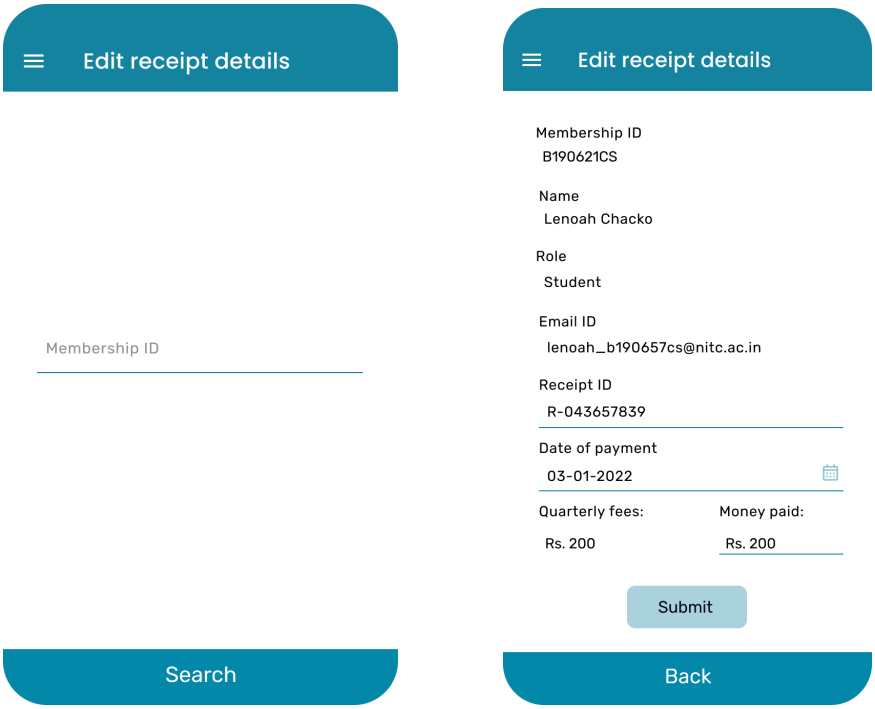
- **History of a specific date range [Corresponds to Use Case #5]**
  This page displays the detailed history of the visits on a specific date range. The details of the visits shown are Membership ID, Date of visit and Start-End time. Pressing the ‘Back’ button will take the user back to the Search Page.

- **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.

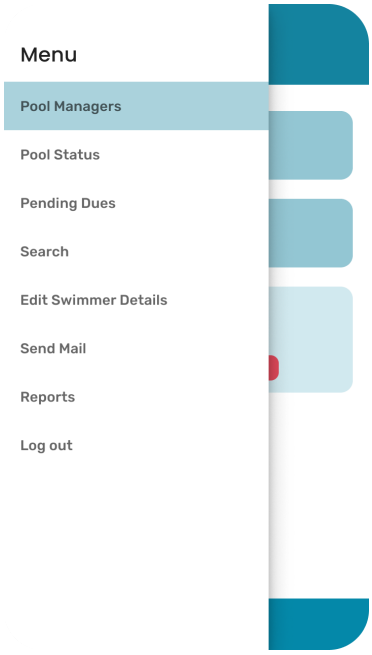


- **Swimmer Details Edit Page [Corresponds to Use Case #15]**
  This page allows the SPM to edit the receipt details of a particular swimmer. First, the SPM has to search for the swimmer using their Membership ID. If the swimmer doesn't exist, an error message will be displayed on the screen. Otherwise, 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.



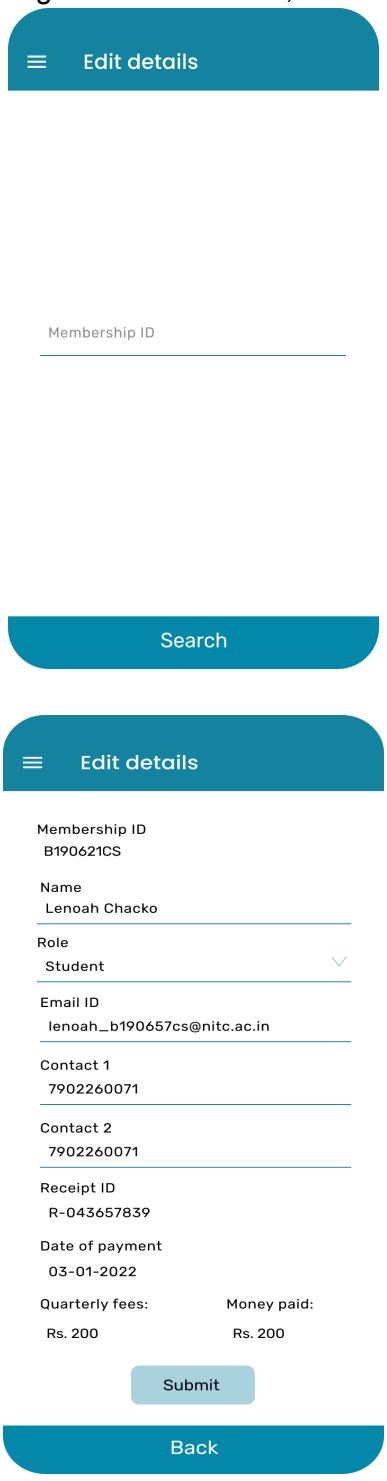
User Interfaces for Admin

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



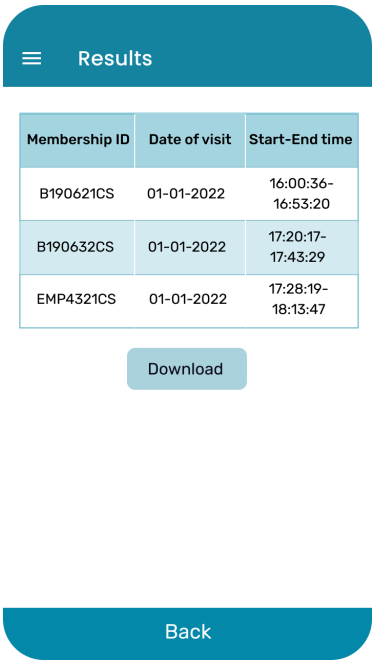
- **View all pool managers [Corresponds to Use Case #11, #13]**
  After clicking the ‘Pool Manager’, the admin can view all the SPMs that are present. If the admin clicks the add button, they will be redirected to a new screen to add the details of the SPMs. If the user clicks on the SPM, the widget 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.

- **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.

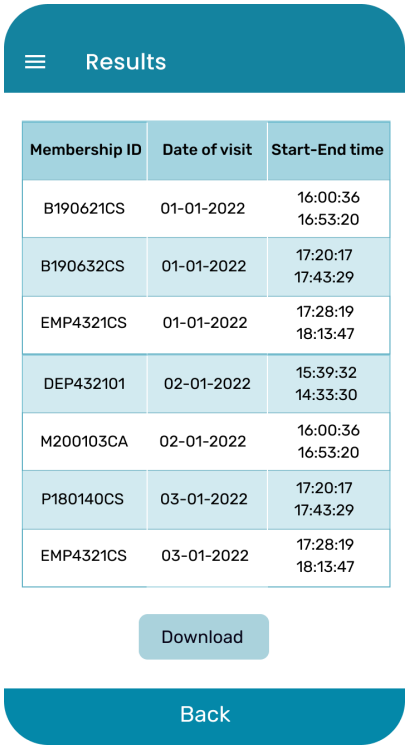


- **View and download user history [Corresponds to Use Case #6]**
  After searching for a student, the admin will be able to view and download the details of all the student's visits.

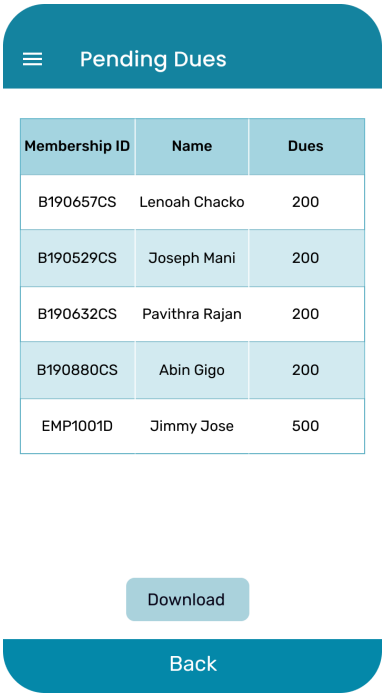
- **History of a specific day [Corresponds to Use Case #5]**
  This page displays the detailed history of the visits on a specific day. The details of the visits shown are Membership ID, Date of visit and Start-End time. Pressing the 'Back' button will take the admin back to the Search Page. The admin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



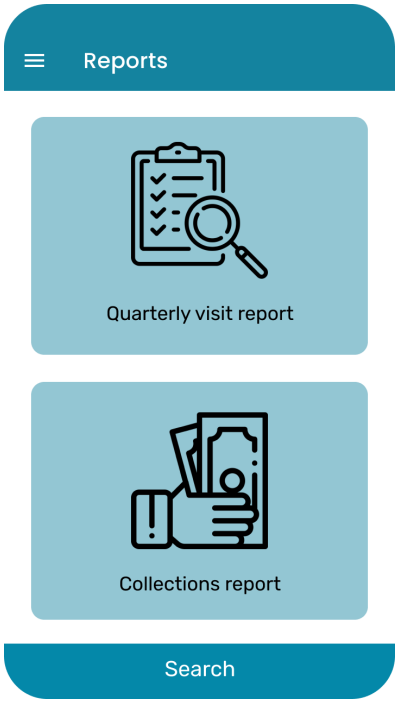
- **History of a date range [Corresponds to Use Case #5]**
  This page displays the detailed history of the visits on a specific date range. The details of the visits shown are Membership ID, Date of visit and Start-End time. Pressing the 'Back' button will take the admin back to the Search Page. The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



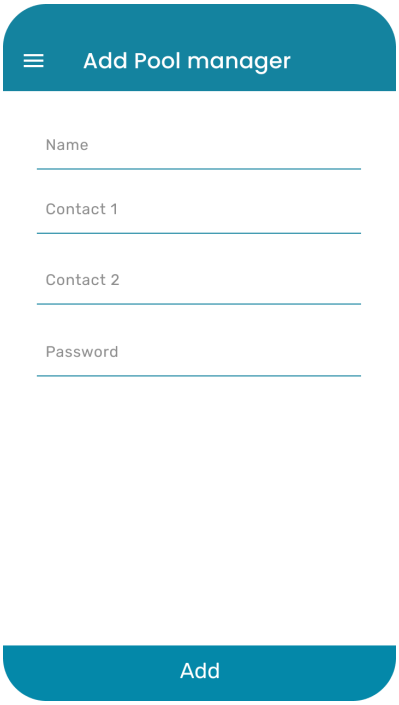
- **View the pending dues [Corresponds to Use Case #10]**
  The admin can view the pending dues of all the swimmers and download the results

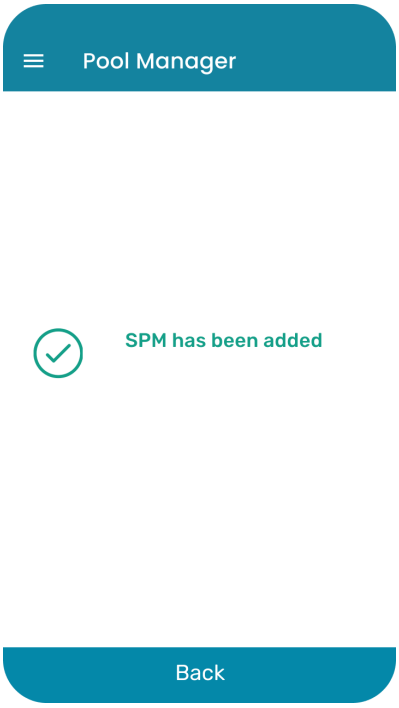


- **Generate and download quarterly visit and collection reports [Corresponds to Use Case #14]**
  After clicking on the Reports option, the admin can generate a quarterly visit report or a collections report.



- **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.




Pool Manager
SPM has been added
Back

Design Document

for

swimiNIT

Version 1.1

Prepared by Team 05:
(Based on SRS Version 1.1 prepared by Team 05)

Lenoah Chacko	**B190657CS**	**lenoah_b190657cs@nitc.ac.in**
Pavithra Rajan	**B190632CS**	**pavithra_b190632cs@nitc.ac.in**
Abin Gigo Joseph	**B190880CS**	**abin_b190880cs@nitc.ac.in**
Joseph Mani Jacob Mani	**B190529CS**	**joseph_b190529cs@nitc.ac.in**
Akshay Kuttikkattuparambil Biju	**B190803CS**	**akshay_b190803cs@nitc.ac.in**
Varun Chittezhath Anilkumar	**B190621CS**	**varun_b190621cs@nitc.ac.in**

Project Owner: **Ms. Lekshmy P Chandran**

Course: **CS4096D Software Engineering Laboratory**

Date: **29/03/2022**

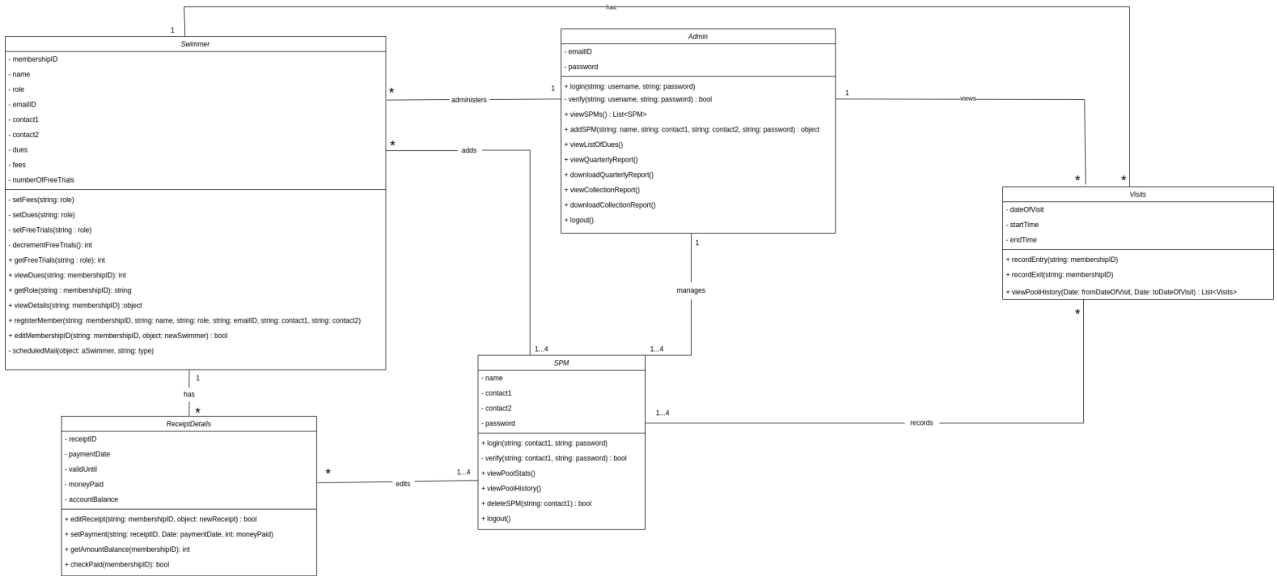
Glossary

SPM	Swimming Pool Manager
SPO	Swimming Pool Office
UML	Unified Modelling Language

1. Detailed Design through UML diagrams

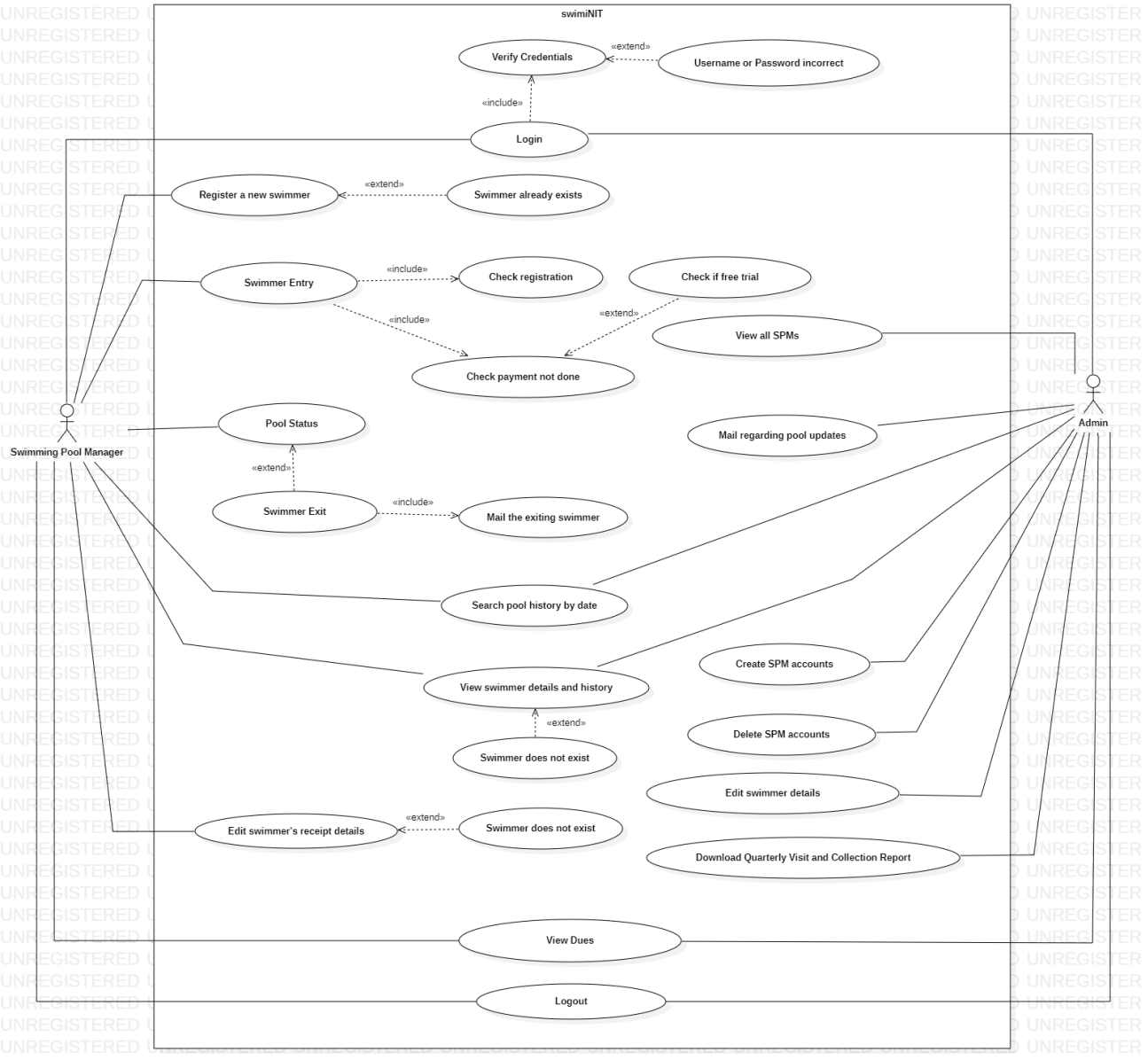
1.1 System model using Class Diagram

Class Diagram in the Unified Modelling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods) and the relationships among classes.



1.2 Responsibilities - Usecase Diagram

Use case diagram graphically depicts the user's possible interactions with the system. It shows the different types of users (actors) and the use cases that the actors perform when they are using the system to solve the customer's problem. The actor is shown as a stick person and the use case is shown as an ellipse. Lines indicate which actors perform which use cases.

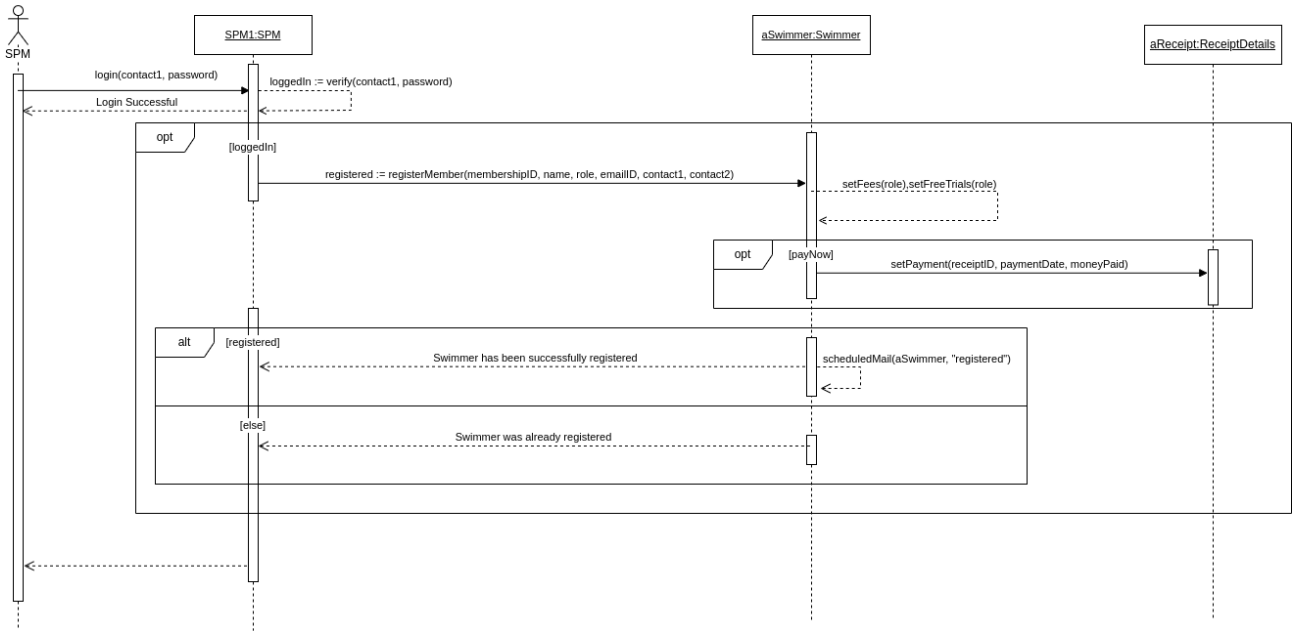


1.3 System Interactions through Sequence Diagrams

Sequence diagrams are interaction diagrams that show the sequence of messages exchanged by the set of objects performing a certain task. A sequence diagram shows, as parallel vertical lines (lifeline), different processes or objects that live simultaneously, and as horizontal arrows, the messages exchanged between them, in the order in which they occur.

1.3.1 Registration of a new swimmer

The SPM logs in to the system after the verification of their login credentials and registers the swimmer after setting the payment option to ‘pay now’ or ‘pay later’. If the SPM chooses ‘pay now’, they will be prompted to register the receipt details of the payment made by the swimmer to the SPO. If the swimmer is already registered to the system, the SPM will be informed via a response message.



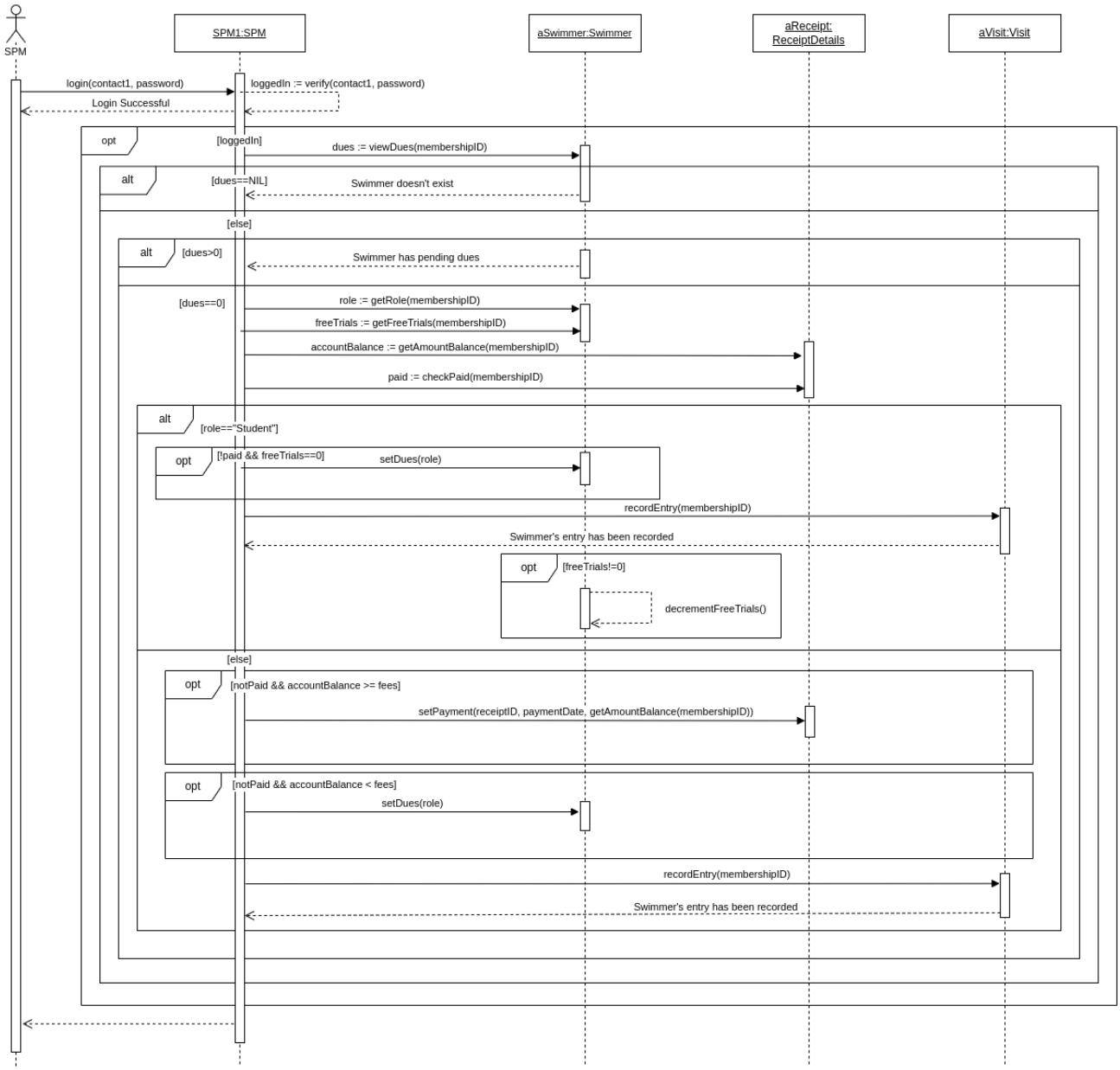
1.3.2 Swimmer Entry

After successfully logging in to the system, the SPM enters the membership ID of the swimmer. If the dues corresponding to the swimmer is NIL, the SPM is notified that the swimmer is not registered. If dues exist, and it is a numerical value, the system checks whether the swimmer has dues or not based on which they are allowed entry into the pool. If the swimmer has standing dues, they are not permitted to enter the pool.

If a student swimmer has not paid and doesn't have any free trials left then dues are recorded. They are permitted to enter the pool and their entry is recorded. Student

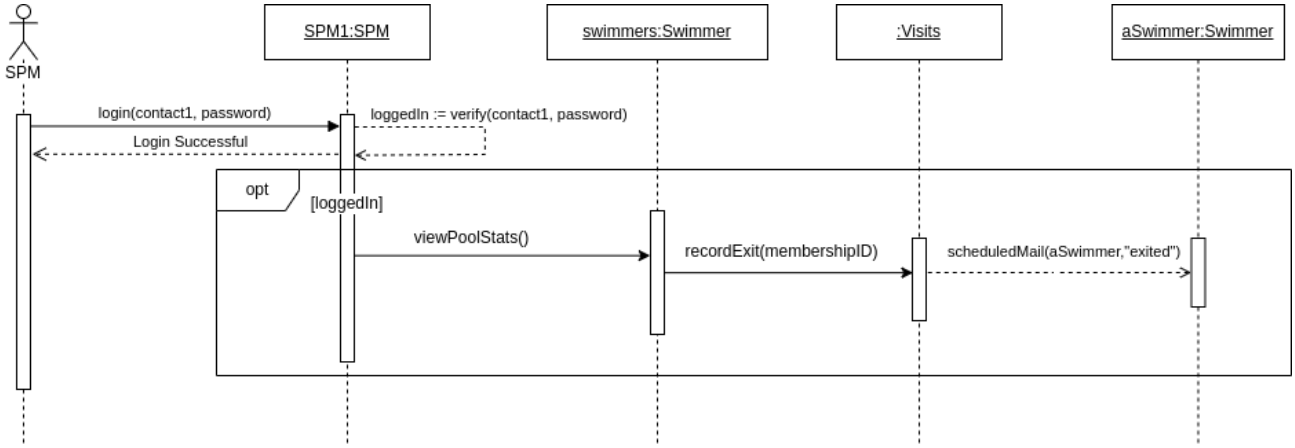
swimmers who have free trials left are permitted to enter the pool and their free trials will be decremented after their entry is recorded.

If a non-student swimmer has not paid and their account has a balance amount exceeding the fees for that quarter then the fees will be deducted from the account balance. They are permitted to enter the pool and their entry is recorded. If they have not paid and their account balance is less than the fees for that quarter then their dues are recorded. They are permitted to enter the pool and their entry is recorded.



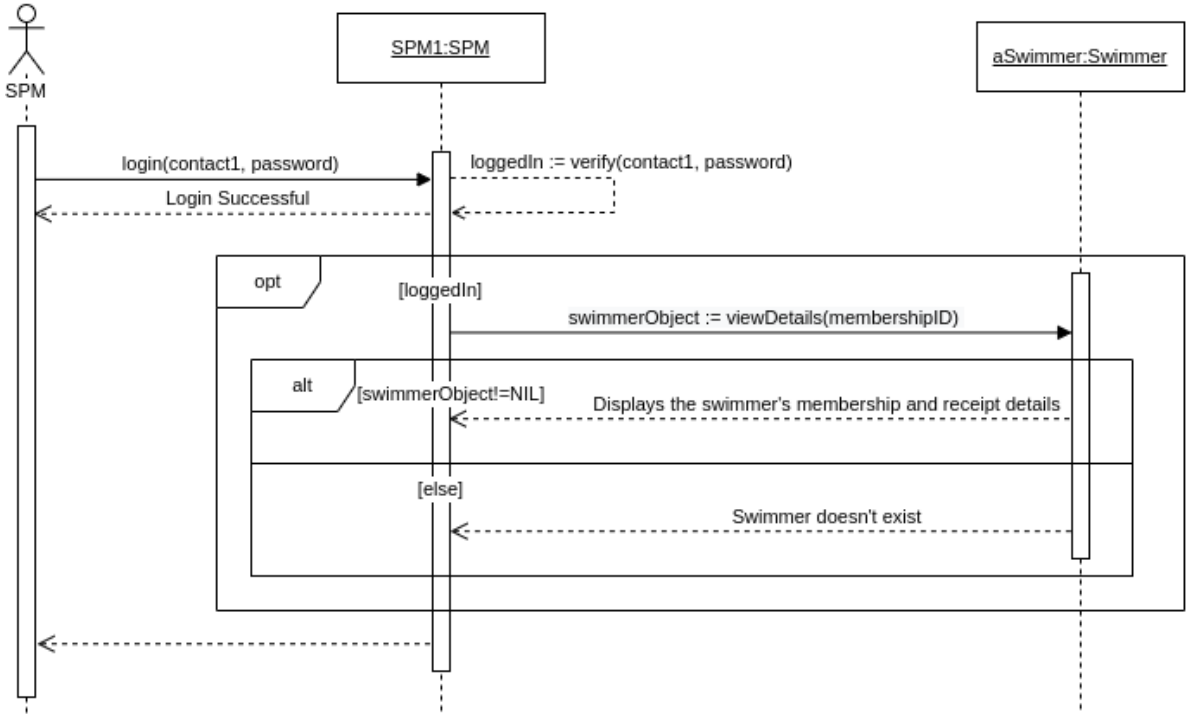
1.3.3 Swimmer Exit

After successfully logging in to the system, the SPM can view the list of swimmers currently in the pool and ‘exit’ a particular swimmer after confirmation. Simultaneously, a mail is automatically sent to the swimmer informing them about the visit.



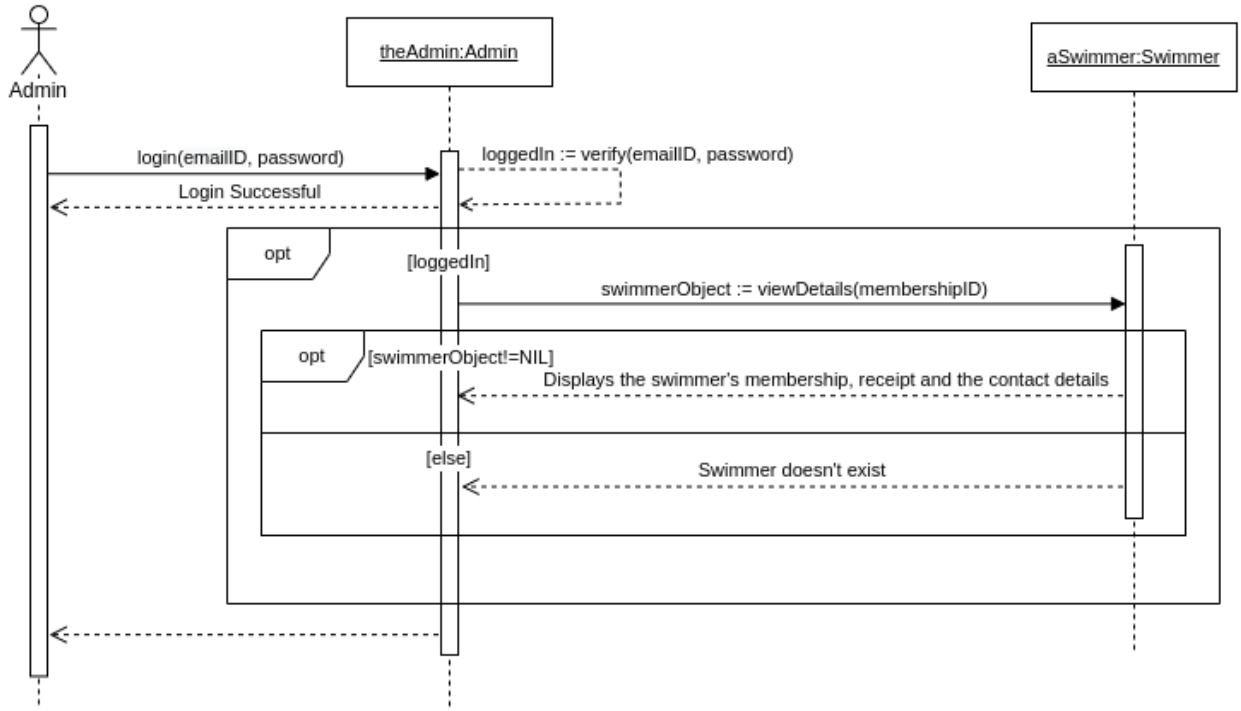
1.3.4 Search swimmer by membership ID - (for SPM)

After successfully logging in to the system, the SPM can view the details of a particular swimmer by entering the Membership ID of the respective swimmer. If the swimmer exists, details like name, role, dues (if any) and payment details will be displayed. Additionally, the visit history of the swimmer will be shown.



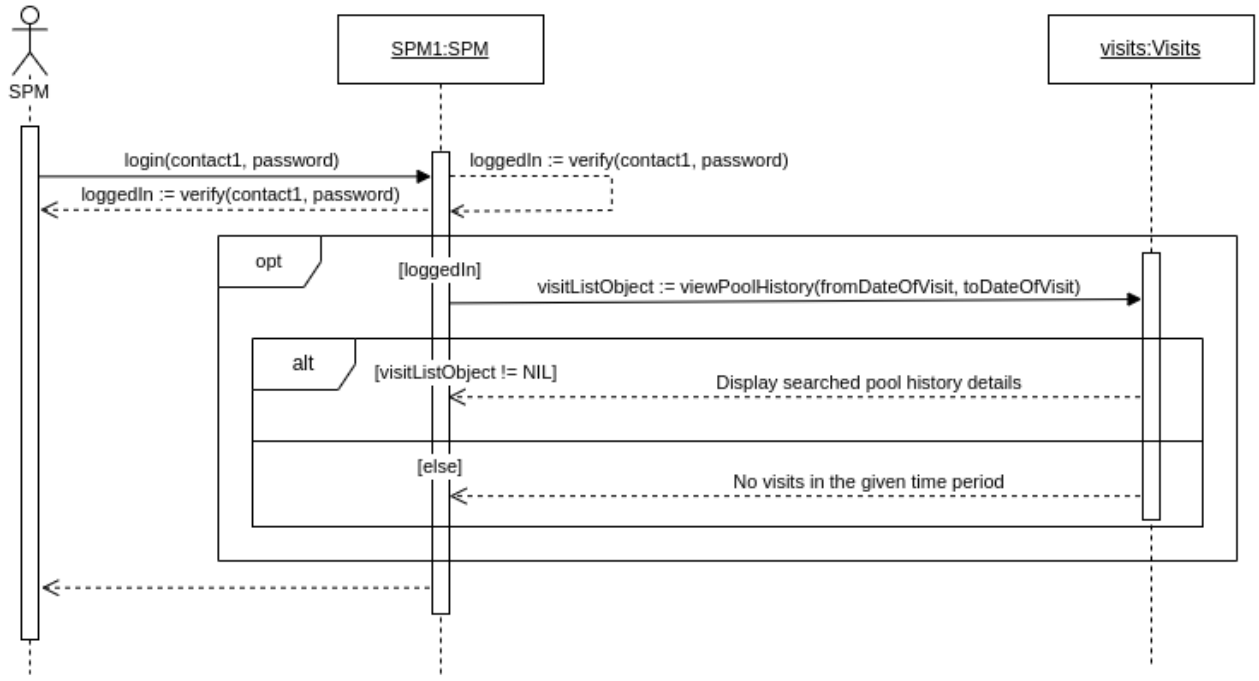
1.3.5 Search swimmer by membership ID - (for Admin)

After successfully logging in to the system, the Admin can view the details of a particular swimmer by entering the Membership ID of the respective swimmer. If the swimmer exists, details like name, role, dues (if any), contact information, email ID and payment details will be displayed. Additionally, the visit history of the swimmer will be shown.



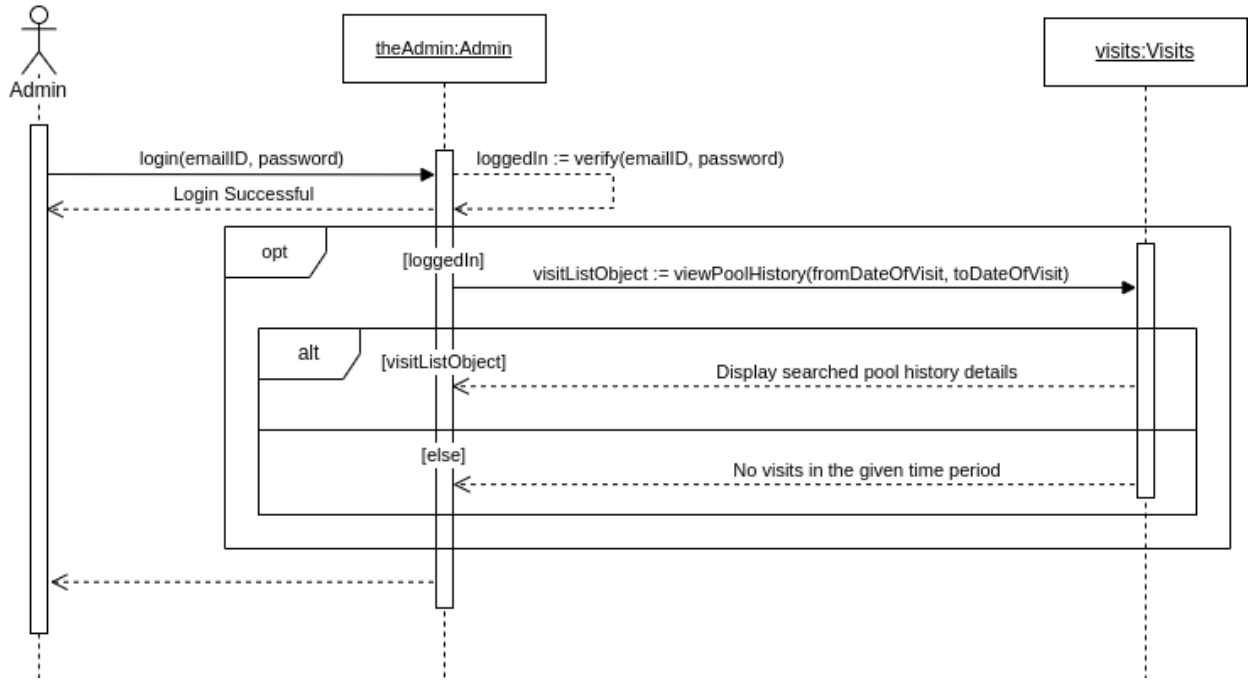
1.3.6 Search by date/date-range - (for SPM)

After successfully logging in to the system, the SPM can search for the swimmers who visited the pool on a particular date/date range. Details such as membership ID, date of visit and start-end time will be displayed.



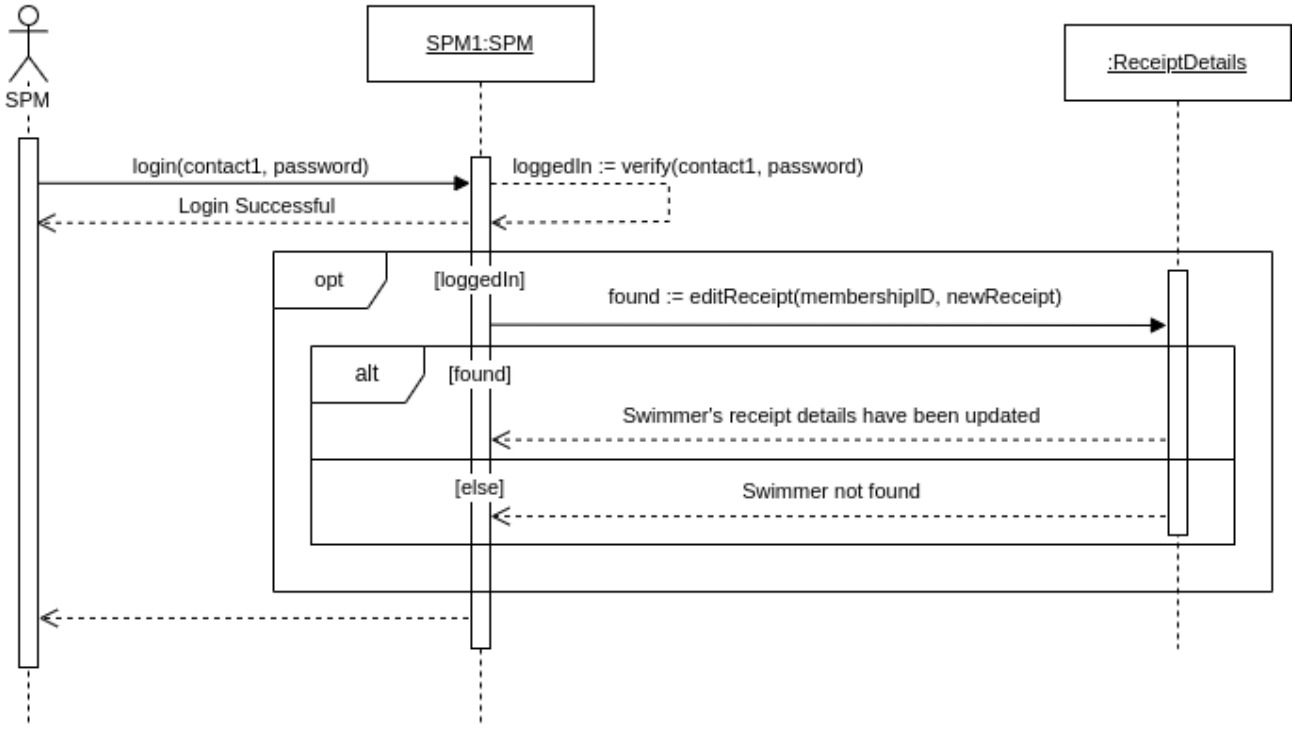
1.3.7 Search by date/date-range - (for Admin)

After successfully logging in to the system, the Admin can search for the swimmers who visited the pool on a particular date/date range. Details such as membership ID, date of visit and start-end time will be displayed.



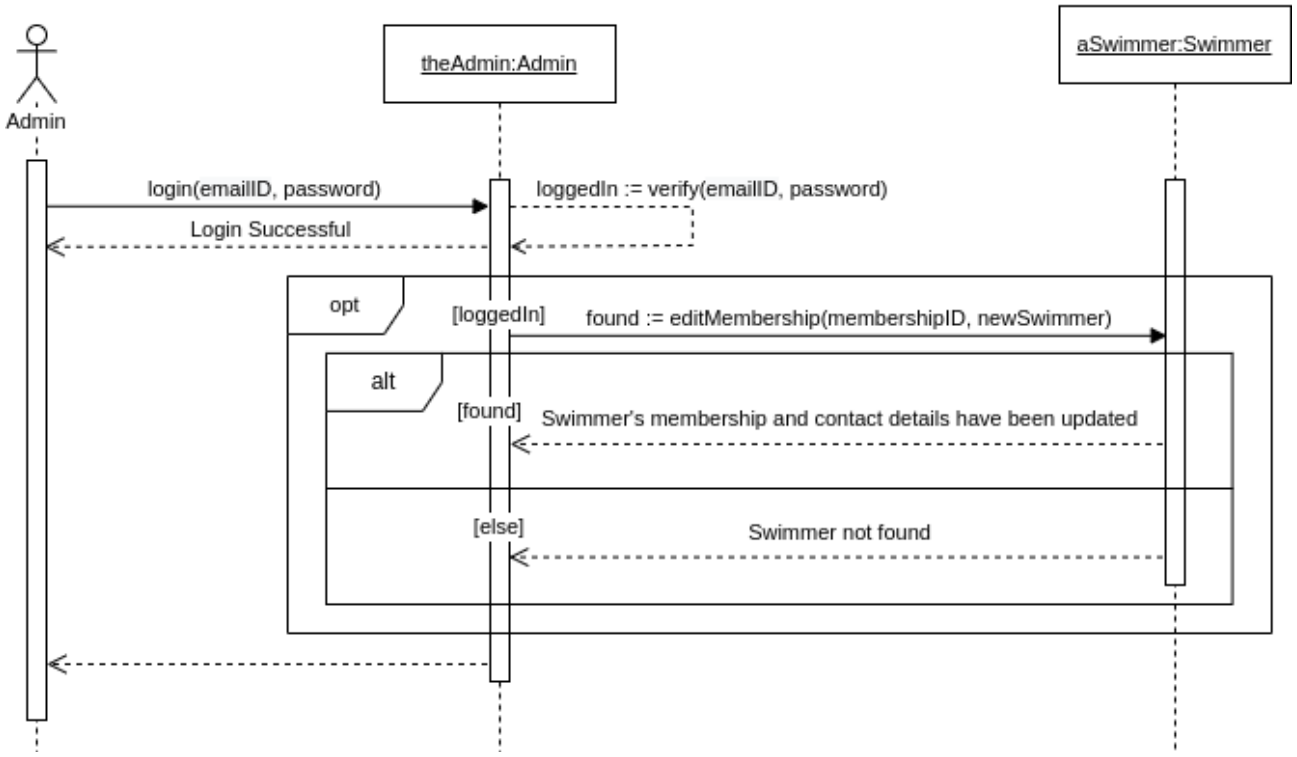
1.3.8 Edit receipt details

After successfully logging in to the system, the SPM can edit the receipt details of a particular swimmer. Initially, they have to search if the particular swimmer is registered to the system, using their membership ID. If the swimmer exists, the membership and receipt details will be displayed of which the receipt details will be made available for editing. The SPM will be informed that the required changes have been made.



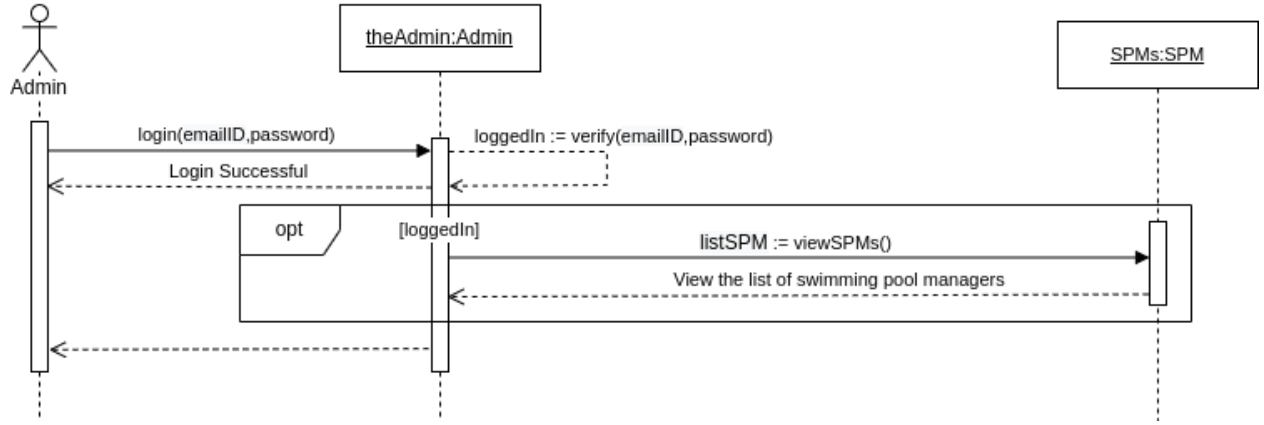
1.3.9 Edit swimmer details

After successfully logging in to the system, the Admin can edit the membership and contact details of a particular swimmer. Initially, they have to search if the particular swimmer is registered to the system, using their membership ID. If the swimmer exists, all the details of a particular swimmer will be displayed and everything except the receipt details and the membership ID can be edited. The Admin will be informed that the required changes have been made.



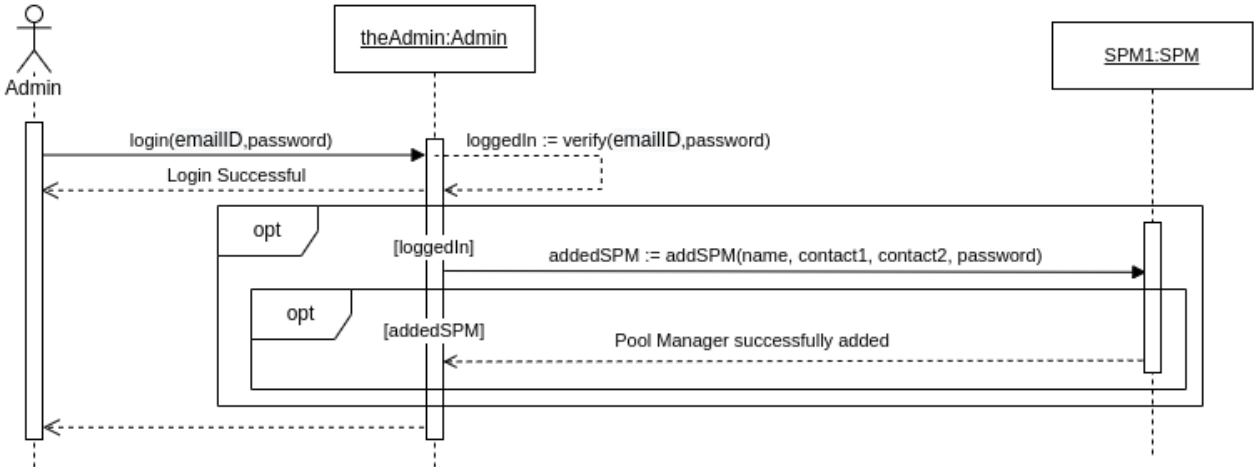
1.3.10 View SPM

After successfully logging in to the system, the Admin can view the list of SPMs.



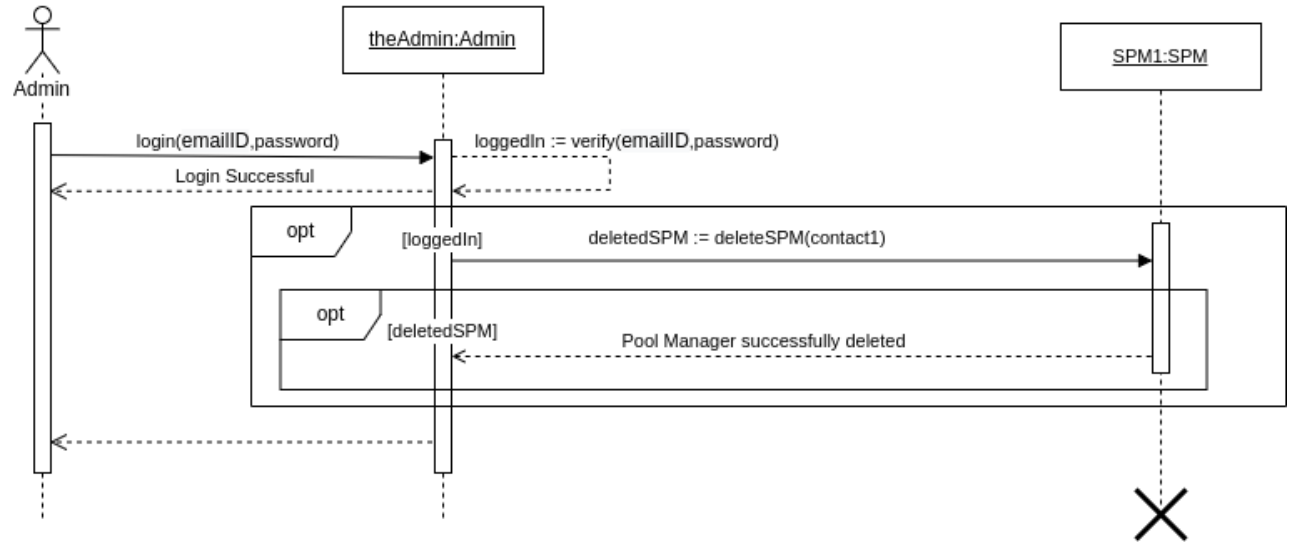
1.3.11 Add SPM

After successfully logging in to the system, the Admin can add an SPM by entering their name, contact details and password. Once the SPM has been added, the Admin will be able to view the SPM.



1.3.12 Delete SPM

After successfully logging in to the system, the Admin can view the list of SPMs, select the SPM that they wish to remove and delete them.

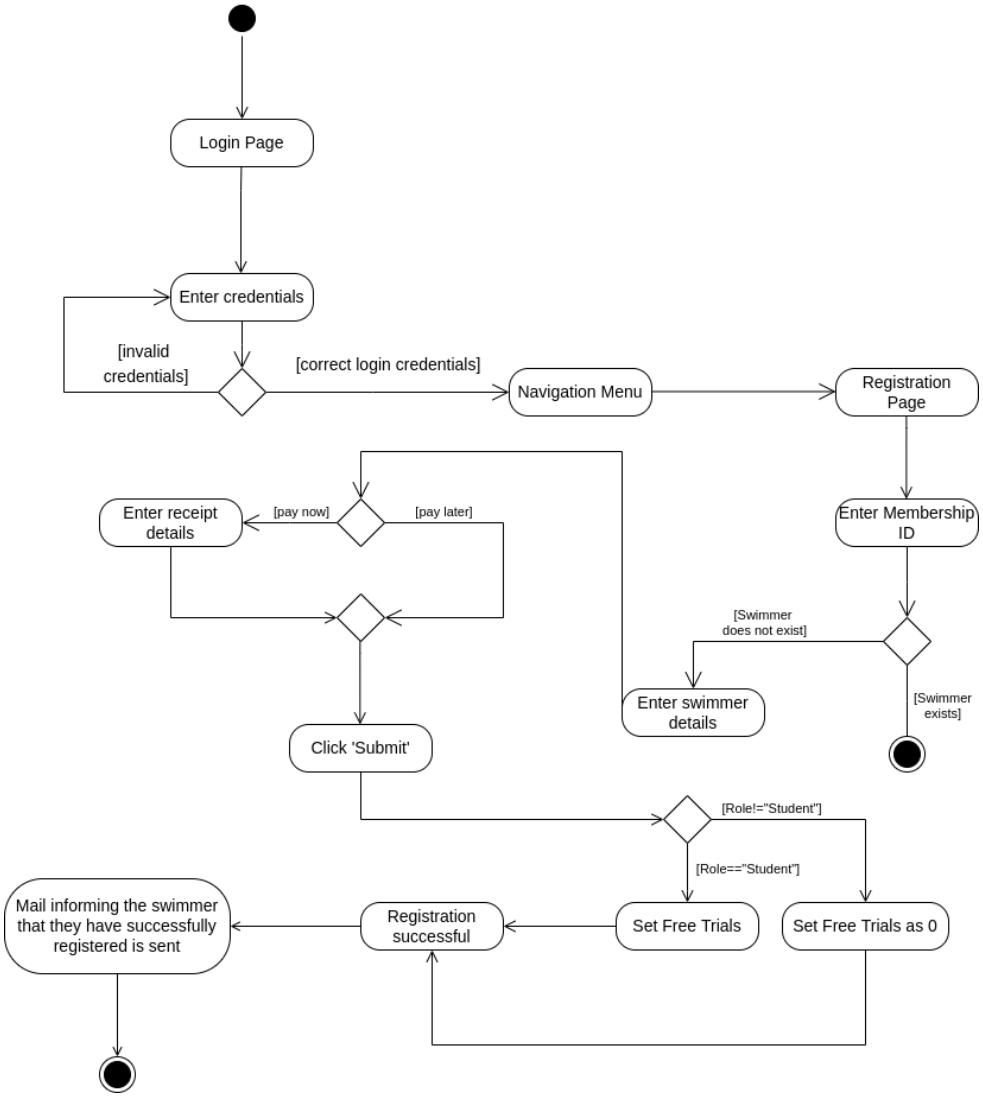


1.4 Control and Data Flows through Activity Diagrams

Activity diagrams graphically represent step-wise activities and actions involved in the workflow within a specific scenario, and helps to understand the flow of work that an object or component performs. Activity diagram uses rounded rectangles to represent a specific system function, arrows to represent flow through the system, decision diamonds to depict a branching decision, and solid horizontal lines to indicate that parallel activities are occurring.

1.4.1 Registration of a new swimmer

The SPM logs in to the system after the verification of their login credentials and registers the swimmer. The SPM will have two options: 'pay now' and 'pay later'. If the SPM chooses 'pay now', they can enter the receipt details of the swimmer. Otherwise, they can click submit to register the swimmer. The system will display to the SPM that the swimmer has been registered if they have been added to the system before.

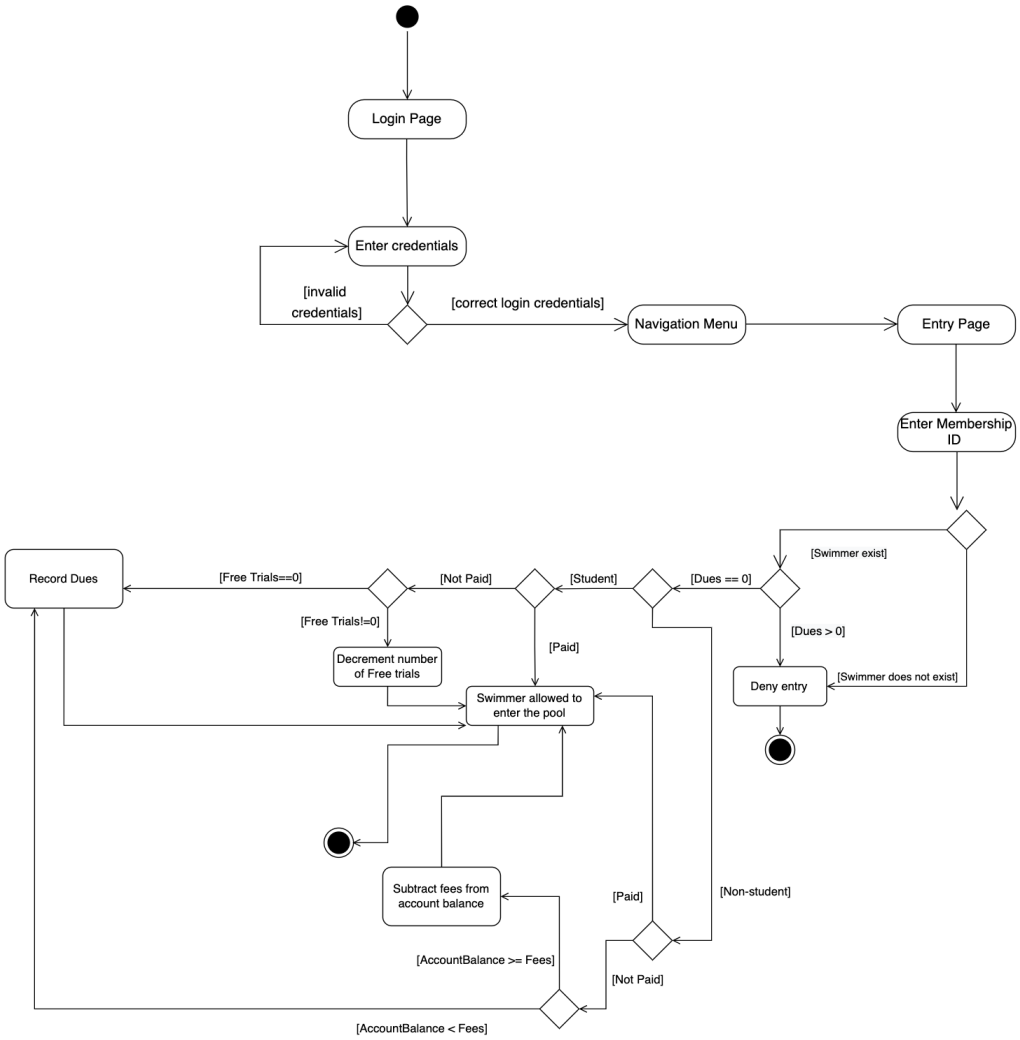


1.4.2 Swimmer Entry

The SPM logs in to the system after the verification of their login credentials. They proceed to the entry page and enter the membership ID of the swimmer. If the membership ID does not exist, the swimmer is denied entry into the pool and if it exists the system checks if the swimmer has dues or not. If they have dues, they are denied entry and if they don't have dues, the system checks if they are a student or non-student.

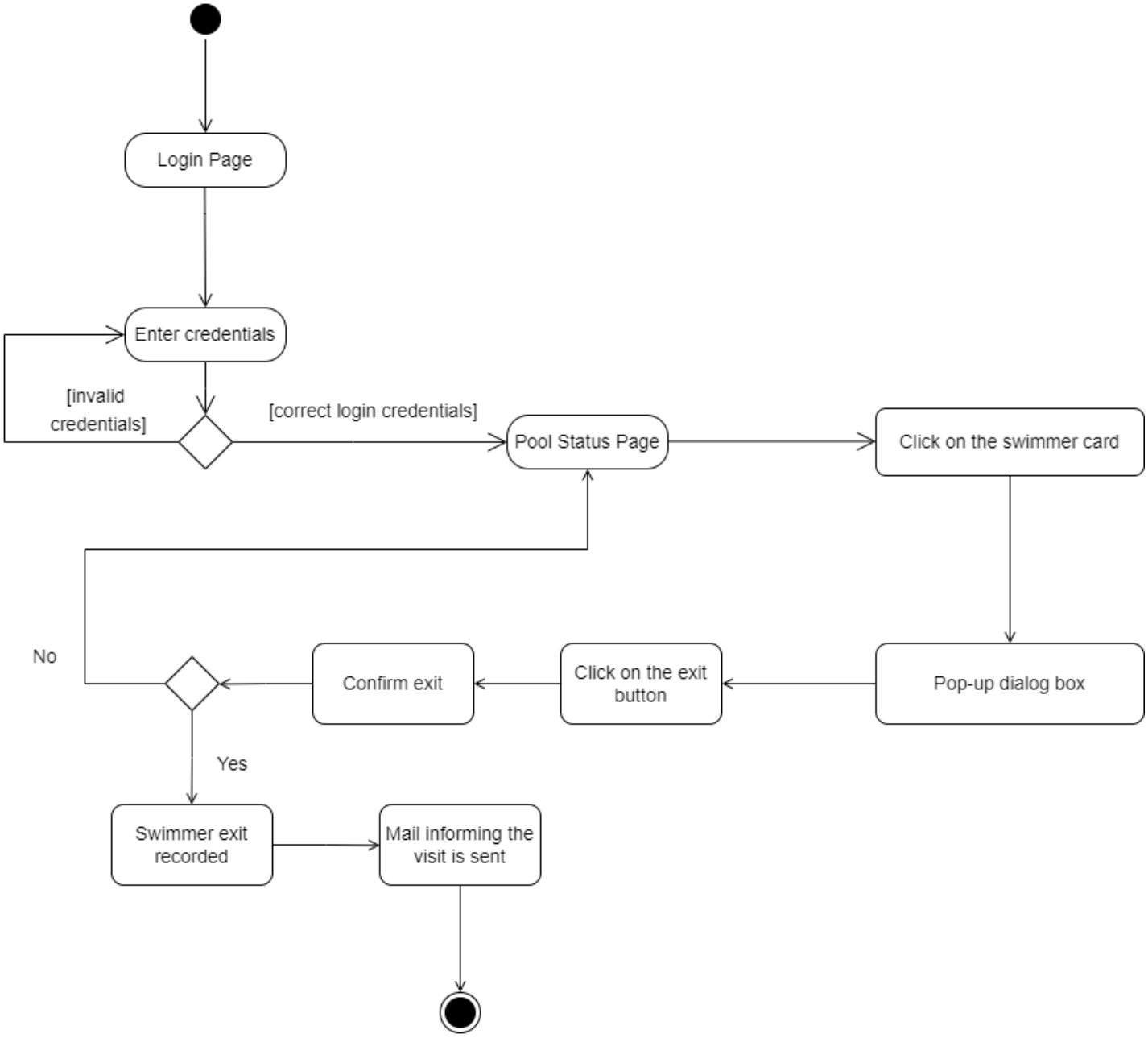
If they are a student, the swimmer is allowed to enter the pool if they have paid. If the student has not paid then the remaining number of free trials is checked. If the number of free trials is zero then the dues are recorded and the swimmer is permitted to enter the pool. If the student has free trials then the swimmer is allowed to enter the pool.

If they are a non-student swimmer and they have paid, the swimmer is allowed to enter the pool. If they have not paid, then the system checks their account balance and if it exceeds the quarterly fees, the fees are deducted from the account balance and the swimmer is allowed to enter the pool. If the account balance is less than the quarterly fees then the dues are recorded and the swimmer is allowed to enter the pool.



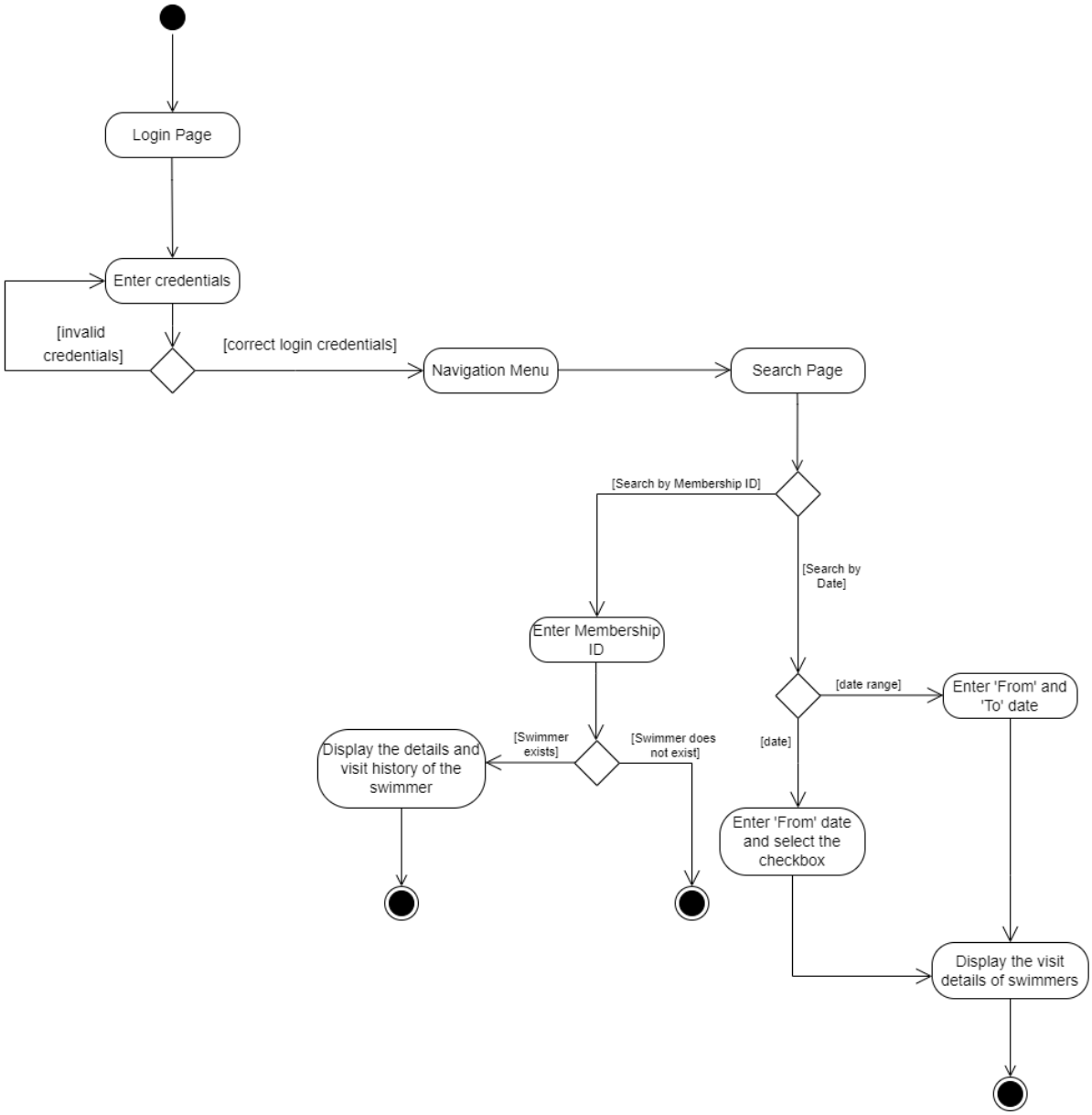
1.4.3 Swimmer Exit

After successfully logging in to the system, the SPM can view the list of swimmers currently in the pool and ‘exit’ a particular swimmer after clicking on the card containing the details of that swimmer and clicking on the ‘Exit’ button. A pop-up dialog box asking for confirmation will be displayed and if the SPM clicks ‘Yes’, the swimmer will be removed from the pool status page and their exit time will be recorded. Simultaneously, a mail is automatically sent to the swimmer informing them about the visit.



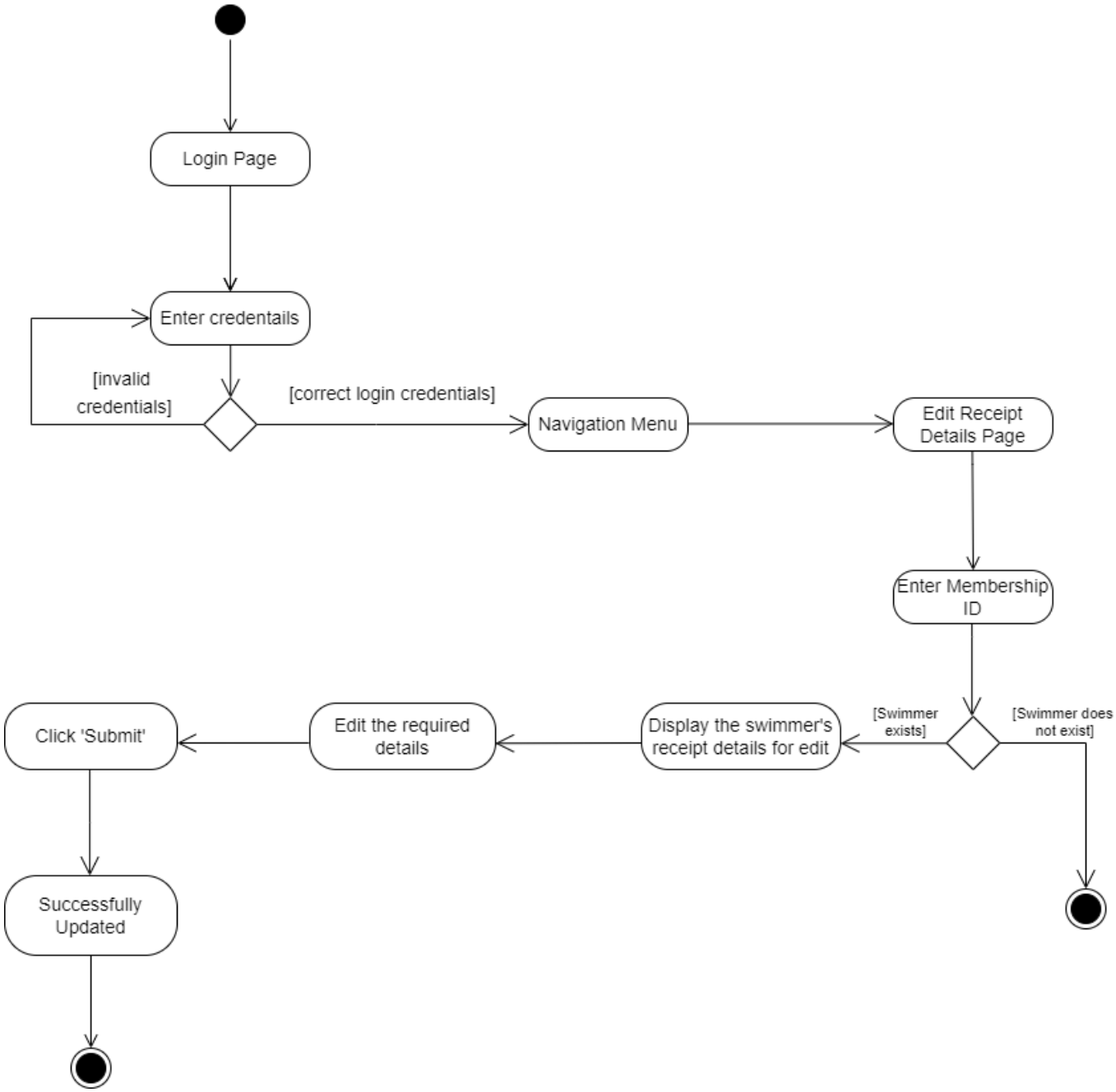
1.4.4 Search swimmer by date/date-range and Membership_ID

After successfully logging in to the system, the SPM and Admin can search for the swimmers who visited the pool on a particular date/date range. Details such as membership ID, date of visit and start-end time will be displayed. They can also find the details of a particular swimmer by entering the membership ID of the respective swimmer. If the swimmer exists, details like name, role, dues (if any) and payment details will be displayed to the SPM. Additionally, the visit history of the swimmer will be shown. For the Admin, contact details will also be visible.



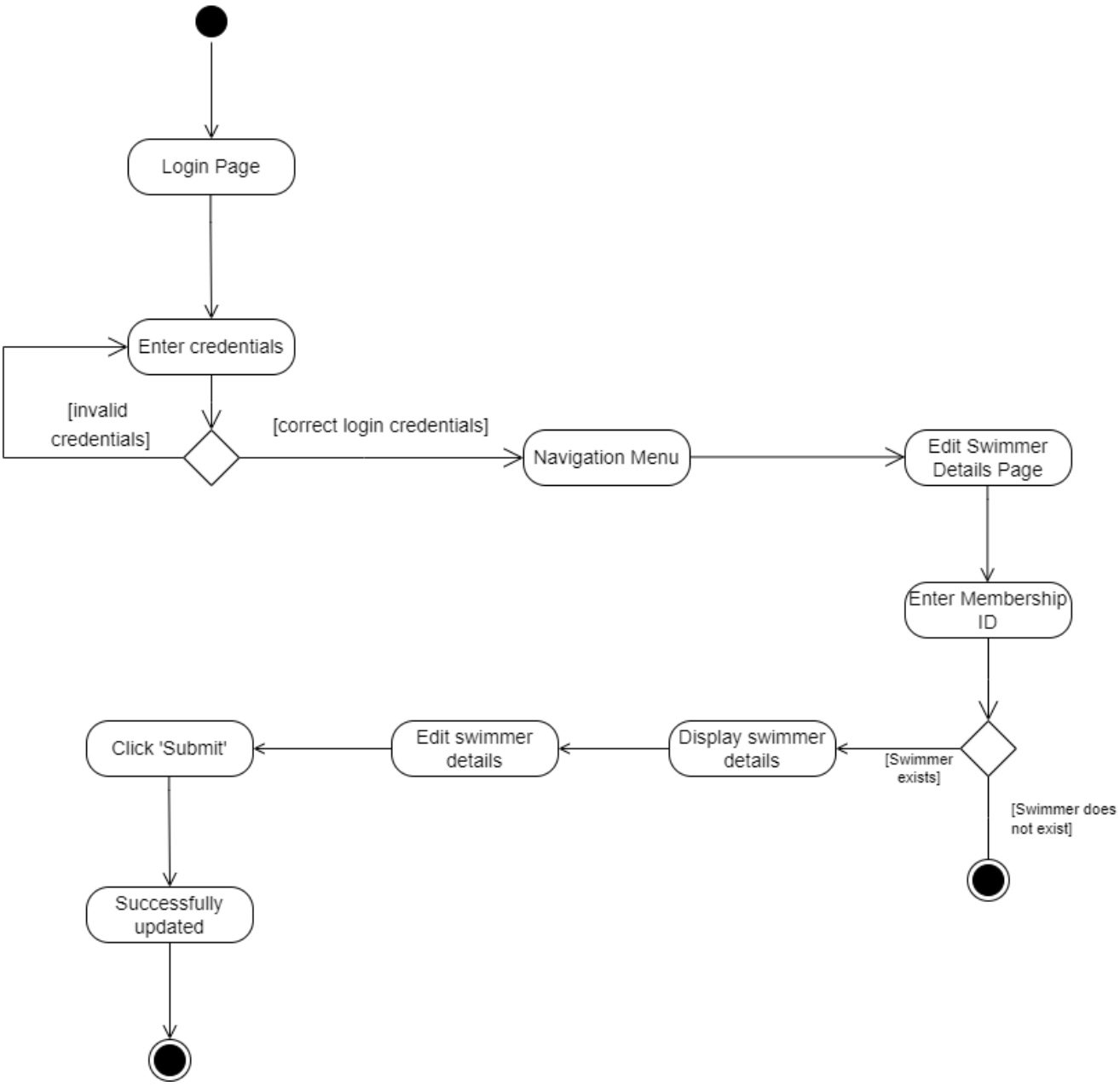
1.4.5 Edit receipt details

The SPM logs in to the system after the verification of their login credentials and proceeds to the edit receipt details page. The SPM enters the membership ID of the swimmer whose receipt details are to be edited. If the membership ID does not exist, the system displays an error message stating that the swimmer does not exist. If the membership ID does exist, all the details of the particular swimmer will be displayed and the receipt details will be made available for editing. The SPM makes the necessary changes and clicks the submit button. The SPM will be informed that the required changes have been made.



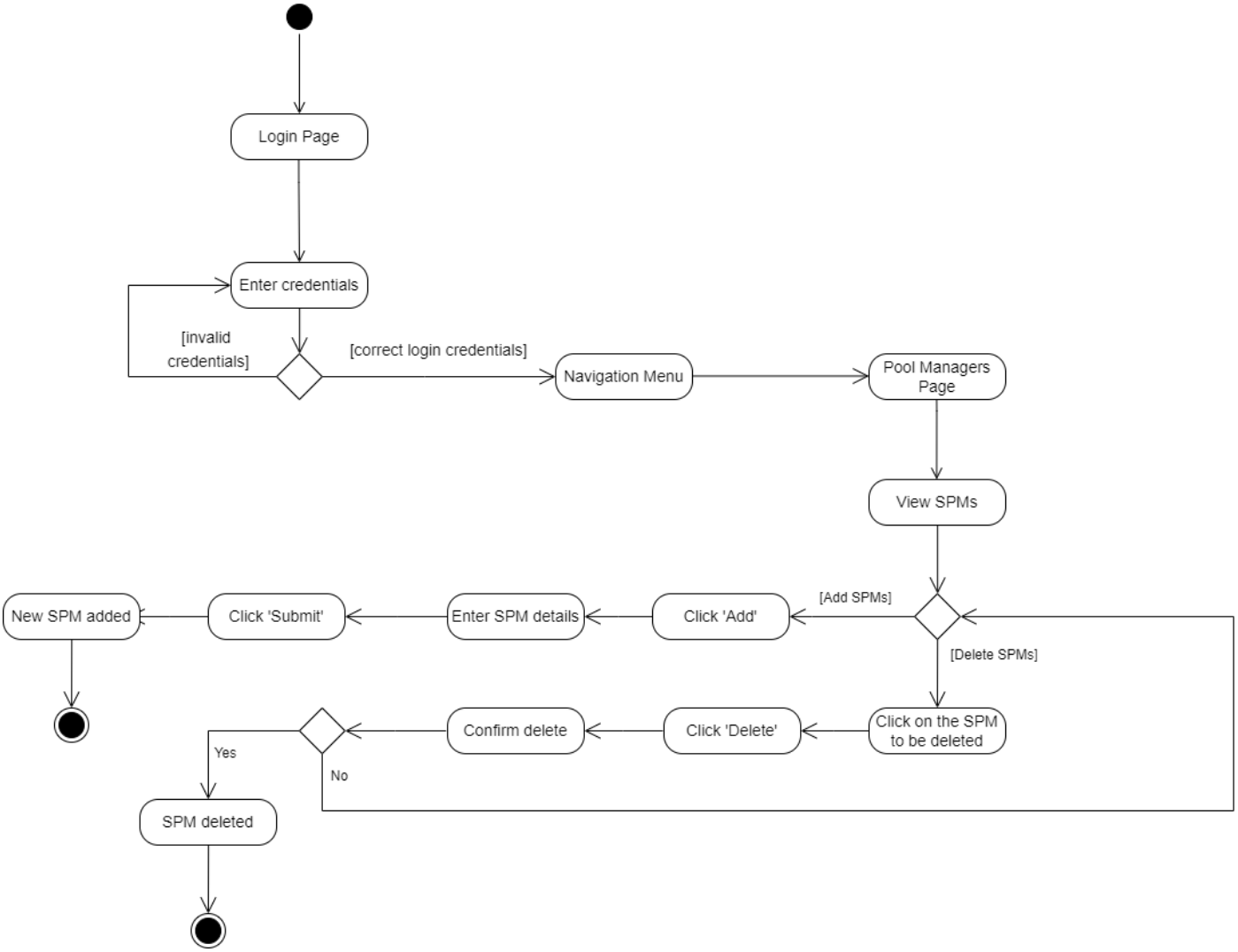
1.4.6 Edit swimmer details

The Admin logs in to the system after the verification of their login credentials and proceeds to the edit swimmer details page. The Admin enters the membership ID of the swimmer whose details are to be edited. If the membership ID does not exist, the system displays an error message stating that the swimmer does not exist. If the membership ID does exist, all the details of the particular swimmer will be displayed and all the details except the receipt details and the membership ID will be made available for editing. The Admin makes the necessary changes and clicks the submit button. The Admin will be informed that the required changes have been made.



1.4.7 Manage SPMs

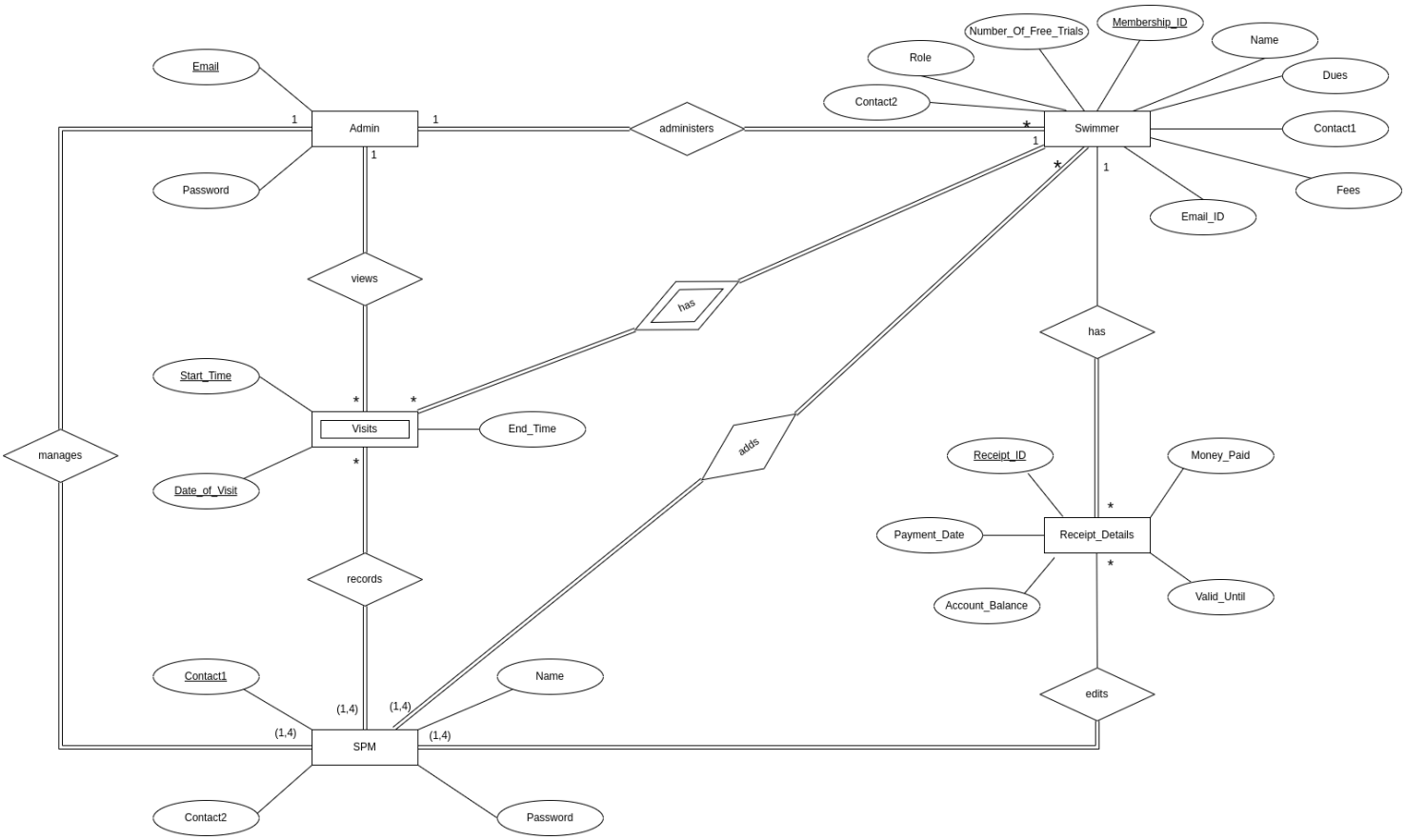
After successfully logging in, the Admin can view the SPMs that have been added. They can add SPMs by entering their name, contact details and password. If the SPM wishes to delete a particular SPM, the Admin has to click on the card of the SPM that they wish to delete and click the ‘Delete’ button. A confirmation dialog box will be displayed. If they click ‘yes’, the SPM will be deleted from the system. Otherwise, they will be redirected to the SPM view screen.



2. Database Design

2.1 ER Diagram

ER (Entity-Relationship) model is designed to represent the things that a system needs to remember in order to perform the system functionalities. It graphically represents the data model that defines the information structure which should be implemented in the database. The data objects (entity) are represented by a labelled rectangle and the relationships are indicated with a labelled line connecting objects.



3. Implementation Plans

3.1 Technology Stack

Framework	Flutter [Dart]
Interface design tool	Figma
Cloud platform [Database Management System]	Firebase
Collaboration tool	Trello
Version Control	Git
Diagram drawing tool	app.diagrams.net, StarUML

3.2 Work Estimates

Description	**Time Estimate (Hours)**	**Team Members Involved**	**Date of Completion**
Landing Page	30 minutes	Lenoah	15.03.22
Login and Navigation menu	2 hours	Pavithra, Varun	16.03.22
Pool Managers Screen (Admin View)	1 hour	Akshay	17.03.22
Add and Delete SPMs (Admin View)	3 hours	Abin, Joseph, Lenoah	18.03.22
Edit swimmer details (Admin View)	4 hours	Akshay, Pavithra, Abin	19.03.22
Display of swimmer details (common view with additional details displayed for Admin)	3 hours	Joseph, Lenoah, Akshay	20.03.22
Search page for membership ID and date/date range (common view)	2 hours	Lenoah, Varun	21.03.22
Display visit details for date range search (common view with download option for Admins)	4 hours	Pavithra, Akshay, Joseph, Abin	22.03.22

Display pending dues (Admin View)	2 hours	Joseph, Akshay	24.03.22
Download Quarterly Collection and Visit Report (Admin View)	2 hours	Varun, Abin	25.03.22
Registration for swimmer (SPM View)	4 hours	Pavithra, Lenoah, Varun	26.03.22
Pool status page with exit functionality (SPM View)	5 hours	Pavithra, Joseph, Akshay, Abin, Varun	27.03.22
Edit receipt details (SPM View)	2 hours	Lenoah, Joseph	28.03.22
Entry swimmer page with warning dialogue boxes (SPM View)	3 hours	Pavithra, Varun, Abin	29.03.22

References

1. T. C. Lethbridge and R. Laganiere (2004), *Object Oriented Software Engineering*, 1/e, Tata McGraw Hill.
2. Massimo Felici (2004), *Sequence Diagram Notes,* School of Informatics.

Test Case Summary

Testing Summary - SWE Lab - 2021W

Project Title: SwimNIT
Testing Team Number: 01
Names of Testing Team Members : Vishnu Ajay, Balaram, Akhil, Revanth, Umesh

(Attach additional sheets if necessary; Additional remarks on testing can be included at the end of the document)

Test Case # (include page# in test cases document)	Functionality (in words)	Pass/Fail	Remarks
1	Login	Pass	Logging out everytime when we minimize the app.
2	Login	Fail ✓	The validation of invalid credentials has been rectified.
3	Register swimmer	Pass	
4	Register swimmer	Pass	
5	Register swimmer	Fail ✓	Now, the app ensures that a registered membershipID is not registered again.
6	ENTRY	Pass	
7	ENTRY	Pass	
8	ENTRY	Pass	
9	Exit swimmer	Pass	It is taking so much time we have to click Back to face changes
10	Exit swimmer	Pass	

Testing Summary - SWE Lab - 2021W

Project Title: SwimiNIT
Testing Team Number: 01
Names of Testing Team Members: Vishnu Ajay, Balaram, Akhil, Revanth, Umesh

(Attach additional sheets if necessary; Additional remarks on testing can be included at the end of the document)

Test Case # (include page# in test cases document)	Functionality (in words)	Pass/Fail	Remarks
11	View list on swimmers on a specific date	Pass	Entry and Exit times are not displaying
12	View ~~list~~ History by membership ID	Pass	
13	View History invalid membership ID	Pass	
14	Send mail	Fail ✓	This has been rectified.
15	Edit swimmer details	Pass	
16	Edit swimmer details	Pass	
17	Pool Status	Pass	
18	Pool Status	Pass	
19	Pending dues	Fail ✓	This has been rectified.
20	View SPM's	Pass	

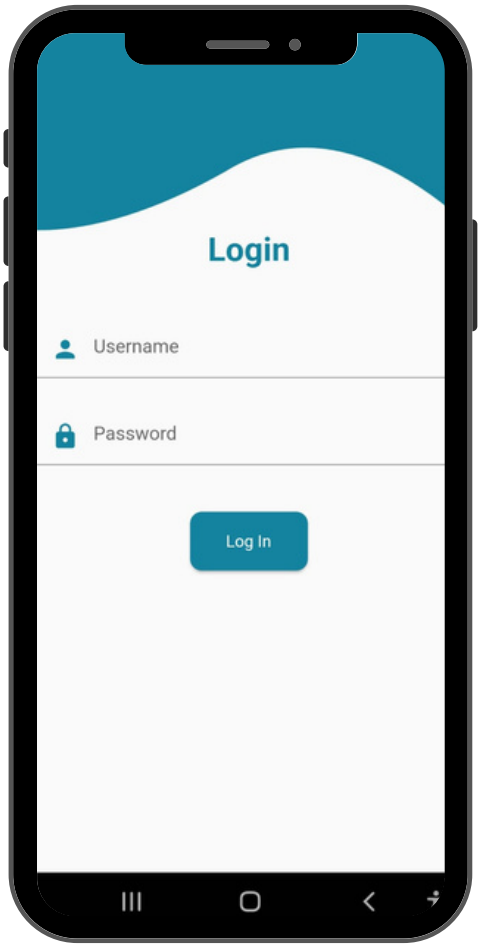
Testing Summary - SWE Lab - 2021W

Project Title: SwimiNIT
Testing Team Number: 01
Names of Testing Team Members : Vishnu Ajay, Balaram, Akhil, Revanth, Umesh

(Attach additional sheets if necessary; Additional remarks on testing can be included at the end of the documen

Test Case # (include page# in test cases document)	Functionality (in words)	Pass/Fail	Remarks
21	ADD SPM	Fail ✓	This has been rectified.
22	DELETE SPM	~~Fail~~ Pass	
23	DELETE SPM confirmation	Fail ✓	This has been rectified.
24	Download Report	Fail	The quarterly reports will be now mailed to the admin's mail.
25	Edit reciept details	Fail ✓	This has been rectified.
26	unsuccessfull in Edit reciept	Pass	
27	Logout	Pass	
~~28~~			
~~29~~			
~~30~~			

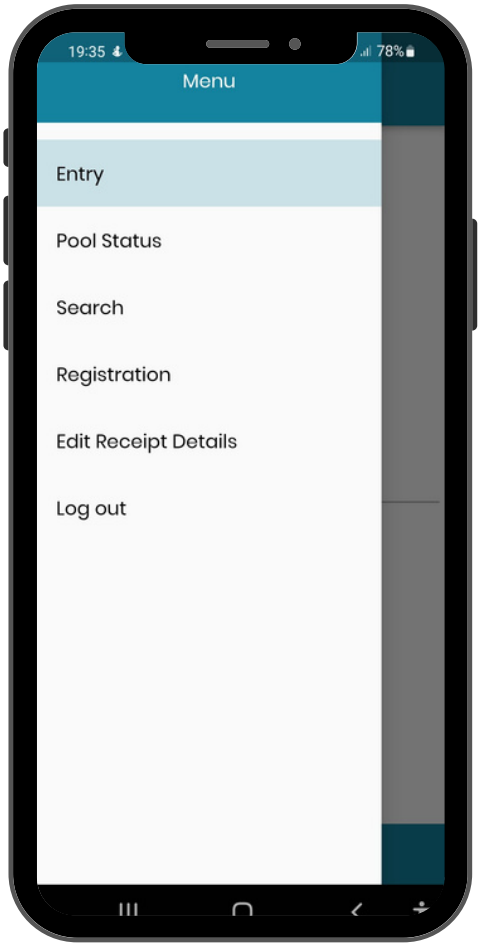
User Manual



Login

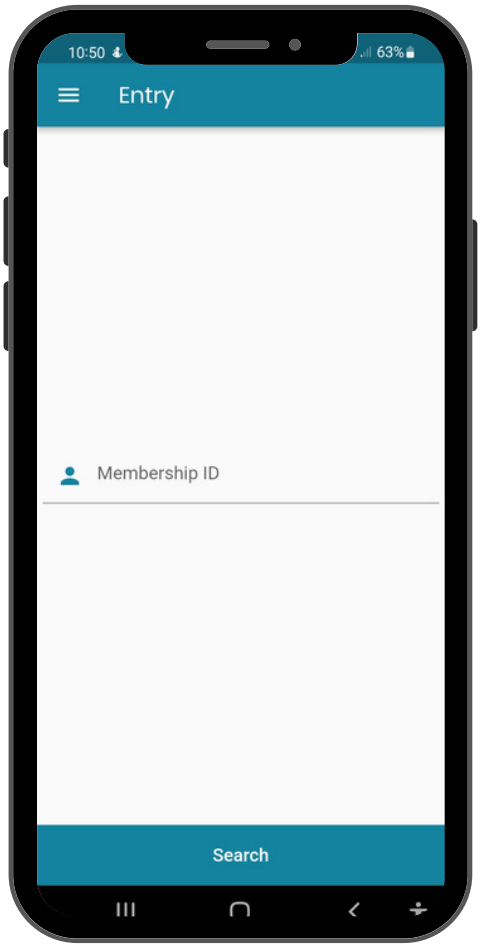
The login screen will be visible upon opening the app.

Enter the valid username and password and click the submit button. If the credentials are valid, you will be redirected to your respective dashboard. If you have entered invalid credentials, you will be notified with a dialog alert box.



Navigation Menu

If you are the **Swimming Pool Manager (SPM)**, this will be the navigation menu that will be visible. Click the required menu option to proceed.

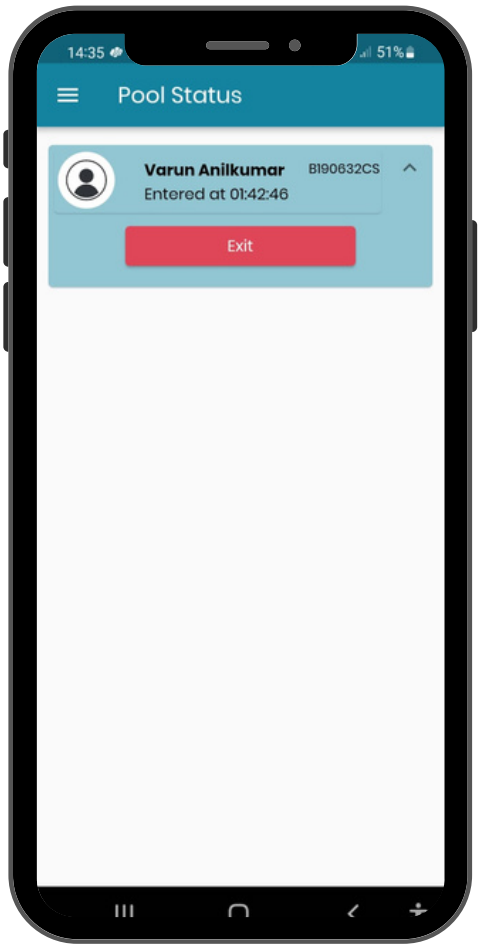


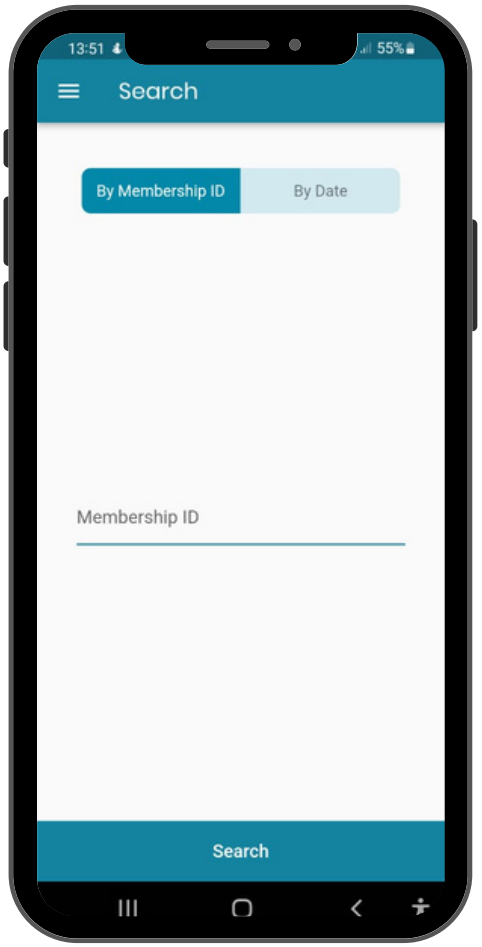
Entry

Click on the 'entry' option to facilitate the entry of a registered swimmer into the pool. Enter the membership ID and click the 'search' button. The details of the swimmer will be shown. Click on 'Allow Entry'. The swimmer will be shown on the 'pool status' page. If the swimmer has any dues, an alert for the same will be shown and the entry will not be recorded.

Exit

In order to 'exit' a swimmer from the pool, click on the card containing the details of the swimmer. It will expand to show an 'Exit' button. Once you have clicked that, a confirmation dialog box will appear. Click 'OK' to proceed.



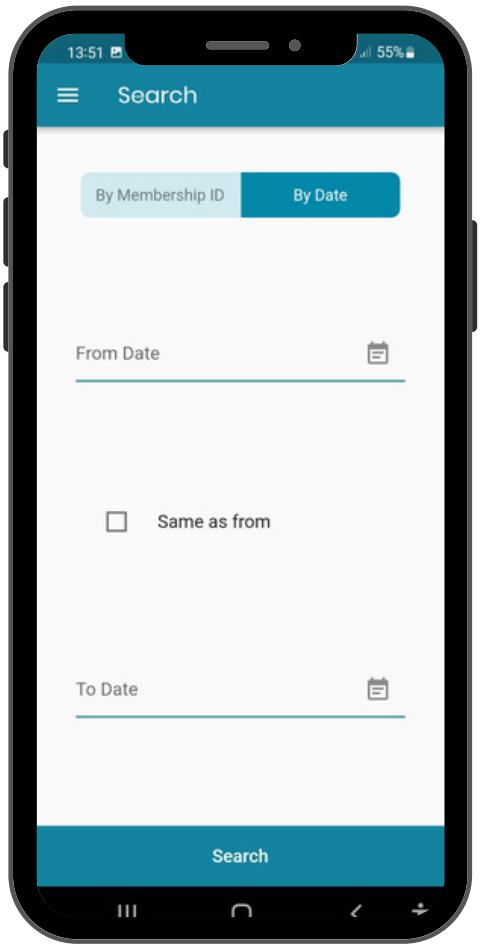


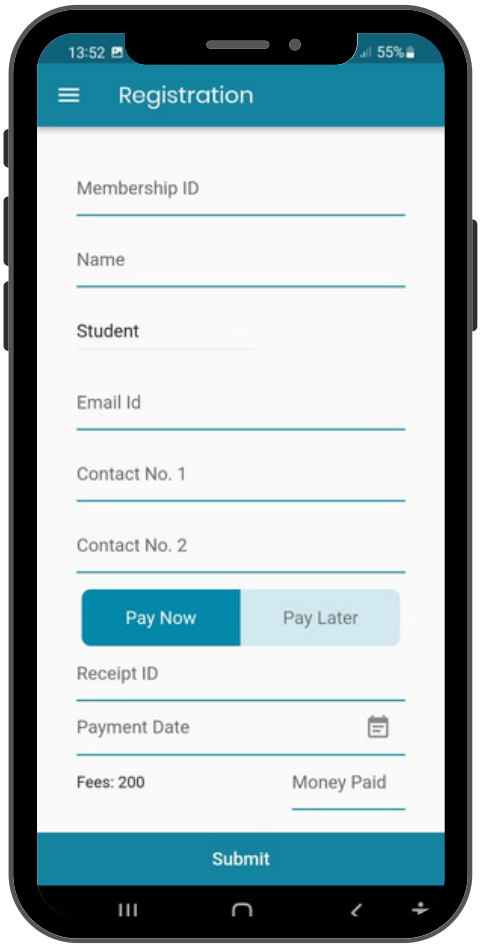
Search by membership ID

Click on the 'search' option to search for the details of a swimmer based on their membership ID, or to retrieve the visits on a particular date/date range. Choose the 'search by membership ID' toggle option and enter the membership ID.

Search by date/date range

Click on the 'search by date' toggle option and choose the 'from' and 'to' dates. If you choose to get the visit history of a particular day then select the 'same as from ' checkbox.



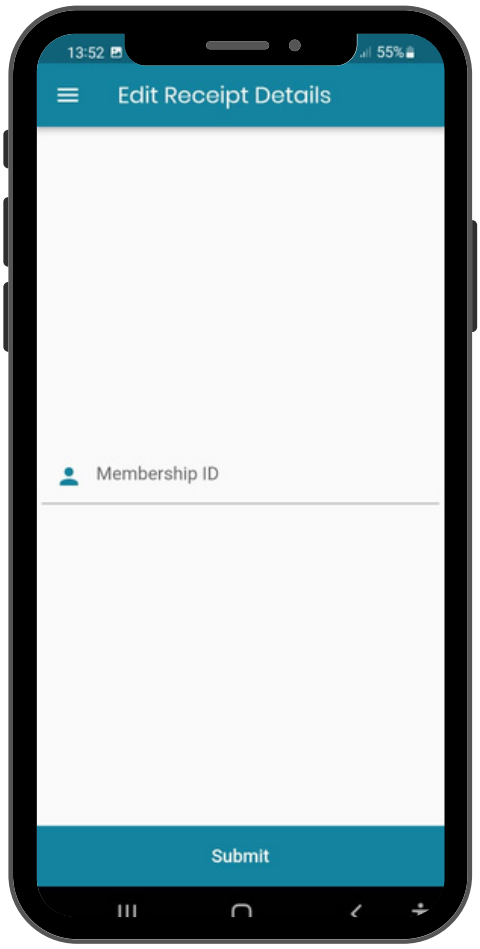


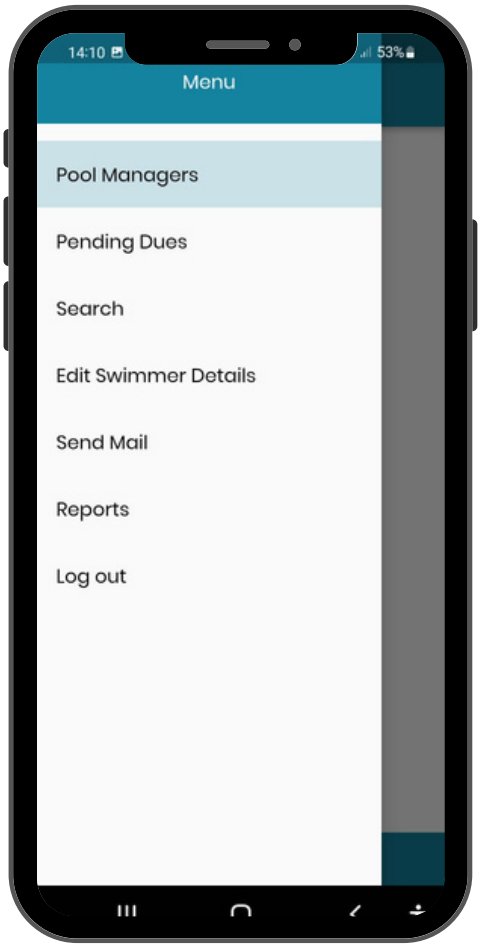
Registration

Click on the 'registration' option to register a swimmer. Enter the details in the fields shown. Click on the dropdown box to choose the role. If the swimmer has paid, click on the 'pay now' toggle to enter the receipt details.

Edit receipt details

Click on the 'edit receipt details' option. Search for the swimmer with their membership ID. Enter the new details and click the submit button.



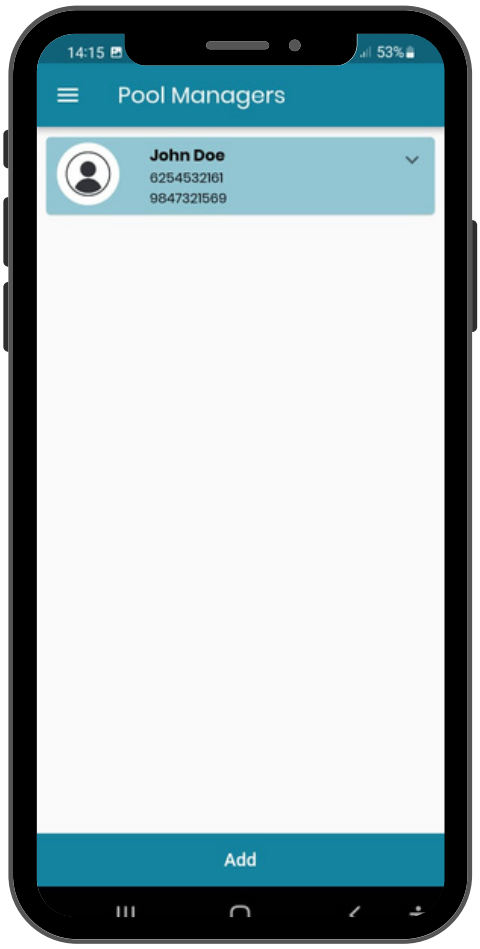


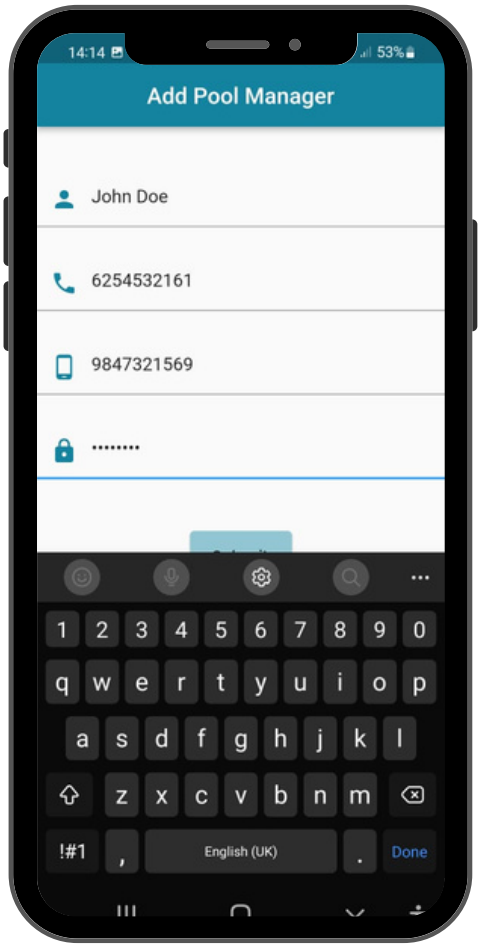
Admin Navigation Menu

If you are the **Admin**, this will be the navigation menu that will be visible. Click the required menu option to proceed.

View Swimming Pool Managers

Click on the 'Pool Managers' option to view the details of the pool managers that have been added. Click on the 'add' button to add new Swimming Pool Managers.



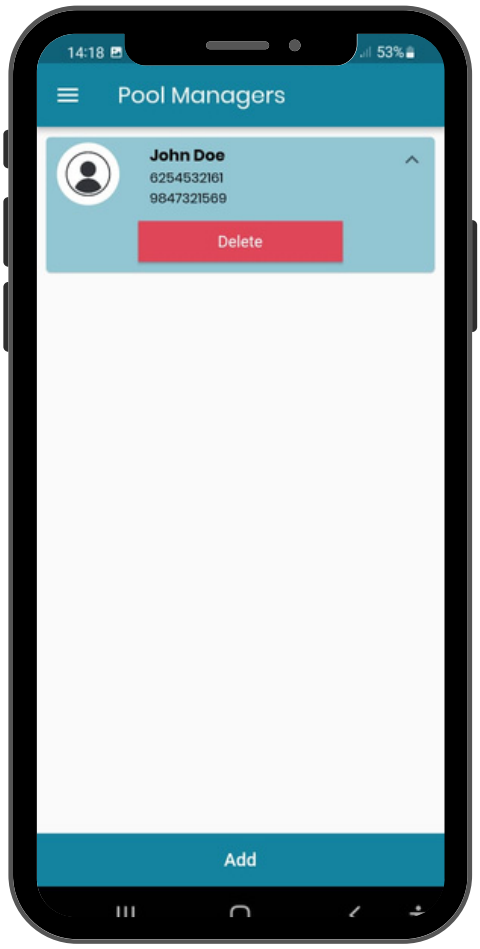


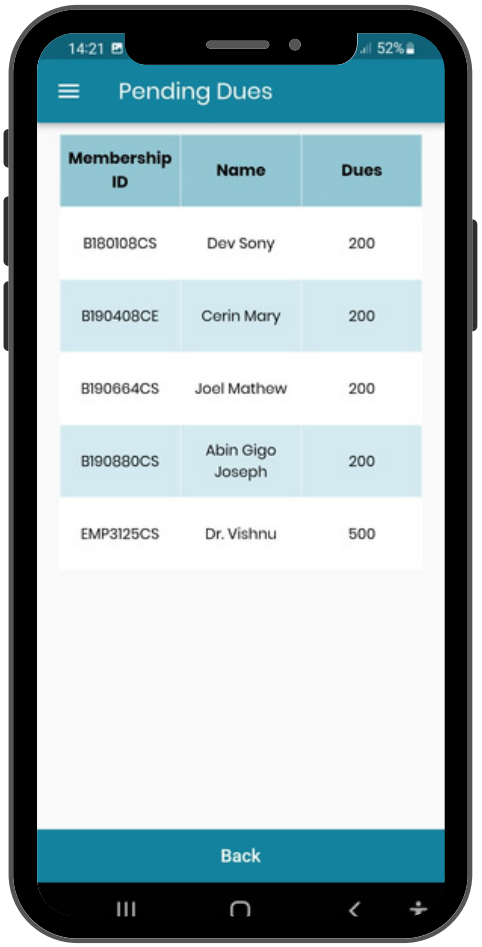
Add Pool Manager

Enter the details of the Swimming Pool Manager that you wish to add and click the 'submit' button. 'Contact1' and the 'password' entered will be the login credentials for the Swimming Pool Manager.

Delete Pool Manager

In order to delete a swimming pool manager, click on the card containing the details of the swimming pool manager and a delete button will appear. Click on it and the confirmation box that follows to delete the swimming pool manager.





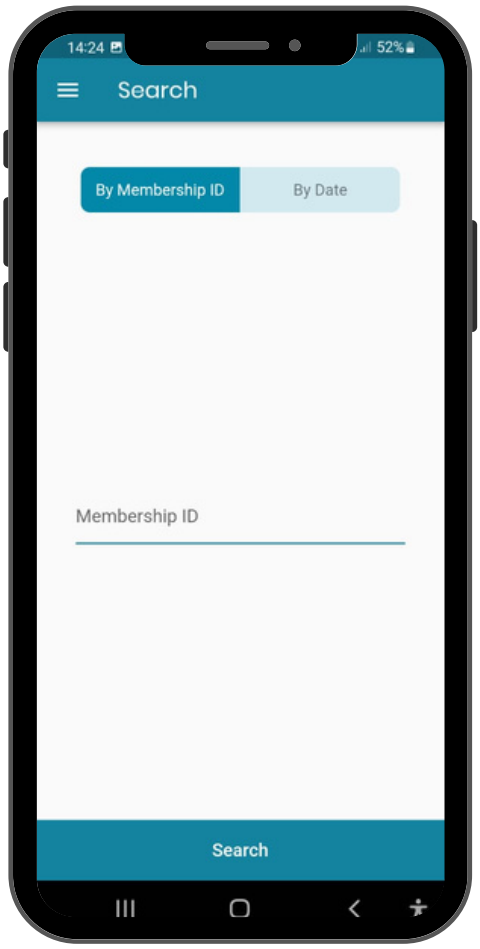
View Pending Dues

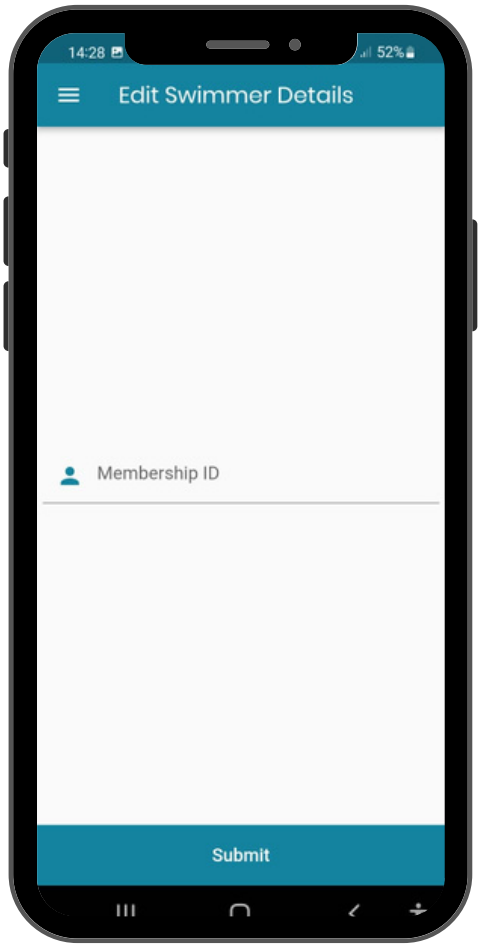
Click on the 'pending dues' option to view the details of the swimmers who have dues.

Search

Click on the 'search' option to retrieve the details of a particular swimmer using their membership ID. In order to do this, click on the 'by membership ID' toggle.

Click on the 'search by date' toggle button and choose the 'from' and 'to' dates. If you choose to get the visit history of a particular day then select the 'same as from ' checkbox.

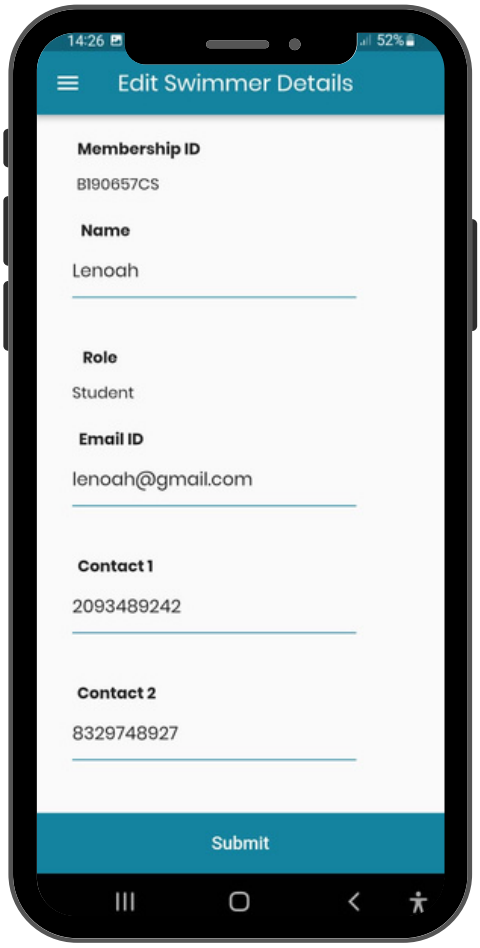


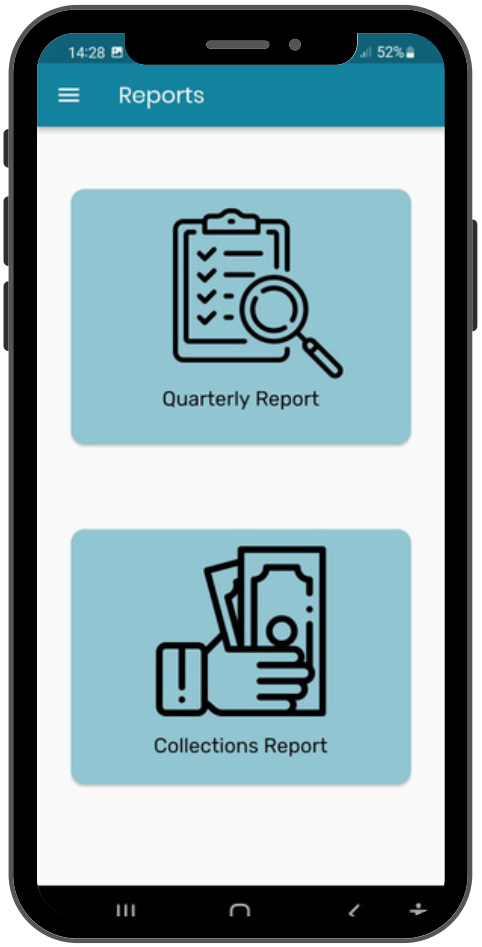


Edit swimmer details

Click on the 'Edit swimmer details' option to edit the details of the swimmer after searching using their membership ID. If the swimmer is registered then their details will be retrieved for edit.

Enter the details you wish to change and hit the 'submit button'.



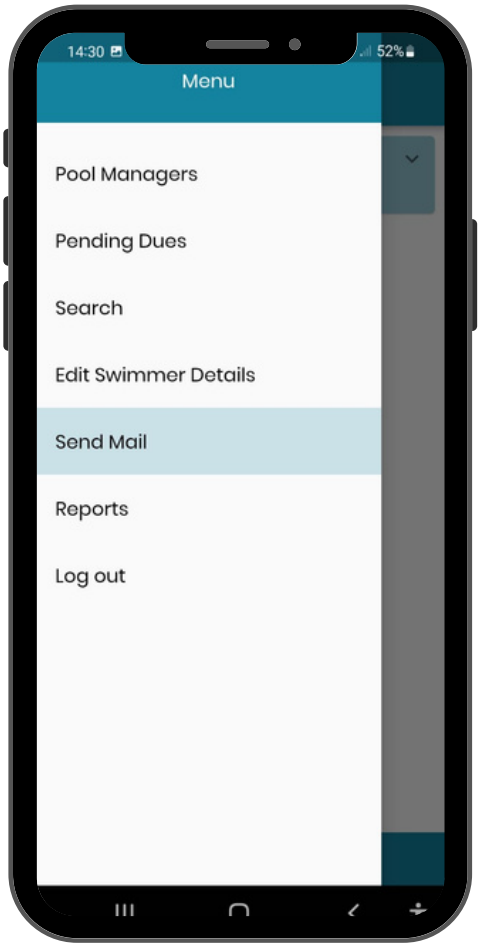


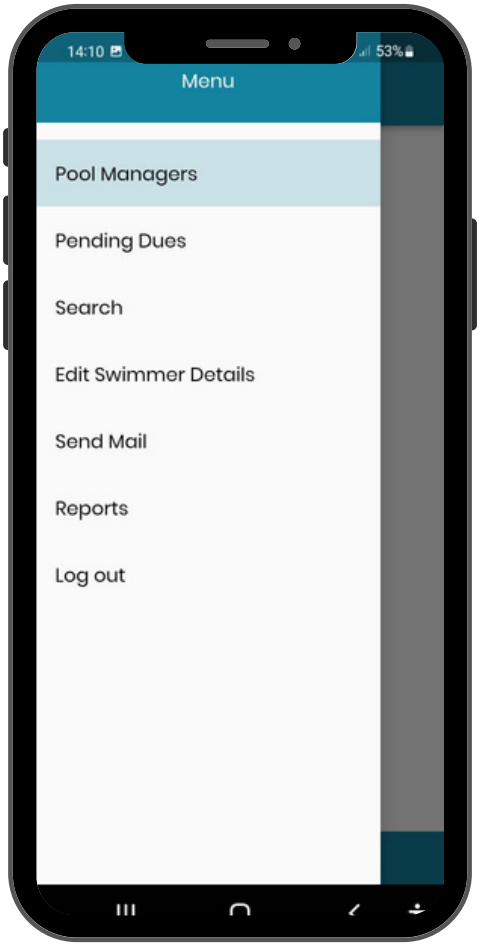
Quarterly Reports

Click on the 'Reports' option to view and download the quarterly collection and visit reports. If you click the 'quarterly report' button, the visit logs of all the swimmers of that quarter will be shown. If you click the 'collection report' button, the money collected as fees in a particular quarter can be viewed.

Send Mail

Click the 'send mail' option to send a mail to all the registered swimmers to inform them about pool maintenance.





Log Out

Both the admin and the swimming pool managers can log out of the app by clicking the 'Log out' option.

Acknowledgement

We would like to express our sincere gratitude to our module owner **Ms. Lekshmy P Chandran** for constantly guiding us throughout the course of this project. Secondly, we would like to thank **Dr. Manjusha K** for teaching us the tools and methodologies of creating various UML diagrams which were indispensable in numerous sections of this document. In addition to this, we take this opportunity to thank **Dr. Vinod Pathari** for playing a crucial role in developing the course plan of this lab course and rendering us the opportunity to proceed with this project.