CSN-254

PROJECT REPORT

GAME-ON

20th April 2023

Submitted by-

Group 4-

- **Rajat Raj Singh.** *Email* - r_rsingh@cs.iitr.ac.in Mob. No.- 8218516522
  Enrollment no. - 21114079
  Contribution- Backend and Realtime database linking

- **Priyanshu Behera.** *Email* - p_behera@cs.iitr.ac.in Mob. No.- 9340371238
  Enrollment no. - 21114077
  Contribution- UI Design

- **Priyansh Mawal.** *Email* - p_mawal@cs.iitr.ac.in Mob. No.- 6378146484
  Enrollment no. - 21114076
  Contribution- UI Design

- **Piyush Arya.** *Email* - p_arya@cs.iitr.ac.in Mob. No.- 9116916870
  Enrollment no. - 21114074
  Contribution- Backend and Realtime database linking

- **Pranavdeep Singh.** *Email* - p_singh2@cs.iitr.ac.in Mob. No.- 8920582347
  Enrollment no. - 21119036
  Contribution- Backend and Realtime database linking

- **Atharv Chhabra.** *Email* - a_chhabra@cs.iitr.ac.in Mob. No.- 6260888533
  Enrollment no. - 21118025
  Contribution- UI Design

Overview:

The purpose of this project is to build an application that lets users book a sports facility at their campus. A need is felt for the automation of this process as it will save time for the users as they don't have to wait in the queue for their turn and also avoid clashes among people. In view of these advantages the General secretary of the sports council as the customer for this sports facility app asked us to build this application. The general secretary of the sports council identified the need for a streamlined and user friendly platform for sports facility reservation and utilization. The end user of the application will be students and faculty of the campus. The application provides users with a dashboard that allows users to select and reserve available facilities at the campus. Users can manage their profile on this platform and leave feedback & reviews about the facility that they used so that the administration of the university can make the changes accordingly.

Software Engineering principles followed:

We analyzed the different software engineering life cycle methods and chose to follow the AGILE methodology due to the following reasons:

- Small-scale project
- Focuses on customer satisfaction
- Helps to remove chances of backlog by scheduling weekly meetings

We created a feasibility document after understanding the requirements of the customer, following which we created an SRS document and submitted it to the customer for it to act as a contract between our organization and the client. We followed the principle of pair programming to remove the possibility of errors and bugs. We performed unit testing for each module and then performed integration testing in a top-down fashion. Finally, we performed extensive system testing to eliminate most bugs from the final deliverable. We conducted weekly catch-up meetings to ensure the development schedule was followed. We created the app incrementally, where each iteration refined the previous one by including better features and functionalities. We informed the client throughout the process and proactively asked for feedback for better customer satisfaction.

Requirements Fulfilled:

Functional Requirements:

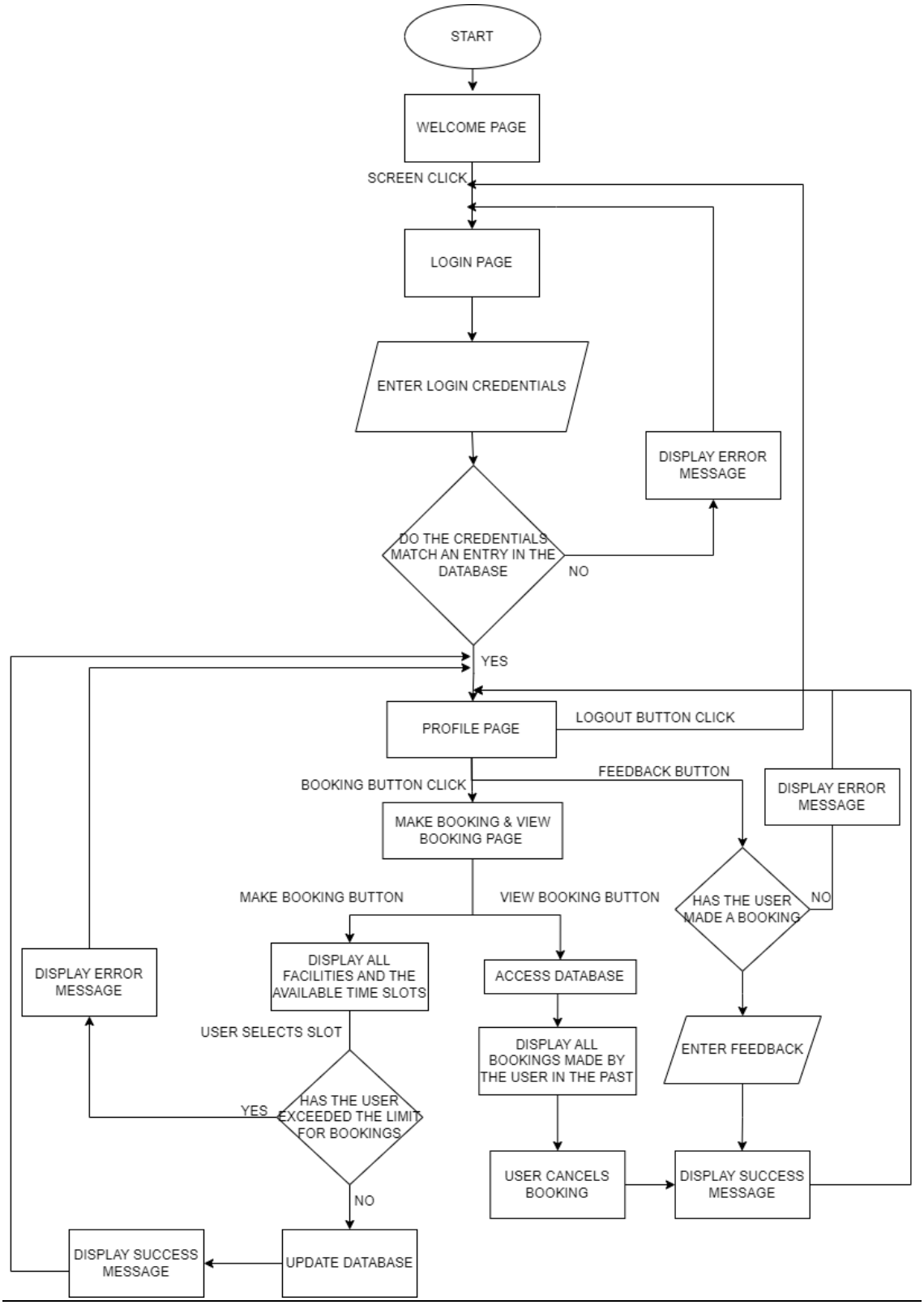
- Login - The users can login to the application using their enrolment number and password. New users are given the option to sign up.
- Change Password - The users are given an option to change their profile's password on their profile page.
- Facility Booking - The user can book various sports facilities such as fields, courts, pools, etc. for a fixed time period of 2 hours.
- Cancel Booking - If the user changes their mind with respect to the booking made, they're given the option to be able to cancel any booking that they've previously made.
- Booking Management - The user is provided with a neat schedule of the facility bookings of the various sporting facilities. The bookings are shown for the upcoming 4 days, including the current day.
- User Profile - Each user has a personal profile page with their enrollment number. It also has the option which allows them to change their profile password.
- Feedback - The users can give their feedback with respect to their experience at the sports facilities that they previously booked. It should be noted that only users who have booked a facility at least once are allowed to give feedback.

Non-functional Requirements:

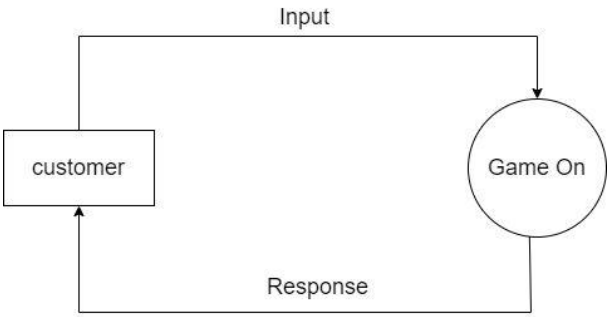
- Performance - The application is able to handle a large volume of users and booking transactions without significant performance issues such as crashes, lagging, etc.
- Reliability - We've ensured that the application is stable and reliable having minimal errors and bugs. Crashes are extremely rare.
- UI - The design of the pages and the overall UI of the application is extremely intuitive and easy to understand and use. The subtle transition animations between pages makes the overall flow of the application fluid. The bookings of the various facilities are displayed in an easy to understand table format. The navigation bar allows the user to easily navigate to a given page of the application.
- Maintainability - The application design is made to be as modular as possible so that it's easy to update the application with new features. The code is properly formatted and the variables are named appropriately.
- Security - The connection to the database is secure and the user information such as their login information and booking details are protected.

Diagrams:

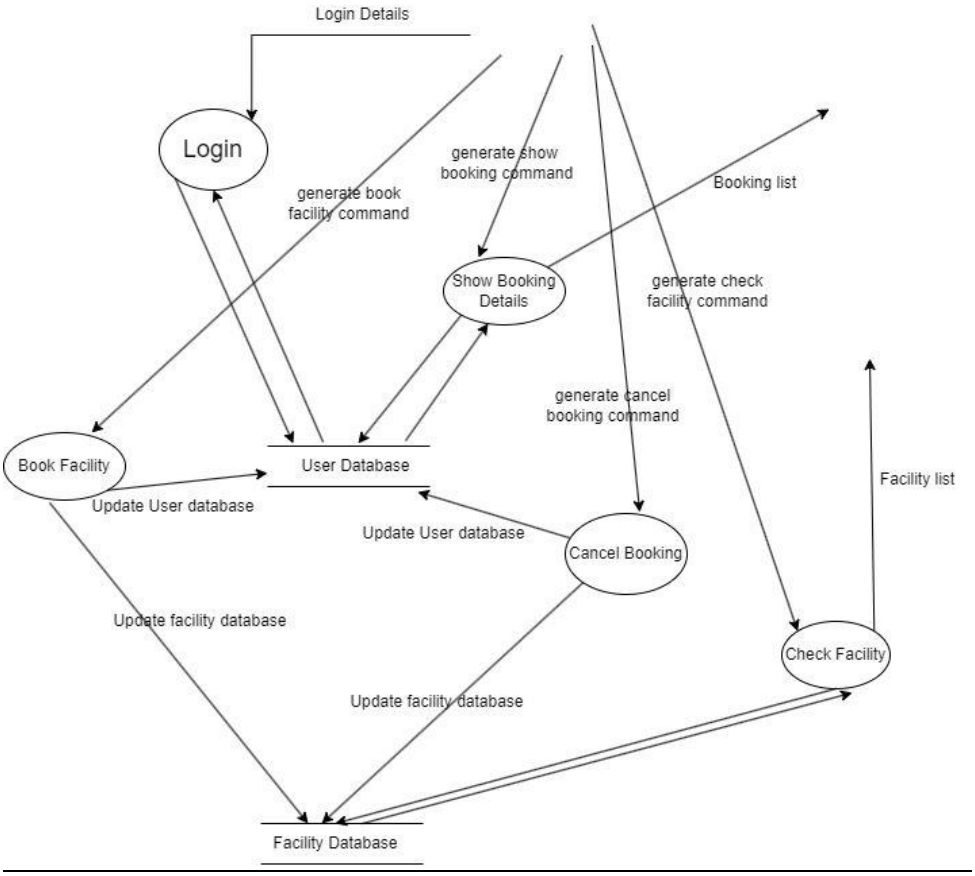
Flowchart:



Data Flow Diagram:

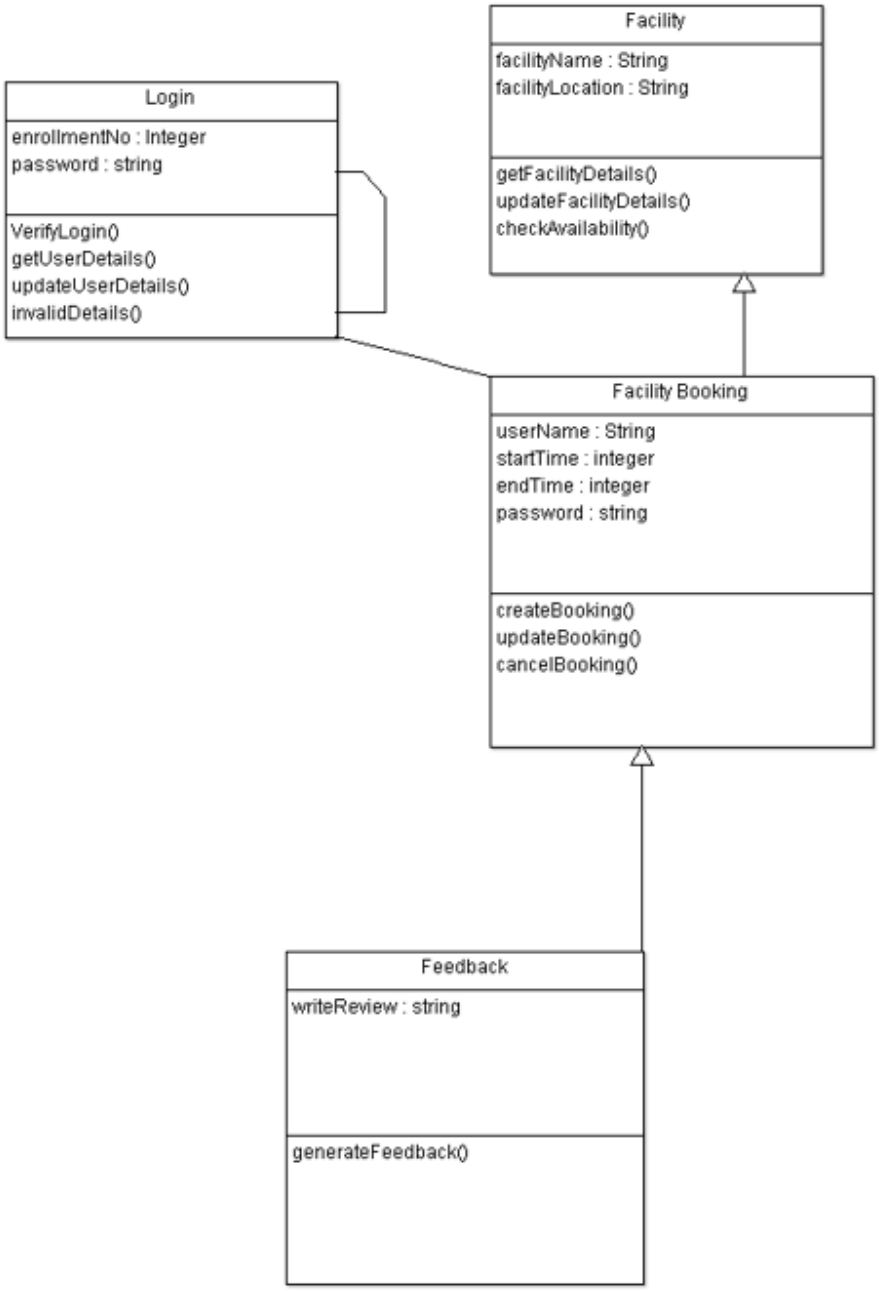


Level 0 DFD Diagram

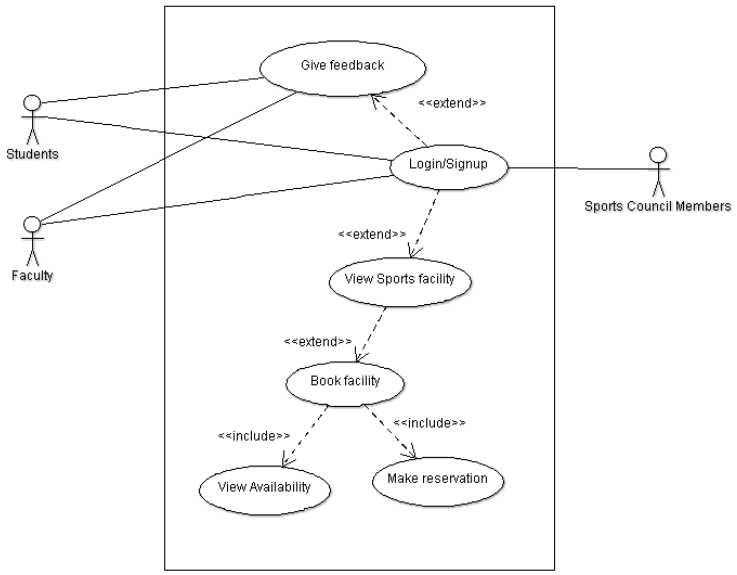


Level 1 DFD Diagram

Class Diagram:



Use Case Diagram:



Important Use Case Descriptions:

Case 1: Book Sports Facility

Mainline sequence-

1- User: Selects View Sports facility option.
2- System: Displays various facilities available in the campus.
3- User: Selects the desired facility.
4- System: Displays the availability status of that facility on different time slots.
5- User: Books the facility.
6- System: Updates the database.

Alternate sequence 1-

After step 3 of mainline sequence:

4- System: Displays that no slots for that facility are available.

Alternate sequence 2-

After step 5 of mainline sequence:

4- System: Displays that the user has already exceeded the maximum booking limit.

Case 2: Cancel Booking

Mainline sequence-

1- User: Selects the View Bookings option.
2- System: Displays bookings of the user.
3- User: Selects the desired facility.
4- System: Deletes the booking and updates the database.

Alternate sequence-

After step 2 of mainline sequence:

3- System: Displays that no cancellable booking is present.

Case 3: Give Feedback

Mainline sequence-

1- User: Selects give Feedback option.

2- System: Opens the write feedback page.

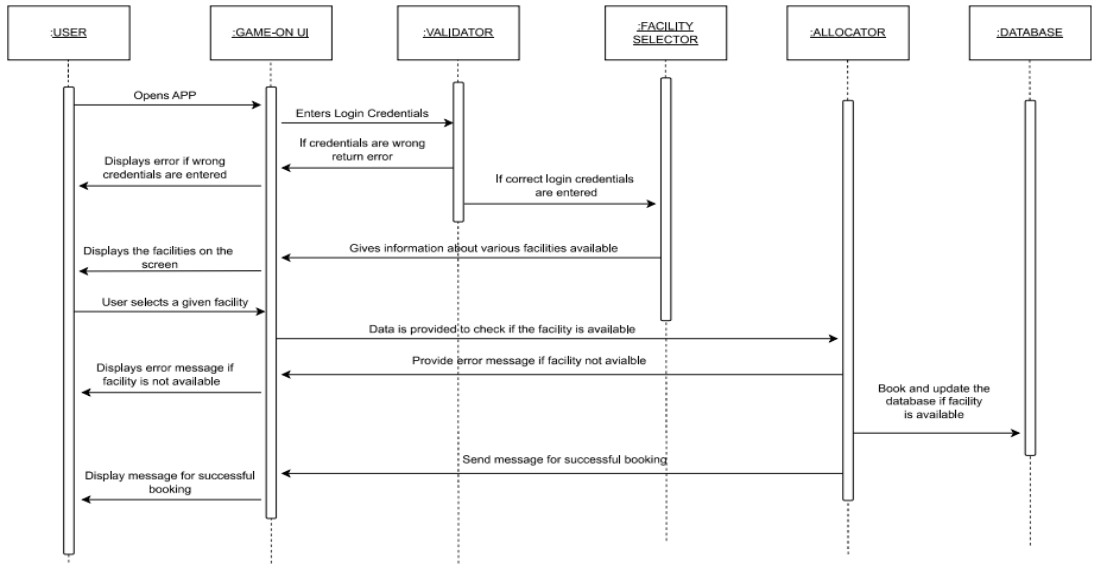
3- User: Writes the feedback.

Alternate sequence-

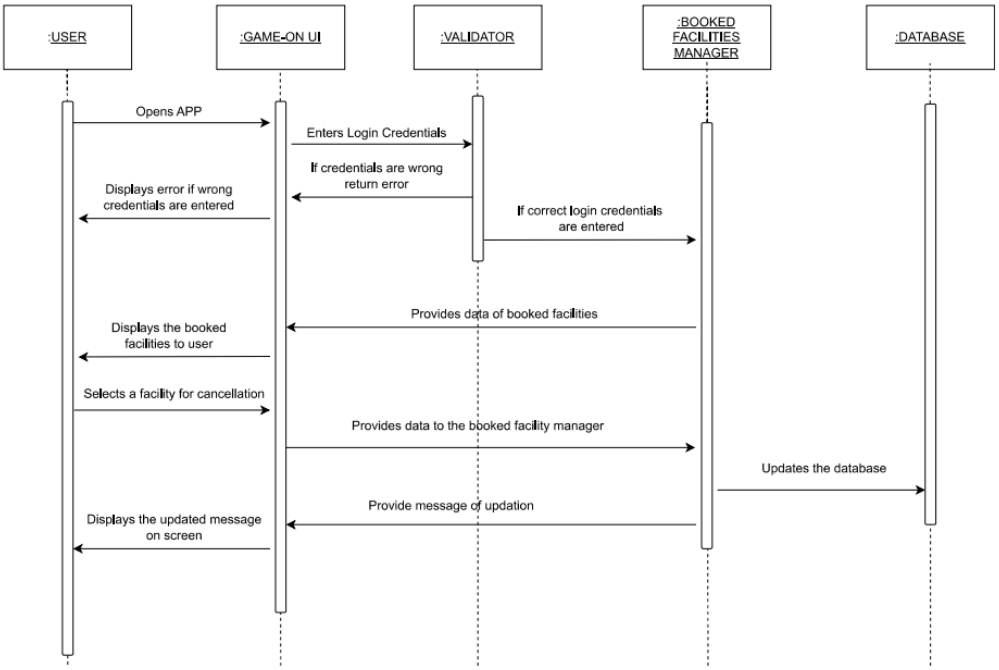
After step 1 of the mainline sequence.

4- System: Displays ineligibility of the user to give feedback due to no prior bookings.

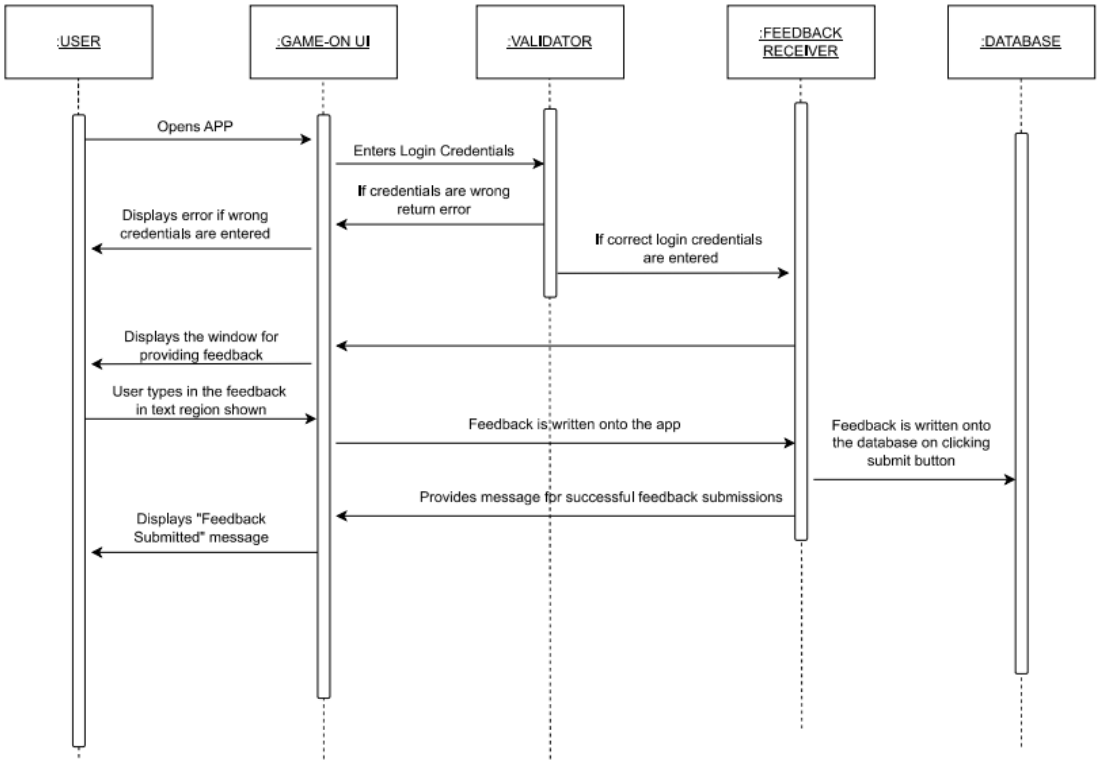
Sequence Diagrams:



SEQUENCE DIAGRAM FOR BOOKING FACILITY



SEQUENCE DIAGRAM FOR CANCELLING A BOOKED FACILITY



SEQUENCE DIAGRAM FOR PROVIDING FEEDBACK

Implementation details:

The major tools used for the development of the application were android studio and figma. First, we created the basic skeleton of the application wherein the different pages were linked, and the basic functionalities were implemented. Then we linked the firebase database to android studio for storing the user's and sports facility's data to update the booking information in real time across different devices. We used intents for linking the pages and passing information like login details, booking details, etc. across different pages. The same was also updated in the firebase database at appropriate times using android studio commands to reflect the changes made by the user in the application. Once all the different pages were identified and linked, we started working on the different use case scenarios. We then synchronized the booking of users in real-time and showed it in a matrix format with the date of the booking. At last, we made some minor changes to optimize our design and make the functioning smooth and streamlined.

The work of the frontend started with designing the background for all the pages of the application. It was accomplished using the online design platform Figma. We used various images and vector assets to give an aesthetic look to the application. To make the application more interactive and lively, we introduced transition effects between the pages. To ease the navigation for the users, a navigation drawer was included in the main page. We then worked on designing the buttons, doing simultaneous color updates of the slots when their state changed. To make the display page regarding the slots for a sports facility more informative, we opted to display the dates of the bookings made. Finally, an appropriate logo for the application was decided after having a collective decision by all the team members. Thus, the front end for the application was dedicated to improving the user experience and hence the usability of the application.

Novelty and Application:

Our application provides the user with an easy way to avoid hassle and save their time by easily booking a given sports facility without any fuss or inconvenience. We've consciously put in effort to make the application's design modular, which makes the code reusable and the application scalable. It can be easily extended to add the higher functionalities which could be needed if the campus chose to extend its sports facilities, such as personalized recommendations based on a machine learning model(which we chose to avoid due to the limited scope of the project). The feedback provided to the facility management helps improve the overall quality of time spent by users on the various facilities. We've carried out real-time database synchronization as well as provided the user with an easy to use and understand interface. We've ensured that there are no clashes in bookings made by distinct users all while providing the user with a pleasing experience while using the application. We've kept the functionalities to the point without including any clutter in the design and interface to avoid wasting the user's valuable time, all while fulfilling the purpose of the application.