

Project Report  
of  
**Software Engineering Laboratory**  
**(CS331)**

**Bachelor of Technology (CSE)**

By  
Kirtan Shah



Department of Computer Science and Engineering  
School Engineering and Technology  
Navrachana University, Vadodara  
Autumn Semester  
(2024-2025)

## Table of Contents

List of Figures

List of Tables

I	Project Description.....	1
1	Project Details.....	1
1a	Project Abstract.....	1
1b	Project Overview.....	1
2	The Purpose of the Project.....	2
2a	The User Business or Background of the Project Effort.....	2
2b	Goals of the Project.....	3
2c	Measurement.....	3
3	The Scope of the Work.....	4
3a	The Current Situation.....	4
3b	Work Partitioning.....	4
3c	Competing Products.....	6
4	Product Scenarios.....	7
4a	Product Scenario List.....	7
4b	Individual Product Scenarios.....	7
5	Stakeholders.....	8
5a	The Client.....	8
5b	The Customer.....	9
5c	Hands-On Users of the Product.....	9
5d	Priorities Assigned to Users.....	11
5e	User Participation.....	11
6	Mandated Constraints.....	12
6a	Solution Constraints.....	12

6b	Implementation Environment of Current System.....	13
6c	Partner or Collaborative Application.....	14
6d	Off-the-Shelf Software.....	15
6e	Anticipated Workplace Environment.....	15
6f	Schedule Constraints.....	16
6g	Budget Constraints.....	18
7	Relevant Facts and Assumptions .....	19
7a	Facts.....	19
7b	Assumptions.....	19
II	Requirements.....	21
8	Product Use Cases.....	21
8a	Use Case Diagrams.....	21
8b	Product Use Case List.....	22
8c	Individual Product Use Cases.....	23
9	Functional Requirements.....	27
10	SDLC.....	29
11	Performance Requirements.....	30
11a	Speed and Latency Requirements.....	30
11b	Capacity Requirements.....	31
11c	Precision or Accuracy Requirements.....	32
12	Dependability Requirements.....	33
12a	Reliability Requirements.....	33
12b	Availability Requirements.....	34
12c	Robustness or Fault Tolerance Requirements.....	35
13	Maintainability and Support Requirements.....	36
13a	Maintenance Requirements.....	36

	13b	Supportability Requirements.....	37
	13c	Adaptability Requirements.....	40
	13d	Scalability or Extensibility Requirements.....	41
	13e	Longevity Requirements.....	42
	14	Legal Requirements.....	43
	14a	Compliance Requirements.....	43
	14b	Standards Requirements.....	44
III		Design.....	46
	15	System Designs.....	46
	15a	Design Goals.....	46
	16	Current Software Architecture.....	48
	17	Proposed Software Architecture.....	51
	17a	Overview.....	51
	17b	Class Diagrams.....	53
	17c	Dynamic Model.....	56
	17d	Subsystem Decomposition.....	61
	17e	Hardware/Software Mapping.....	61
	17f	Data Dictionary.....	63
	17g	Persistent Data Management.....	66
	17h	Access Control and Security.....	68
	17i	Global Software Control.....	70
	17j	Boundary Conditions.....	72
	18	Subsystem Services.....	74
	19	User Interface.....	76
IV		Test Plans.....	81
	20	Features to be Tested/not to be Tested.....	81

21	Pass/Fail Criteria.....	83
22	Test Scenarios.....	85
23	Test Cases.....	86
24	Testing Schedule.....	91
V	Project Issues.....	94
25	Project Planning.....	94
25a	Project Planning.....	94
25b	Planning of Development Phases.....	96
26	Migration to New Product.....	99
26a	Requirements for Migration to New Product.....	99
26b	Data that has to be Modified or Translated for New System.....	101
27	Risks.....	102
27a	Risk Identification.....	102
27b	Risk Assessment.....	105
27c	Risk Projection.....	107
27d	Risk Mitigation, Monitoring and Management (RMMM).....	109
28	Costs.....	111
29	Project Retrospective.....	113
VII	References.....	116

## List of Figures

Figure 1: Use Case Diagram .....	22
Figure 2: Class Diagram .....	55
Figure 3: Sequence Diagram.....	57
Figure 4: Data Flow Diagram Level 0 .....	58
Figure 5: Data Flow Diagram Level 1 .....	58
Figure 6: Data Flow Diagram Level 2 .....	59
Figure 7: Activity Diagram.....	60
Figure 8: ER Diagram.....	66
Figure 9: UI Design for Home Page .....	79
Figure 10: UI Design for Login Page .....	79
Figure 11: UI Design for Sign Up Page.....	80
Figure 12: UI Design for Facility Details Page .....	80

## **List of Tables**

Table 1: Business Events and corresponding Use Cases .....	4
Table 2: Users Table .....	64
Table 3: Facilities Table.....	64
Table 4: Staff Table .....	65
Table 5: Bookings Table.....	65
Table 6: Payments Table.....	65
Table 7: Feature Testing Scope.....	82
Table 8: Pass/Fail Criteria.....	84
Table 9: Testing Schedule.....	92
Table 10: Timetable for Migration to New Product .....	100
Table 11: Risk Assessment Summary.....	107
Table 12: Risk Projection with Potential Impacts and Mitigation Strategies .....	108

# **I Project Description**

## **1 Project Details**

### **1a Project Abstract**

This application is designed to streamline the process of discovering, booking and managing sports facilities, catering to the needs of facility managers, athletes and sport enthusiasts. The application offers a user-friendly solution for the users to search and participate in their choice of sport at the available sport facility, compare prices and amenities and make bookings in real-time. It offers a platform for multiple sports like cricket, football, tennis, golf, pickle ball and more. The platform also enables the facility managers to efficiently manage and optimize bookings and also monitor usage and revenue helping to maximize profits.

### **1b Project Overview**

This project is all about making it easier for people to find, book, and manage sports facilities, whether you're a facility manager, an athlete, or just a sports enthusiast. The goal is to bring everything into one easy-to-use app, where users can browse and book spaces for sports like cricket, football, tennis, golf, pickle ball, and more, all in real-time. Instead of hopping from website to website, users can compare prices, check out the amenities, and secure their spot right from their phone or computer.

For facility managers, the app offers tools to help streamline their operations—whether it's optimizing bookings, tracking usage, or boosting revenue. Managers can manage their facility profiles, update what's available, keep track of real-time bookings, and even view booking schedules. It also helps them handle payments and generate reports on their facility's performance.

The main motivation behind this project is to solve the common pain points with booking sports venues. Right now, people often have to go through the hassle of calling or visiting multiple websites just to find an available spot. This app brings everything together, making it much simpler and quicker for users to discover, compare, and book facilities. Plus,

---



facility managers benefit from better organization and improved resource management, making their jobs easier and their venues more profitable.

The app is designed to be accessible across different devices and integrates smoothly with popular payment platforms, ensuring a seamless and user-friendly experience for everyone involved.

## **2 The Purpose of the Project**

### **2a The User Business or Background of the Project Effort**

With increasing sports clubs & grounds, facilities getting involved in this app will gain more customers by providing easy access to book their games at their own time with the flexibility on their screen remotely. This will help facility managers to manage bookings efficiently on the site and avoid the issue of inefficient resources. It is designed to revolutionize the way athletes, sport enthusiasts and facility managers interact with sport facilities.

The primary motivation behind this project is to address the fragmented and inefficient process of booking sports facilities. Traditionally, finding and booking sports facilities was time-consuming and inconvenient, involving either booking offline or via calls or visiting booking website of the sports facilities' personal website. This application simplifies this problem by providing single integrated solution for real-time booking, easy comparisons, and direct booking options. This project is driven by the need to create more efficient, transparent, and user-friendly approach to manage and book sports facilities.

The challenge of finding and booking a sports facility is a pain for lot of users. The users need to go through multiple websites of each sport facility to book their slots. Athletes and sports enthusiasts often find it difficult to locate available facility in their vicinity and budget. The facility managers also struggle with managing bookings and management tasks. By simplifying the booking solution and providing management tools, the application not only provides user friendliness but also support facility managers to increase their revenue and optimize operations. This need for the solution justifies the importance and relevance of the project.

## **2b Goals of the Project**

The application aims to centralize fragmented sports facilities and helps users to discover and book their choice of facility in real-time. This streamlines the process for both customers and facility managers, reducing manual errors and eliminating the need of back-and-forth communication.

It aims to help users to discover and book sports facility of their choice at their convenience from a single application. The facility owners can get help in managing their bookings, getting discovered and maximizing their profits.

## **2c Measurement**

The goal of the application is to develop a user-friendly application to achieve seamless and efficient booking experience.

The measurable metrics that could support this goal:

1. Increase in Repeat Rate of users: The increasing repeat rate of users in a particular period of time would signify the application being adopted among the users and also shows the customers being pleased with it.
2. Customer Ratings and Review: Higher ratings and positive reviews indicates the users adopting and accepting the application.
3. Revenue from facility bookings: Monitor the revenue generated from the bookings made online via the application and compare it with the offline bookings. An increase in the revenue from online bookings would indicate the application helps in increasing profits and attracting customers.
4. Rate of double bookings or errors in booking: Measuring the frequency of errors in booking process help us indicate the reliability and effectiveness of the system.
5. Operational efficiency: Convenience and time saved by facility managers to schedule, book and manage facility resources.

## 3 The Scope of the Work

### 3a The Current Situation

1. **Manual Booking:** Facilities use manual book-based logs to reserve bookings where staff manually fills out the details. It is prone to errors like overbooking, incorrect details, and is also time-consuming.
2. **Phone call reservations:** Users make call to the facility to make reservations. Facility then manually logs the details into their schedule. The chances of miscommunication are high and can lead to overbooking or incorrect booking and also sometimes the customers need to communicate back-and-forth to keep up with the booking.
3. **Spreadsheets:** Facilities use spreadsheets like excel to manage bookings and availability. It needs to be updated often and lacks real-time updates.
4. **Basic Software Systems:** Facilities have their own software or website to manage bookings. The main issue in this method is that users find it difficult to discover the facility and also don't have the features of comparing various facilities. Many times the websites lack user-friendliness and are cost intensive.

### 3b Work Partitioning

In the context of the Sports Facility Booking Management Application, we identify various business events that influence the system's functionality. Each event corresponds to a discrete use case that contributes to the overall capabilities of the application. The following is a list of key business events along with their details:

**Table 1: Business Events and corresponding Use Cases**

Event Name	Input from Adjacent Systems	Output to Adjacent Systems	Brief Summary of Business Use Case
Search Facilities	User search criteria (e.g., sport type, location, date)	List of available facilities	Users can search for available sports facilities based on their preferred criteria, retrieving a list of suitable venues.

Book Facility	Selected facility, user details	Booking confirmation	Users can select a facility from the search results and proceed to book it by providing necessary details and payment.
Check Facility Availability	Requested booking details (date, time, facility)	Availability status	The system checks the selected facility's availability for the requested date and time before proceeding with the booking.
Process Payment	Payment details and booking confirmation	Payment confirmation	The system handles payment processing through a secure gateway, confirming the transaction before finalizing the booking.
Cancel Booking	Booking reference, user authentication	Cancellation confirmation	Users can cancel their bookings, which will update the facility's availability and inform users of the cancellation status.
Send Reminder Notifications	User booking details	Reminder notifications	The system sends automated reminders to users about their upcoming bookings, reducing no-show rates.
Generate Reports	Administrative requests	Generated report	Facility managers can request various reports (e.g., booking statistics, user engagement) for analysis and operational improvements.
User Registration	User registration details	Registration confirmation	New users can register on the platform, providing necessary details for account creation and access to services.
Update Facility Information	Facility manager updates	Confirmation of updates	Facility managers can update their facility's details, ensuring accurate and up-to-date information for users.
User Feedback Submission	User feedback	Acknowledgment receipt	Users can submit feedback or ratings for facilities, which can be used for service improvement and future user recommendations.

The primary motivation for identifying these business events is to decompose the overall system functionality into logical chunks that can be further analysed and designed. Each

business event represents a specific response the system must handle, allowing us to systematically uncover detailed requirements and organize our analysis. By partitioning the work in this manner, we can streamline the development process and ensure that each component aligns with user needs and business goals.

This structured approach not only aids in managing detailed analysis and design but also enhances communication among stakeholders by providing clear definitions of each use case and its purpose within the system.

### 3c Competing Products

The competing products in the market currently are:

1. Hudle is an application related to sports facility booking management. They do offer valuable functionalities but have its own drawbacks based on user feedback.
  - User Experience Issues: The interface is not user-friendly, users find it difficult to navigate and use to system to maximum efficiency.
  - Shift from the focus: The focus of the system has shifted from helping users to discover and book sports facility to creating a sport culture among youth and more about team sports and tournaments.
2. Khelomore is an application related to catering all sport needs of athletes. It offers marketplace for sporting goods, offers listing for coaching and facilities and help users build connection in the sporting environment.
  - Payment Processing Issues: The payment gateway integration is unreliable and users face frequent errors in performing payments.
  - Mobile Functionality: Khelomore mobile application has limited features as compared to the desktop version which poses as a problem to mobile based users.

While the solutions do exist there is lack of focus on catering to the needs of individual users. The focus of both the solutions is more focused on the facility owners and managers. They lack user-friendliness and doesn't cater to the needs of users. They have many extra features that divert the focus from the goal and make the application complex and inefficient to use.

---

## 4 Product Scenarios

### 4a Product Scenario List

1. Facility booking by an user
2. Facility management by staff
3. Reporting and analysing the use of the facility by the manager
4. Scheduling an event

### 4b Individual Product Scenario

1. Facility booking by an user

Scenario: A tennis player, wants to book a court to play a match with his friend the next morning. He opens the application and logs into his account. On the home page, he selects the location to view the nearby facility listings. He further opts to use the filter to see only the facilities that have tennis courts. He selects the facility of his choice then selects the date and time slot he wants to select. Then he gets a choice of courts that he can select in the facility. After selecting the date, time and court he is then redirected to the payment page. After successful payment he receives an email for successful confirmation of his booking.

2. Facility management by staff

Scenario: The facility staff member, wants to perform maintenance on his courts. He logs into the system and accesses the booking schedule. He checks the calendar to check for the date and time with no current bookings made. He hires the maintenance crew to arrive at the time with no bookings made. He then inputs the time slots on that day as booked so that no other further reservations are made at that time period.

3. Reporting and analysing the use of the facility by the manager

Scenario: The facility owner, wants to make a monthly report on facility usage and revenue. He logs into the system and checks the total time the facility was booked in the past month, the total revenue generated for the whole month, and also the time of the day that the facility was booked the most for. Such analytics helps him to create a monthly report that helps the team in strategic planning for the upcoming month.

#### 4. Scheduling the event

Scenario: An event manager, wants to host a tournament event at a sports facility. She logs into the application and browses for the facility that has courts large enough for her event to be held. After selecting the facility from the list of facilities from the home page, she checks for the availability of the facility on the dates she is comfortable with. After selecting the dates, time and payment, she will get automated email response from the application of successful booking. Using this application, she could avoid clashes and schedule her event on time comfortable to her and also select the facility that would provide the services necessary for her event.

## 5 Stakeholders

### 5a The Client

The client for this Sports Facility Booking Management System would typically be an organization or enterprise involved in the sports infrastructure sector. This may include sports infrastructure companies, sports facility operators, or organizations managing multiple sports venues. Such clients are usually invested in providing seamless booking and management solutions to enhance user experience, streamline operations, and increase revenue through optimized facility use.

These clients generally lead in sports infrastructure development, installation, and maintenance, acting as pivotal contributors to the sports ecosystem. They specialize in establishing and maintaining high-quality sports facilities and are now looking to integrate advanced management systems to serve athletes, sports enthusiasts, and facility managers effectively.

These clients are well-established and deeply connected within the sports industry, often with numerous facilities across various regions. Their goal is to expand into the sports facility booking management market alongside infrastructure development. The Research and Development teams within these organizations oversee project implementation, ensuring the final product aligns with their operational and user-experience goals, ultimately having the authority over product acceptance. The system must therefore meet rigorous quality and functionality standards to secure their approval.

---

## 5b The Customer

The customers for this product would be:

- **Sports Facility Operators:** Sports facility operators who manage and operate sport facilities like courts, turfs, courses or grounds.
- **Local Clubs and Associations:** Community clubs that run and operate different sports at their sport facility
- **Resorts or Hotels:** Resorts or Hotels who have sport facilities like pool, golf courses, courts or turfs and want to rent them separately.
- **Educational Institutes:** Educational Institutes that have sport facility vacant during the non-operational hours can rent them separately.

They need efficient management tool for scheduling, booking and maintaining their facilities. They are looking for platform that simplifies the booking process, manages usage and time and also provides discoverability among new users.

## 5c Hands-On Users of the Product

The hands-on users of the product will be:

### **Facility Managers:**

- **User Role:** Manage and check day to day bookings, maintain the facility and co-ordinate with customers and staff.
- **Subject Matter Experience:** Master – has deep understanding of the business and understands the facility booking process.
- **Technological experience:** Novice – in most cases is only familiar to basic software and may require training to adopt the new features.
- **Intellectual abilities:** It would be beneficial if he has customer support and customer relation skills.
- **Attitude towards Job:** Professional and should be focused on efficiency
- **Attitude towards technology:** Should positively accept technology and use it to efficiently manage the facility.
- **Education:** Should have basic knowledge about the sports.
- **Linguistic Skills:** Fluent with the primary language of the region

Users are facility managers who use the application for managing day to day operations like scheduling the bookings and managing the facility and check for availabilities or

---



vacancies. A facility manager needs the application to overlook the performance of the facility, overlook the revenue and profits and work on the reviews and ratings left by the users to make the facility better.

Administrative staff needs to use the application to manage the bookings. The staff would need to prepare the facility before the customer arrives, confirm with the customers regarding their booking.

**Individual Users (Athletes/ Sport Enthusiasts/ Young adults):**

- User Role: Book and use sports facilities for personal training or recreational purposes.
- Subject Matter Experience: Novice – Basic knowledge about facility usage and sport activities.
- Technology Experience: Novice – Should be comfortable with online booking system and digital payment systems.
- Attitude towards technology: Positive, should adopt to online booking system.
- Age group: Broad, including children to senior players (15 to 35)

Users are athletes, sport enthusiasts or players who use the application to conveniently discover and book sport facility who like to play sports for training, recovery or recreational purposes. Like

- Users are students, sports enthusiasts, sports players, teenagers, corporate employees or event planners.
- Students aspiring to learn or play different sports can access the application to discover and book the facility of their choice.
- Sport Enthusiast who like to play sports in their free time can search for different facilities to book and have a fun time.
- Sport players who play to train or recover for some tournament can frequently book facility with ease with the help of application.
- Teenagers who like to connect with friends and meet new people can book facility and play with their friends.
- Corporate Employees from big corporations are usually involved in activities like team building and recreational sports to maintain the work balance at the office.

- 
- Event Planners who are looking for a venue to host their events can browse through the application and book the facility that meets all their needs.

### **5d Priorities Assigned to Users**

- **Key Users:** Facility Managers and Athletes are the key users of the application. Highest priority is to be assigned to the facility managers as they are responsible for smooth operation of facility and affect the overall performance of the user experience. Also high priority is assigned to athletes as they have higher influence on the facility's reputation and these group of users have the highest usage and repeat booking rates.
- **Secondary Users:** Administrative Staff and users like students, teenagers and sports enthusiasts are secondary users of the application. Secondary priority is assigned to the administrative staff as they manage the day to day operation of the facility. Other users like students, teenagers and sport enthusiasts are also assigned secondary priority as they use the facility less frequently and also their requirements are much less than of athletes. The feedback of this users is important but not the core of the system.
- **Unimportant Users:** Casual Users or corporate employees are the low priority users as they use the application very less frequently. Their usage is low and also the repeat business from them is usually in long cycles with large time gaps.

### **5e User Participation**

User Participation for the product will be:

- Facility Manager to provide with deep knowledge of the business and current workflows, also mentioning the additional requirements for the application and the major problems and errors faced in maintaining the facility. Around 15-18 hours over the course of project would be required.
  - Administrative Staff to provide with input on features concerning booking management, user support and user behaviour. Also participate in reviewing and validating system design and usability and provide feedbacks on prototypes. Around 8-10 hours over the course of project would be required.
  - Athletes to provide with insights into the facility usage needs such as requirements at facility, scheduling preferences and psychology behind selecting
-

the correct facility. Provide with insights on specific needs useful in booking system for sport facilities. Around 7-10 hours during the feedback and testing phase would be required.

- Students to provide with feedback on ease of using the application and also to check the features of booking process and participate in the user testing phase. Around 4-5 hours would be required during the feedback and testing phase.
- Casual Users to provide with feedback on the overall user experience and booking process. Around 2-3 hours for the testing sessions would be required.

User participation help in defining the user requirements and helps in keeping the project around the needs of the users. With user participation during every phase of development, the rate of user adopting the project in later stages is greatly increased. With requirements made clear from the initial stages, the resources must be collected to allocate to the project.

## **6 Mandated Constraints**

### **6a Solution Constraints**

- **User Authentication:** Implementing robust user authentication mechanisms is essential for securing user data. This includes multi-factor authentication (MFA) to enhance security, email verification to confirm user identities, and password recovery options to assist users in regaining access to their accounts in case of forgotten passwords. This multi-layered approach ensures that user data is protected against unauthorized access and enhances trust in the platform.
- **Performance Limits:** The system must be designed to efficiently handle a specific number of concurrent users—such as 500—without experiencing performance degradation. This involves optimizing the app's architecture to manage user requests effectively, minimizing response times, and ensuring smooth operation under peak loads. Load testing should be part of the development process to identify potential bottlenecks.
- **Functional Requirements:** The app should offer a comprehensive suite of features that cover all aspects of sports booking. This includes booking management for users to easily create, modify, and cancel reservations; payment processing to

facilitate secure financial transactions; user profile management allowing customization and storage of user preferences; and facility management to enable administrators to manage resources effectively. Each feature should be intuitive and seamlessly integrated to enhance the user experience.

- **Compliance:** It is crucial to adhere to relevant regulations, such as the General Data Protection Regulation (GDPR), especially when handling user data in regions with stringent privacy laws. This includes implementing policies for data collection, storage, and processing, ensuring users have rights over their data, and maintaining transparency about how their information is used. Regular audits and updates to compliance measures should be planned as regulations evolve.
- **Accessibility:** To cater to users with disabilities, it is important to incorporate accessibility features into the design. This includes adherence to the Web Content Accessibility Guidelines (WCAG), ensuring that the application is usable for everyone, including those with visual, auditory, or motor impairments. Features such as screen reader compatibility, text resizing, and alternative text for images will enhance usability and compliance with accessibility standards.

## **6b Implementation Environment of the Current System**

- **Development Tools:** Utilizing Android Studio as the primary Integrated Development Environment (IDE) will provide the necessary tools for efficient app development. It supports a variety of programming languages, including Java and Kotlin, and offers features such as code editing, debugging, and testing tools, which are crucial for building, testing, and deploying the application.
- **Operating System:** The application should be rigorously tested on the latest android versions, specifically targeting Android 8.0 and above. Ensuring compatibility with older versions is also important, as this broadens the potential user base. Testing across various devices with different screen sizes and hardware specifications will ensure a consistent user experience regardless of the device used.
- **Database Solutions:** Choosing the right database solution is crucial for the application's performance and scalability. SQLite can be used for efficient local data storage, allowing for quick access and updates without requiring a constant internet connection. Alternatively, Firebase offers a cloud-based solution that can

scale easily as user numbers grow, providing real-time database capabilities and reducing the burden on local storage.

- **API Integration:** The application should be designed to integrate seamlessly with external APIs for essential functionalities, including payment processing and location services. This integration will enhance the app's overall functionality, enabling features such as secure payment transactions and the ability to locate nearby facilities or events, thereby improving the overall user experience.
- **Development Hardware:** Developers should have access to a range of physical devices and emulators for comprehensive testing. This diversity will help identify any compatibility issues and ensure the application runs smoothly across different hardware specifications and screen sizes, which is vital for delivering a consistent user experience.

## **6c Partner or Collaborative Applications**

- **Payment Gateways:** Forming partnerships with reputable payment gateway services like Stripe or PayPal will enable secure and efficient payment processing. This not only ensures that transactions are handled securely but also offers users a variety of payment options, enhancing their overall experience.
- **Booking Systems:** Collaborating with existing booking platforms can enhance the app's functionality and streamline the booking process. Integrating with established systems may also provide additional features and capabilities that can improve user experience without the need to develop everything from scratch.
- **Social Media Integration:** Incorporating features that allow users to share their bookings or events on social media platforms such as Facebook and Twitter can enhance visibility and engagement. This not only promotes the app but also creates a community around it, encouraging users to interact and share their experiences.
- **Analytics Providers:** Utilizing analytics tools like Google Analytics or Firebase Analytics will facilitate real-time tracking of user interactions and app performance metrics. These insights can help inform decision-making, allowing for adjustments to be made based on user behaviour and preferences, ultimately leading to a more tailored user experience.

- **Email Services:** Integrating with reliable email service providers like SendGrid or Mailgun for managing notifications, confirmations, and updates will ensure that users are kept informed about their bookings and any changes. Effective communication is crucial for user satisfaction and engagement.

## **6d Off-the-Shelf Software**

- **UI Libraries:** Leveraging established UI libraries such as Material Components will help create a cohesive and responsive user interface. This consistency not only enhances the aesthetic appeal of the app but also improves usability, making it easier for users to navigate and interact with the app.
- **Networking Libraries:** Implementing networking libraries like Retrofit or Volley for efficient network requests will facilitate smooth data handling and enhance communication between the app and external services. These libraries simplify the process of managing network calls, reducing development time and improving reliability.
- **Database Libraries:** Utilizing Room as a database management library for SQLite will streamline CRUD (Create, Read, Update, and Delete) operations and improve data integrity. Room abstracts away much of the complexity of SQLite, making it easier to work with the database while maintaining performance.
- **Testing Frameworks:** Employing automated testing frameworks such as JUnit and Espresso will ensure code reliability and validate functionality. Automated testing reduces the likelihood of bugs and issues slipping into production, thereby enhancing the overall quality of the application.
- **Crash Reporting:** Integrating monitoring services like Crashlytics will allow for real-time capture and analysis of app crashes and errors. This will facilitate prompt resolution of issues, improving app stability and user satisfaction over time.

## **6e Anticipated Workplace Environment**

- **Team Structure:** Establishing a multidisciplinary team that includes developers, UX/UI designers, project managers, and quality assurance specialists will foster collaboration and leverage diverse skill sets. A well-rounded team can tackle various challenges and contribute to a well-rounded product.

- **Development Methodology:** Adopting Agile methodologies for project management will enable iterative development and adaptive planning. Regular feedback loops through structured sprints will allow the team to make adjustments based on user feedback and evolving requirements.
- **Collaboration Tools:** Utilizing communication tools such as Slack for team interactions and GitHub for version control will enable collaborative coding and efficient project tracking. These tools enhance team coordination and transparency, which are critical for successful project outcomes.
- **Work Location:** Implementing a hybrid work model that offers both in-office and remote work options can accommodate team preferences and promote a flexible work environment. This flexibility can improve job satisfaction and productivity among team members.
- **Meeting Cadence:** Establishing regular stand-up meetings, sprint planning sessions, and retrospectives will ensure ongoing alignment among team members and facilitate effective progress tracking. These meetings provide opportunities for feedback and adjustments, keeping the project on track.

## 6f Schedule Constraints

### Project Timeline

- **Objective:** To maintain a structured development flow, a detailed project timeline will outline key milestones and their target completion dates for each phase of development.
- **Milestones:**
  - Design Completion: November 30, 2024
  - Phase 1 (Core Functionality): December 31, 2024
  - Phase 2 (Payment Integration & Advanced Booking): February 28, 2025
  - Phase 3 (Notification System & UI Enhancements): April 15, 2025
  - Phase 4 (Reporting & Admin Dashboard): June 15, 2025
  - Phase 5 (Final Testing, Deployment & User Training): July 31, 2025
- **Updates:** Timeline will be revisited and updated bi-weekly to reflect progress, any scope adjustments, and potential rescheduling needs.

---

## Testing Schedule

- Objective: To allocate dedicated time for testing phases to ensure a thorough evaluation of the application at each stage of development.
- Phases:
  - Alpha Testing: December 1 – December 6, 2024 (Core Functionality Testing)
  - Beta Testing: February 15 – February 28, 2025 (Advanced Features & Payment Integration)
  - User Acceptance Testing (UAT): June 1 – June 15, 2025 (Full System Testing)
- Responsibilities: Testing will involve designated QA teams, with additional user involvement for UAT to gather feedback and resolve any final issues before launch.

## Launch Timing

- Objective: To optimize user engagement by aligning the launch with high-demand sports seasons.
- Planned Launch Date: August 1, 2025
- Rationale: Aligning with the upcoming cricket and football seasons in India for maximum initial user engagement and adoption.

## Resource Availability

- Objective: To manage team resources effectively by scheduling around team members' availability, taking into account holidays and prior commitments.
- Resource Constraints:
  - Development & Testing Teams: Full availability required from December 2024 to August 2025.
  - Adjustments: Resource availability to be reassessed monthly, and schedules will be adjusted to account for any upcoming leaves or holidays.

## Contingency Planning

- Objective: To incorporate buffer time for handling unforeseen challenges and ensure project flexibility.
  - Buffer Time Allocation:
    - Phase 1 - Core Functionality: 1-week buffer ending January 7, 2025
-



- 
- Phase 2 - Payment & Booking Features: 1-week buffer ending March 7, 2025
  - Phase 3 - Notification & UI Enhancements: 1-week buffer ending April 22, 2025
  - Phase 4 - Reporting & Analytics: 1-week buffer ending June 22, 2025
  - Review Frequency: Contingency planning reviewed monthly to address and mitigate any emerging risks.

## **6g Budget Constraints**

- Development Costs: Clearly defining a budget for hiring developers, UX/UI designers, and other essential team members is crucial. This budget should also include provisions for potential overtime or additional resources needed to meet project demands.
- Licensing Fees: Account for any licensing fees related to third-party libraries, APIs, and cloud services in the overall budget. Ensuring that all necessary costs are incorporated will prevent financial shortfalls during the development process.
- Marketing Budget: Allocating sufficient funds for post-launch marketing efforts, including social media advertising, promotional events, and outreach campaigns, is essential for maximizing user acquisition and retention. A strong marketing strategy will enhance the app's visibility and user engagement.
- Operational Costs: Factoring in ongoing operational costs for server hosting, domain registration, and maintenance activities following the app's launch is critical for sustainable financial planning. This ensures that the app remains operational and up-to-date.
- Risk Management: Setting aside a contingency budget (typically 10-15% of total costs) to address unexpected financial needs or challenges that may arise during the project lifecycle is essential. This financial cushion can provide peace of mind and enable quick responses to unforeseen circumstances.

---

## 7 Relevant Facts and Assumptions

### 7a Facts

Facts that affect the products but are not necessarily required provide context and help shape the design and development of the product. Like the current system already in use, the user base, operational hours, historical data to analyse trends and integration needs.

Having access to relevant facts help the stakeholders understand the factors that influence the project. These help in shaping the strategies required develop the project. Like:

- **Current System:** Many facilities currently use combination of some software and manual bookings to operate the facility.
- **User Base:** The user base also includes users from varied age group with different level of technology awareness.
- **Operational Hours:** Facilities across different locations have different operational hours ranging from early morning to midnight hours.
- **Historical Data:** Historical data of the facility affects the strategy to utilize and efficiently use the software for the facility. It helps in analysing the trends and revenue.
- **Integration:** Payment Gateways will be needed to integrate with current payment system of the facility.

### 7b Assumptions

- **Technological Setup:** It is assumed that the users will have access to internet connectivity and software suitable for accessing the software.
- **User Training:** It is assumed that users will require minimal time and training to adapt and use the application as it will be designed to be user-friendly.
- **Data Transfer:** It is assumed that the previous and historic data can be transferred from existing manual records to new system easily with minimal data loss.
- **Availability of APIs:** It is assumed that third-party APIs like map and calendar will be available to the user.
- **Integration:** It is assumed that new system will be easily integrated to the existing systems used by the facilities.

- 
- Maintenance: It is assumed that support team will be able to handle user issues and maintenance after the deployment phase.
  - Payment Processing: It is assumed that the facility will have payment processing systems that can be easily integrated with application.

It is declared to make everyone aware of the assumptions already made and provide a solid foundation for the project to ensure that all stakeholders have shared understanding of the constraints affecting the development of sports facility management application. The considerations for the product assumed are:

- Technical setup such as a tablet or mobile phone should be available supporting either IOS or Android OS.
- Location services such as the facility should be registered location on google maps.
- Banking needs like business bank accounts, UPI account and should be certified by payment processor for card payments.

## II Requirements

### 8 Product Use Cases

This section describes more specific and precise detail exactly what steps the system takes in the course of its performance.

#### 8a Use Case Diagrams

**Actors:**

- **New User:** Represents a person who has not yet registered in the system. They can register to access the system.
- **Register User:** Refers to a user who has already registered and has access to the system. This user can perform various actions like booking facilities, viewing booking history, and cancelling bookings.
- **Facility Manager:** This actor manages the facility, including updating facility info, handling bookings, and generating monthly reports.
- **Administrative Staff:** Responsible for administrative tasks such as supporting users and maintaining the booking timetable.

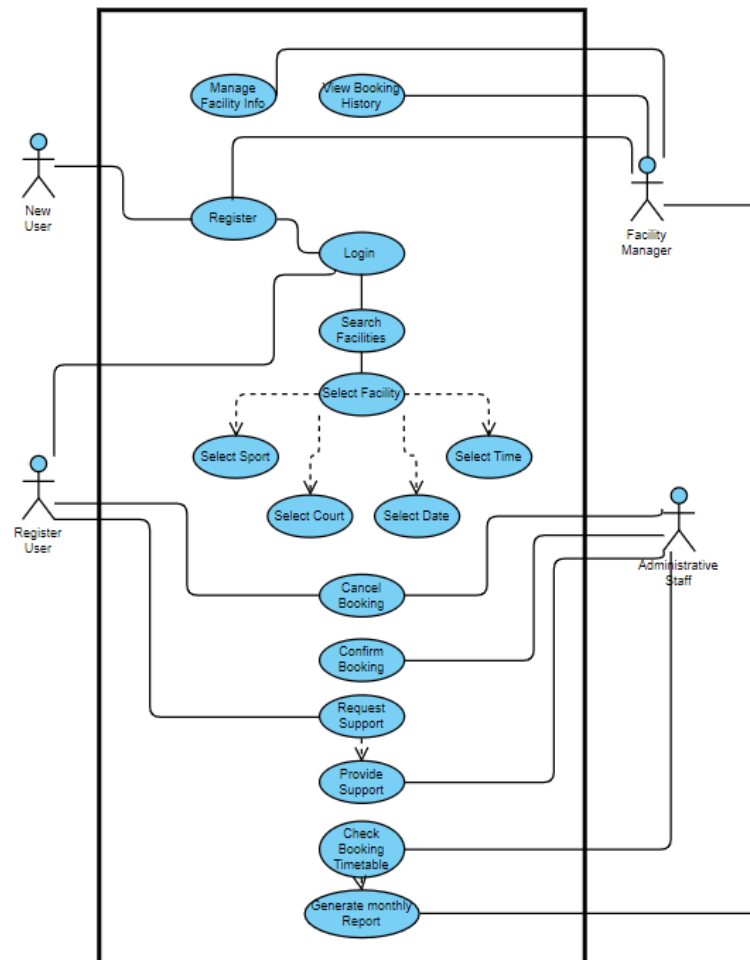


Figure 1: Use Case Diagram

## 8b Product Use Case List

- User Registration
- User Login
- Search Facilities
- View Facility Details
- Book Facility
- Cancel Booking
- View Booking History
- Manage Facility Information
- Request Support
- Provide Support
- Generate Reports

---

## 8c Individual Product Use Cases

- **Use case name:** User Registration

Participating actors: New Users

Flow of events:

1. The new user opens the application
2. The application responds to the user requesting either login details or to register as new user and presents a registration form to the user.
3. The new user completes the form by filling the details like name, email ID, contact number, password, gender, date of birth and age. Once the form is completed, the user submits the form.
4. The application receives the form and registers the user with provided details. The application displays the acknowledgement of successful registration.

Entry condition: The user doesn't have his email ID or contact number already registered.

Exit condition: The new user is successfully register and has a new account created for him.

Quality requirements: Detect repeated users registering to create multiple accounts.  
Detect false information being added.

- **Use case name:** User Login

Participating actors: Facility Managers, Individual Users

Flow of events:

1. User opens the login form.
2. User fills the login details like email ID or contact number and enters the password.
3. The application confirms whether the details entered are correct or not.
4. After successful confirmation from the system, the user is logged into his account on the application.

Entry condition: The user is already registered.

Exit condition: The user enters correct credentials and is logged into his account.

Quality requirements: Notifies the user via mail after 5 unsuccessful attempts to log in.

---

- **Use case name:** Search facilities

Participating actors: Individual users

Flow of events:

1. The user opens the application and browses the facilities.
2. The user then applies filters necessary to him like location, sport types, available amenities to filter down to the best available option to him.

Entry condition: The user must select the city to view the facilities available.

Exit condition: The user must be displayed list of facilities.

Quality requirements: The user must be displayed the list of facilities according to the filters applied.

- **Use case name:** View facility details

Participating actors: Individual Users

Flow of events:

1. The user selects a particular facility.
2. After landing on the details page of the facility, the user can check for the sports available, amenities available, description about the facility and ratings and reviews left the users can also be visible.
3. Availability to book the facility can also be checked here.

Entry condition: The user musts select a particular facility

Exit condition: The user has received details and information regarding the facility.

- **Use case name:** Book Facility

Participating actors: Individual Users

Flow of events:

1. The user selects a facility they want to book.
2. The application presents the available time slots for the chosen facility.
3. The user selects a time slot and confirms the booking.
4. The application processes the booking, and the user provides necessary payment details if required.
5. The application confirms the booking and sends an acknowledgment to the user via email or app notification.

Entry condition: The user is logged in and has selected a facility to book.

Exit condition: The user has successfully booked the facility and received confirmation.

---

- **Use case name:** Cancel Booking

Participating actors: Individual Users

Flow of events:

1. The user views their upcoming bookings.
2. The user selects the booking they wish to cancel.
3. The application prompts the user for confirmation before proceeding.
4. Upon confirmation, the booking is cancelled, and a confirmation notification is sent to the user.

Entry condition: The user has an existing booking that they wish to cancel.

Exit condition: The booking is successfully cancelled, and the user receives confirmation.

- **Use case name:** View Booking History

Participating actors: Individual Users

Flow of events:

1. The user logs into their account and navigates to the booking history section.
2. The application displays the user's previous and upcoming bookings with details such as date, time, and facility information.

Entry condition: The user has successfully logged into their account.

Exit condition: The user can view the history of their bookings.

- **Use case name:** Manage Facility Information

Participating actors: Facility Managers

Flow of events:

1. The facility manager logs into their account and navigates to the facility management section.
2. The manager edits details such as facility description, available amenities, pricing, or time slots for booking.
3. The application updates the facility information in real time.

Entry condition: The facility manager is logged in with the necessary privileges.

Exit condition: The facility information is successfully updated and saved in the system.

- **Use case name:** Request Support

Participating actors: Individual Users

Flow of events:

---



1. The user opens the support section in the application.
2. The user fills out a support request form, providing details about their issue or inquiry.
3. The user submits the request.
4. The application acknowledges receipt of the request and assigns a ticket to the issue.

Entry condition: The user has encountered an issue or needs assistance.

Exit condition: The user receives acknowledgment of their support request with a reference number.

- **Use case name:** Provide Support

Participating actors: Support Agents

Flow of events:

1. The support agent accesses the list of submitted support requests.
2. The agent selects a support ticket and reviews the user's issue.
3. The agent communicates with the user via the application's support messaging system or email.
4. Once resolved, the agent closes the ticket and informs the user.

Entry condition: The support agent is logged in and viewing active support tickets.

Exit condition: The support issue is resolved, and the ticket is closed.

- **Use case name:** Generate Reports

Participating actors: Facility Managers

Flow of events:

1. The facility manager accesses the reporting section in their account.
2. The manager selects the type of report they want to generate (e.g., booking statistics, revenue reports, user activity).
3. The application generates the requested report, which can be viewed or downloaded.

Entry condition: The facility manager has access to reporting features and necessary data.

Exit condition: The requested report is generated and available for the facility manager.

---

## 9 Functional Requirements

### 1. User Registration and Login

- Account creation: User must be able to create an account by entering details like name, email, contact number, password, DOB and gender.
- Login: User must be able to log into their account by entering valid username and password.
- Forget Password: User should have option to reset their password if they forget theirs.

### 2. User Profiles

- Edit Profile: Users must be able to view and edit their information like contact number, email, password or location.
- Facility Profile Page: Facility manager must be able to manage his facility profile by adding description, amenity details and needful information.

### 3. Facility Management

- Facility Listings: Displaying a list of available sports facility with its details like name, location, average prices, amenities and available sports.

### 4. Booking Management

- Facility Booking: Users should be able to search and book facilities on the criteria of type of sport, date, time and available court.
- Real-time availability: Show users if facility is available at the selected time to prevent double-bookings.
- Booking confirmation: Users should be sent a confirmation via email to update them on the booking status.
- Booking cancellation: Users should have the feature of cancelling the booking within a time frame.

### 5. Payment

- Payment Integration: Integration with payment gateway to manage the online payments made on the application.
- Payment confirmation: Users should be sent payment confirmation to update them of the booking status.

## 6. Searching and Filtering

- Searching features: Users should be able to search for facilities based on location, type of sport, availability or other criteria.
- Filter options: Users should be provided with filter search options to filter the facility listing based on type of sports, availability, price range, amenities and others.

## 7. Booking Schedules

- Schedules: Users (facility managers) should be able to check the booking schedules for the day in a timetable format.

## 8. Admin Management

- Facility Rules: User should be able to add and update booking rules for the facility, payment settings and adjust terms and conditions specific to the facility.
- Facility management: User should be able to manage hours and days the facility might be unavailable and close the bookings in case of unforeseen circumstances.

## 9. Reports and Analytics

- Monthly reports: User should be able to check the monthly reports including payment details, payment modes and monthly revenue.
- Usage analytics: User should be able to generate the analytic report on usage hours, peak hours, booking trends and user activity.

## 10. Customer Support

- Help: users should have access to FAQ's, guides and troubleshooting guides.
- Agent Support: User should be able to contact to the support team with the help of email, chat or phone call.

## 11. Access

- Design: The system should be designed to be accessible on different devices.

## 12. Integration

- Payment Integration: Users should be allowed to integrate and perform payments with the help of external application like Gpay or Paytm.

## 10 SDLC Model

### **Agile Model:**

The Agile Model is the most suitable approach for this project. Its flexibility and iterative nature allow for incremental delivery of small, functional parts of the product. It promotes adaptability to changing requirements, which is crucial for this project, as user needs and market trends may evolve over time. Agile's emphasis on continuous feedback and regular collaboration with stakeholders ensures alignment with user expectations. However, this model requires constant user involvement and can pose challenges in maintaining the scope, managing user requirements, and meeting deadlines due to the frequent adjustments.

### **Waterfall Model:**

The Waterfall Model is a linear and sequential development approach with clearly defined phases. While it is straightforward and works well when the requirements are well-understood from the start, it may not be the best fit for this project. Given that the project's user needs could evolve and trends may change, the Waterfall Model's rigid structure may make it challenging to accommodate such changes once the project is underway. Therefore, this approach would be less adaptable to the dynamic nature of the project.

### **Incremental Model:**

The Incremental Model involves developing and delivering the system in parts, with the final system becoming available only after the last increment. While this model offers some flexibility, it can lead to delays in delivering the complete system. The integration of various services like real-time booking, payment systems, and facility management may increase the complexity of the project. This model could potentially delay the full system's availability, which may not be ideal for a project that aims to provide a comprehensive solution in a timely manner.

### **Spiral Model:**

The Spiral Model is designed for large, high-risk projects that require frequent iterations, detailed risk assessments, and ongoing refinements. However, this model is less suited for our project, as it demands significant investment and highly skilled resources, making it a costly

---

approach for a project of this scale. Given the relatively smaller scope and budget of this project, the Spiral Model's complexity and capital-intensive nature are unnecessary and inefficient.

### **Rapid Application Development (RAD) Model:**

The RAD Model prioritizes quick development and user feedback through prototypes, enabling fast delivery of working software. While it may reduce development time, this approach requires substantial resources, including continuous availability of designers and developers. Due to the high budget and resource demands, and the inherent risks of investing too many resources in a short time frame, RAD is not ideal for this project. The risks involved outweigh the potential benefits of reduced time-to-market.

### **Conclusion**

The Agile Model has been selected as the development methodology for this project to ensure flexibility, regular feedback, and a focus on delivering value incrementally. This approach will allow the project team to efficiently adapt to changing requirements, manage risks, and keep the project aligned with stakeholder expectations throughout the development process.

## **11 Performance Requirements**

### **11a Speed and Latency Requirements**

Specifies the time available to complete tasks, focusing on response times and ensuring the software operates at an optimal speed for booking and managing sports facilities. These requirements are critical for ensuring a seamless user experience and smooth operations for facility managers and users.

For a sports facility booking app, speed is crucial for real-time booking and management. Users expect quick responses when searching for facilities, comparing prices, checking availability, and making bookings. Facility managers need fast updates to optimize bookings and respond to inquiries. Delays can lead to lost bookings, customer dissatisfaction, or double-bookings. Speed and efficiency ensure the system handles a high volume of concurrent users, especially during peak times. Example:

- 
- The system shall load available sports facilities within 2 seconds of the search query.
  - Booking confirmation must be completed within 3 seconds after payment.
  - The product shall display real-time availability of a facility within 1 second of selecting a date and time.
  - Notifications to users and facility managers about booking status shall be sent within 10 seconds of status change.
  - The product shall refresh facility status every 30 seconds during high-demand periods (e.g., weekends, holidays).

Most performance requirements are quantifiable for this app:

- The system shall respond to user queries in less than 2 seconds for 90% of the requests.
- No user query shall take longer than 3 seconds.
- Facility status updates must be reflected in real time with a maximum delay of 30 seconds.
- 95% of notifications must be delivered within 10 seconds.

Considerations for the product:

- During peak usage (e.g., large events, tournaments), the app must maintain its speed without degrading user experience.
- Prioritize speed for essential user actions such as booking and facility management updates over less time-critical actions like account settings changes.
- If the app connects with external systems (e.g., payment gateways), their speed and latency need to be considered, with appropriate fallbacks in place for slower systems.

## **11b Precision or Accuracy Requirements**

Specifies the necessary precision and accuracy for results produced by the sports facility booking software, ensuring reliable bookings and financial transactions.

Establishes user and stakeholder expectations regarding the precision of facility availability, payments, and user interactions with the software.

- All monetary transactions shall be accurate to two decimal places (₹0.01).

- 
- The accuracy of facility availability timings shall be within  $\pm 1$  minute to prevent booking conflicts.
  - Geo-location services for facility directions shall provide accuracy within  $\pm 10$  meters in urban areas and  $\pm 25$  meters in more rural areas.
  - The product shall maintain precise synchronization with IST (Indian Standard Time) to ensure accurate booking times and shift management.
  - All usage statistics and facility utilization reports shall be accurate within  $\pm 1\%$  to ensure the reliability of management decisions.

Consideration for the product:

- The system must ensure accurate time management to avoid booking conflicts, especially in high-demand facilities. Any shift or slot-based system must prevent overlapping bookings.
- The software must handle various modes of payment (including digital wallets, UPI, and net banking), requiring highly accurate transaction data to prevent errors during financial reconciliation.
- Currency handling must account for precision during conversion, especially if the software integrates international transactions.

## 11c Capacity Requirements

Specifies the volume of users, bookings, and data storage the sports facility booking software must handle to ensure smooth operation and scalability.

To ensure the software can process expected user volumes, facility bookings, and manage concurrent data requests during peak times, providing a seamless experience without performance degradation. Example:

- The product shall support up to 500 simultaneous users during peak hours (6:00 A.M. to 10:00 A.M. and 5:00 P.M. to 9:00 P.M.).
- During normal usage, the product shall support 300 concurrent users without significant latency or degradation in user experience.
- The system shall handle a maximum of 1000 bookings per hour during peak seasons.
- The product shall store a minimum of 2 years of facility usage data for reporting and analytics purposes, which includes user activity, bookings, payments, and feedback.

- 
- The software must handle up to 10,000 registered users and process up to 500 transactions per hour through various payment modes.

The fit criterion should be:

- The product's performance shall not degrade or exceed a response time of 2 seconds for 95% of user actions during peak hours.
- The system shall ensure that up to 500 concurrent users can book, cancel, or modify bookings without any system crashes or severe delays.
- Storage capacity shall accommodate facility data for up to 2 years without the need for additional manual data archiving procedures.

## 12 Dependability Requirements

### 12a Reliability Requirements

For the Sports Facility Booking Management Application, reliability is critical to ensure smooth, uninterrupted operation for both facility managers and users. The reliability requirements will be quantified based on the maximum allowable downtime and failure rates, especially considering real-time booking and payment processes. This ensures that users can trust the application for booking sports venues without delays or errors.

The app must be highly reliable to maintain user trust and meet operational expectations, especially since real-time bookings and payments are core functions. Ensuring reliability will prevent negative user experiences and optimize the performance of sports facilities. Setting appropriate expectations around failure and recovery also helps manage user expectations and contributes to better customer satisfaction. Reliability Requirements are:

- The application shall have a maximum failure rate of once per month.
- No booking, payment, or user data shall be lost or corrupted in the event of a failure.
- The system should ensure that data is backed up and recoverable in case of system downtime or crashes.
- The application must automatically recover from a failure, with users experiencing minimal service disruption.
- Users will be notified of any downtime or failure, along with an estimated recovery time, and critical functions like ongoing bookings or payments will be prioritized for restoration.



- The focus will be on ensuring high availability, where the system is continuously operational and service interruptions are minimal.
- Assess the cost implications of achieving high reliability (99.9% uptime) and how that aligns with the expectations of users and facility managers.
- Redundant systems and backups should be in place, especially for crucial modules like booking and payment, to mitigate the impact of system failures.

## 12b Availability Requirements

This section ensures that the Sports Facility Booking Management Application operates reliably, supporting uninterrupted access for both facility managers and users, while also specifying fail-safe measures for unforeseen events.

The availability of the Sports Facility Booking Management Application is crucial for delivering a seamless user experience. Since users will rely on the system to book and manage sports facilities in real-time, it must be highly available with minimal downtime. Availability is calculated as a fraction of total time the system is operational, considering the mean time between failures (MTBF), mean time to recovery (MTTR), and scheduled maintenance periods.

The application's availability is key to user satisfaction and operational efficiency, especially for facility managers and sports enthusiasts who require access to bookings at any time. Availability ensures that even if the system encounters failures, downtime is minimized, and users can depend on the application being accessible when needed.

- The application shall achieve 99.9% availability.
- The system shall be available for use 24 hours per day, 365 days per year, except during scheduled maintenance.
- In case of a failure, the system must be restored within a maximum downtime of 30 minutes.
- Routine maintenance will be scheduled during off-peak hours, preferably from 12:00 AM to 4:00 AM, to reduce the impact on users.
- Users must be notified at least 48 hours in advance of any scheduled maintenance or planned downtime.
- The need for 99.9% availability requires thorough planning of redundancy measures and ensuring that failures do not lead to prolonged downtime.

- Regular maintenance will be performed during times of low usage to limit disruptions to users.
- Achieving a balance between high availability and cost efficiency is important, as higher availability often incurs additional costs for infrastructure and support.

## 12c Robustness or Fault Tolerance Requirements

Robustness refers to the ability of the Sports Facility Booking Management Application to maintain functionality and provide core services in the presence of abnormal circumstances, such as system failures, network issues, or unexpected user behaviour.

The goal is to ensure that the application can continue to operate effectively during abnormal events or partial system failures, providing users with uninterrupted access to essential features like viewing, booking, and managing facilities. This is crucial for maintaining user trust and ensuring smooth operations for both facility managers and users.

- The application shall continue to allow users to browse facility details and manage offline bookings in the event of a network outage. Data will be synchronized when the connection is restored.
- The system shall support emergency operations for up to 10 minutes if the connection to the central server is lost, allowing users to complete ongoing bookings or payments.
- If the application loses connection to external payment gateways, users shall be notified, and the system shall queue the transactions for processing once connectivity is restored.

The application must be designed to handle various types of disruptions, including:

- Network Failures: In case of internet connectivity loss, essential features like booking and viewing schedules should be available in offline mode. Once the network is restored, all changes should be synced to the central server.
- Server Failures: The application shall implement redundancy and fallback mechanisms, ensuring that local functionalities continue to work. A disaster recovery plan shall also be in place to ensure rapid restoration of services and minimize downtime.
- Power Outages: Emergency measures must be integrated to keep critical functionalities operational during brief outages, ensuring that users can complete ongoing transactions.

---

In case of major system failures or disasters, the application shall have a disaster recovery plan to re-establish normal performance as quickly as possible. This includes:

- **Regular Backups:** Data backup routines shall be implemented to ensure that no booking or payment information is lost during failures.
- **Failover Mechanisms:** Redundant servers or backup systems shall be in place to take over if the primary system becomes unavailable.

## **13 Maintainability and Support Requirements**

### **13a Maintenance Requirements**

This section defines the time and effort required to make specific modifications or updates to the Sports Facility Booking System. The system's design must support maintainability to ensure that it can be updated and extended with minimal downtime and resource expenditure.

To ensure that all stakeholders are informed of the product's maintenance needs, ensuring the smooth operation, timely updates, and long-term sustainability of the system.

- **Response Time for System Modifications:** All requested updates or enhancements to the system, once approved, must be implemented and deployed within a specific timeframe depending on the nature of the change. For example:
  - **Critical fixes** (e.g., bugs affecting the booking process or payment system) must be resolved within 24 hours of detection.
  - **Feature updates** (e.g., adding a new sport or modifying booking workflows) must be delivered within one week from when the requirement is finalized.
  - **Cosmetic changes** (e.g., UI/UX improvements or content updates) must be implemented within two business days.
- **System Scalability:** The system must allow for the addition of new sports facilities or sports types without disrupting existing functionality. New sports facilities should be available for booking within 12 hours of being entered into the system.
- **Maintainability by Non-Original Developers:** The system must be developed in a way that allows developers who were not part of the original team to easily maintain and modify it. This includes well-structured code, clear comments, and adherence to standard programming practices.

- **End-User Maintenance:** Facility managers, who are not technical experts, must be able to manage basic system functions such as adding new sports facilities, modifying availability, adjusting pricing, and generating reports without needing external technical assistance.

Example:

- **New Facility Addition:** If a new sports facility (e.g., a cricket pitch or tennis court) is added to the platform, it must be available for booking by users within 12 hours of setup.
- **New Reporting Features:** Any requested MIS reports (e.g., a detailed report on facility usage or revenue generation) must be made available within one week of confirming requirements.

Consideration for the product will be:

- **Maintainability Documentation:** Comprehensive documentation must be provided to ensure that any developer or system administrator can easily navigate and update the system. This documentation should include the system's architecture, workflows, and maintenance procedures.
- **Training and Support:** Proper training materials and user manuals must be provided to facility managers and administrators to ensure they can maintain and operate the system effectively without needing external assistance for routine tasks.
- **Testability:** Every update to the system must be thoroughly tested in a staging environment before being deployed to production. Automated testing should be implemented for critical workflows such as bookings, payments, and user authentication. Additionally, regression testing must be conducted after each update to prevent unintended side effects on other system functions.

## 13b Supportability Requirements

This section outlines the level of support the Sports Facility Booking System needs to ensure a smooth experience for users, including athletes, sports enthusiasts, and facility managers. It specifies how support will be provided, either through a help desk or built directly into the product itself.

By clearly defining the support requirements, we can ensure users get the help they need when they need it, minimizing downtime and frustration while maximizing the system's usability.

---

- **Help Desk Availability:**

A dedicated help desk will be available to handle user questions and issues. This team will operate between 8 AM and 8 PM to cover the busiest hours.

- **Advantages:** Provides real-time assistance during peak times, helping users feel supported.
- **Consideration:** Limited hours may not accommodate users outside of these times, especially in different time zones.

- **Quick Responses:**

Critical issues, such as problems with bookings or payments, will receive a response within 1 hour and be resolved within 24 hours. Less urgent issues will be addressed within 24 hours and resolved in 3 business days.

- **Advantages:** Clear timeframes for resolution help manage user expectations.
- **Consideration:** During high-demand periods, there might be challenges maintaining these response times.

- **Self-Help Tools:**

Users will have access to an in-app help section with FAQs, tutorials, and guides for resolving common issues on their own. Additionally, interactive guides will walk them through tasks like making or managing bookings.

- **Advantages:** Empowers users to solve problems quickly without waiting for help, reducing the burden on the help desk.
- **Consideration:** Some users may still prefer personal assistance, particularly for more complex issues.

- **Clear Error Messages:**

If something goes wrong (like a failed booking or payment), the system will display a clear message explaining the issue and offering suggestions for fixing it, along with links to support resources.

- **Advantages:** Helps users understand what's happening and reduces confusion.
- **Consideration:** If the error message isn't specific enough, users might still need to contact support.

- Knowledge Base:

A searchable online resource will offer detailed articles, video tutorials, and how-to guides for both users and facility managers. This will help users find answers independently.

- Advantages: Available 24/7, users can access it anytime, providing an instant support option.
- Consideration: The knowledge base must be regularly updated to stay relevant, which requires ongoing effort.

- Support for Facility Managers:

Facility managers will have access to a dedicated support portal with advanced troubleshooting resources and a ticketing system for reporting technical issues related to bookings or facility management. Live chat support will be available during peak hours for quicker responses.

- Advantages: Tailored support for managers helps ensure the smooth operation of their facilities, directly impacting revenue.
- Consideration: Live chat may require additional staffing to handle peak demand efficiently.

- Feedback Mechanism:

The platform will feature an integrated feedback tool for users to report bugs, suggest improvements, and rate their experience.

- Advantages: Collecting user feedback regularly can lead to continuous improvement of the system.
- Consideration: Without timely responses or visible changes, users may feel their feedback isn't valued.

Additional Consideration required would be:

- No Printed Manuals:

All support will be provided digitally via the help desk, in-app tools, and the knowledge base. Printed manuals will not be offered, streamlining the process and keeping everything online.

- Advantages: Reduces printing costs and ensures information remains up to date.
- Consideration: Some users may still prefer printed materials, particularly if they struggle with digital tools.

- **Self-Support Focus:**

The system will emphasize self-support features to help users resolve issues on their own, reducing dependency on the help desk.

- **Advantages:** Quicker resolutions and less strain on support staff.
- **Consideration:** Some users may find self-help tools insufficient, particularly for more complicated problems.

- **Multilingual Support:**

To accommodate a diverse user base, the help desk and knowledge base will support multiple languages, including English, Spanish, and regional languages based on the user base.

- **Advantages:** Increases accessibility and inclusivity.
- **Consideration:** Translating and maintaining resources in multiple languages may require significant investment.

- **Proactive Notifications:**

The system will notify users in advance about any scheduled maintenance or system downtimes via in-app notifications and emails/SMS.

- **Advantages:** Keeps users informed and reduces frustration caused by unexpected downtime.
- **Consideration:** If notifications are not timely or clear, users may still feel inconvenienced.

### 13c Adaptability Requirements

This section specifies the platforms where the Sports Facility Booking System will operate, with a focus on mobile devices. While the application is designed for mobile use initially, there are plans for future expansion to other devices like smart watches.

To clarify the current and potential future platforms the product will support, ensuring that users and stakeholders have a clear understanding of where the application can be accessed.

Requirements for the adaptability will be:

- **Mobile Platforms:**

The application will be available on:

- **Android Devices:** Compatible with smartphones running Android 9.0 (Pie) or later.

- **Future Expansion to Smart watches:**

Although not available at launch, future versions of the app may be extended to support smart watches. This could include basic features such as notifications, booking reminders, and facility availability updates.

Fit Critereon for the product:

- **Mobile Compatibility:** The product will be fully operational on mobile devices with the specified versions of Android.
- **Smart watch Support in the Future:** The app should be adaptable for integration with smart watches within 12 months of the decision to move forward with this feature.

Consideration for the product:

- **Mobile-First Approach:** At present, the system will prioritize mobile platforms, with no desktop or web versions planned. All user interactions, including booking, notifications, and facility management, will be handled through the mobile app.
- **Regional Considerations:** The app is optimized for users in certain regions, taking into account local currencies, languages, and regulations, with potential plans for expansion depending on market demand.

By focusing on mobile platforms for now and planning future expansions, the app remains flexible and poised for growth as user needs evolve.

### **13d Scalability or Extensibility Requirements**

This section outlines the scalability and extensibility expectations for the Sports Facility Booking System, ensuring the product can handle increased user and booking volumes as the user base grows.

To ensure that the system is designed with future growth in mind, allowing it to scale smoothly as the number of users, facilities, and bookings increases over time.

- **User Growth:**

The system is currently expected to support up to 50,000 active users at launch. It should be scalable to accommodate 200,000 active users within three years. The platform must be designed to allow for user growth without compromising performance or user experience.

- **Booking Volume:**

At launch, the system must be capable of handling 5,000 bookings per day. Within two years, the system should be able to process 25,000 bookings per day, ensuring



that spikes in activity (e.g., during tournaments or seasonal sports) can be managed efficiently.

- **Facility Manager Growth:**

The system is expected to support 500 facility managers at launch. This number is projected to grow to 2,000 facility managers within the next three years. The platform must be able to handle this increase in managed facilities and optimize scheduling and resource allocation without performance degradation.

- **Transaction Handling:**

The system should be capable of processing 2,000 transactions per hour at launch, with the ability to scale up to 10,000 transactions per hour within two years.

- **Data and Reporting:**

As more users and facilities are added, the system must efficiently manage and store user and booking data, ensuring that facility managers can continue generating accurate reports on bookings, revenue, and usage patterns even as the database grows.

The fit criteria will be:

- **User and Booking Volume:** The product must handle up to 200,000 users and 25,000 daily bookings within three years without performance degradation.
- **Transaction Capacity:** The system should scale to handle up to 10,000 transactions per hour by the second year after launch.

By designing with scalability in mind, the Sports Facility Booking System will be equipped to handle future growth while maintaining high performance and user satisfaction.

## **13e Longevity Requirements**

The Sports Facility Booking System is expected to provide long-term value and operate efficiently within a defined maintenance budget.

To ensure the product delivers a sustainable return on investment and remains functional over time without excessive upkeep.

- **Minimum Lifetime:**

The system must function effectively for at least five years with regular maintenance, including updates and minor improvements.

- **Maintenance Budget:**

---

Annual maintenance costs, covering security updates and optimizations, must not exceed 10% of the initial development cost.

- **Adaptability:**

The system must accommodate updates to mobile platforms and technologies over the five-year lifespan without requiring a major overhaul.

- **Operational Longevity:** The product must run smoothly for five years with manageable maintenance.
- **Budget Compliance:** Maintenance expenses must remain within 10% of initial costs annually.
- **Technology Changes:** The system should adapt to evolving mobile OS updates over time.
- **Future Growth:** Performance must scale with user growth without compromising stability or increasing maintenance costs.

By ensuring longevity and cost efficiency, the system will offer sustainable value for users and stakeholders.

## **14 Legal Requirements**

### **14a Compliance Requirements**

To comply with the law in order to prevent delays, lawsuits, and legal fees during and after development. Like:

- Personal information collected for user registrations and bookings shall comply with the Data Protection Act and GDPR.
- The system shall adhere to online payment regulations, including PCI DSS compliance for secure handling of payment information.
- Lawyers' opinion that the sports booking system complies with all applicable legal regulations and does not infringe on any laws.
- Consult with legal professionals to identify relevant legal requirements for the sports booking system.
- Are there any copyrights, trademarks, or other intellectual property that must be protected?

- 
- Does the system incorporate any proprietary algorithms or methods that need to be protected?
  - Are there regulations specific to the sports industry that need to be adhered to?
  - If the system provides services across different countries, ensure compliance with international sports booking regulations.
  - The Unlawful Internet Gambling Enforcement Act (UIGEA), the Gambling Act 2005 (UK), and other local gambling laws may apply. Consult legal experts.
  - Are there any restrictions on advertising sports booking services in certain regions or media?
  - Consider the implications of the Americans with Disabilities Act (ADA) for providing accessible web services.
  - Are there any tax laws specific to sports-related revenues, such as event ticketing or betting?
  - Ensure compliance with local and international labour laws if offering event staffing or employment services.

## 14b Standards Requirements

A statement specifying applicable standards and referencing detailed standards descriptions relevant to the sports booking system.

To comply with industry standards to ensure high quality, reliability, and security, and to avoid delays due to non-compliance.

- The system shall comply with the ISO/IEC 27001 standard for information security management.
- The user interface and experience shall adhere to the Web Content Accessibility Guidelines (WCAG) to ensure accessibility for all users.
- The software development process shall follow the Agile methodology as specified by the Agile Manifesto and its principles.
- The system shall comply with PCI DSS standards for secure payment processing.
- Data exchange formats shall comply with the ISO 20022 standard for financial data transmission.

The appropriate standard-keeping body certifies that the sports booking system has adhered to the specified standards.

- 
- Identify any industry-specific standards, such as those set by sports associations, booking agencies, or technology bodies.
  - Check if there is a code of practice for online sports booking services, or any standards set by watchdog organizations.
  - Consider if there are any best practices or standards provided by industry associations, such as the World Lottery Association (WLA) or the International Sports Betting Integrity Association (IBIA).
  - Ensure adherence to standards for digital transactions, such as the EMV standard for payment cards if applicable.
  - Verify if any national or regional standards are applicable, such as those set by the National Institute of Standards and Technology (NIST) for cyber security.
  - Determine if there are special development methodologies recommended for high-traffic or real-time booking systems, such as Test-Driven Development (TDD) or Behaviour-Driven Development (BDD).
  - Investigate whether there are any standards for integrating with third-party APIs, such as OAuth for authentication.
  - Ensure that data privacy and protection standards, such as GDPR, are incorporated into the development process to protect user data.

---

## III Design

### 15 System Designs

#### 15a Design goals

Design goals define the critical properties of the Sports Facility Booking Management System that need to be optimized to enhance the user experience and ensure efficient functionality. These goals provide direction for the design and development of the system, focusing on achieving optimal performance, scalability, security, and user satisfaction. Unlike strict requirements, design goals do not specify exact values but represent desired characteristics that the system should strive to maximize.

The following are the primary design goals for our Sports Facility Booking Management System:

1. Speed and Performance

- The system must respond swiftly to user actions, such as searching for facilities, applying filters, and completing bookings. Since users may make bookings at the last minute or under time constraints, minimizing delays is critical.
- The booking and payment processes, in particular, should be optimized for speed. Although the system has specific response-time requirements, the goal is to enhance performance as much as technically possible, ensuring a seamless and responsive experience across all functions.

2. Scalability

- The system should be designed to handle an increasing number of users and bookings without compromising performance. As the user base and the number of facilities in the system grow, the system architecture must scale smoothly to accommodate the additional load.
- This includes ensuring that the system can efficiently manage peak-time usage, such as weekends and holidays, when booking traffic is higher. Scalability will support business growth by accommodating new users, facilities, and additional services.

### 3. Reliability and Availability

- Ensuring that the system is reliable and available 24/7 is essential, as users may access the application at any time to make bookings. Any unexpected downtimes or system failures would negatively impact user trust and satisfaction.
- While there are specific uptime requirements outlined, the design goal is to maximize reliability and availability, implementing redundancy and failover mechanisms to handle potential disruptions.

### 4. User-Friendly Interface

- The system interface should prioritize ease of use, making it simple for users to navigate, search for facilities, apply filters, select time slots, and finalize bookings. The booking process should be intuitive and user-friendly, reducing the effort needed to accomplish tasks.
- This goal emphasizes a streamlined design across both mobile and web applications to ensure accessibility for all users, regardless of their level of technical expertise.

### 5. Security and Data Privacy

- Security is a critical design goal, given the need to protect user data and payment information. The system must use robust encryption and secure payment gateways to ensure data privacy and prevent unauthorized access.
- While the system must comply with industry security standards, the goal is to exceed these requirements where possible, creating a secure environment that fosters user confidence in data protection and privacy.

### 6. Modularity and Maintainability

- The system should be designed with modularity in mind, allowing individual components to be updated or replaced without affecting other areas of the system. This modular approach simplifies maintenance and allows for future enhancements.
- For instance, if an update is made to the payment module or booking logic, it should not disrupt other functionalities. The design goal is to ensure that the system is maintainable and adaptable to evolving business needs.

## 7. Integration with External Systems

- The system should be flexible enough to integrate with external systems, such as third-party payment gateways or facility management software. Smooth integration enables data flow and compatibility with other platforms that users or facility managers may use.
- Although specific API requirements are outlined elsewhere, the goal is to make the system as flexible as possible to support current and future integrations without extensive redesign.

## 8. Data Accuracy and Consistency

- Data accuracy and consistency, particularly in terms of booking availability and payment statuses, are essential for a reliable user experience. Any inaccuracies could lead to issues such as double bookings or resource mismanagement, affecting both users and facility managers.
- While data verification requirements are specified, the goal is to further enhance accuracy by incorporating validation checks within the booking, availability, and payment modules.

# 16 Current Software Architecture

The architecture of the Sports Facility Booking Management System is designed specifically for a mobile application, focusing on delivering a high-performance, scalable, and secure booking experience. This architecture ensures that users can search for, filter, and book sports facilities seamlessly. The system is optimized for mobile app performance, balancing ease of use, reliability, and secure operations to create a smooth booking process for both end users and facility managers.

Key considerations for this application architecture include:

- **Mobile Performance Optimization:** Since the application is designed solely for mobile use, the architecture is optimized to deliver fast response times and smooth interactions, particularly for booking and payment tasks.
- **Scalability and Modular Design:** As demand grows, especially during peak usage times, the system must scale effectively. By structuring the app around distinct modules—such as Booking, Payment, and Facility Management—it's easier to maintain and update the system, without interrupting user experience.

- **Security and Compliance:** To protect user data and payment information, the architecture includes robust security features, such as data encryption and secure authentication. These are essential to ensure compliance with data protection regulations.
- **High Availability and Reliability:** Mobile app users expect the application to work whenever they need it. Therefore, the architecture incorporates mechanisms to ensure uptime and reliability, even if parts of the system encounter issues.
- **Integration with External Systems:** The app integrates with third-party services like payment gateways and facility management systems, ensuring a smooth experience from booking through payment.
- **User-Centric Mobile Experience:** The architecture is designed to prioritize an intuitive, responsive mobile user experience, making it simple for users to book facilities on the go.

Here's a breakdown of the application's architecture:

1. **Presentation Layer (Mobile Application Interface):**

- This layer includes the mobile app's user interface, where users can search for facilities, view details, select timeslots, and make bookings.
- The interface is optimized for mobile navigation and responsiveness, providing a smooth, user-friendly experience.
- It communicates with backend services through REST APIs designed specifically for mobile performance, ensuring efficient data transfer and minimizing load times.

2. **Application Layer:**

This layer houses the core mobile services, structured as micro services for scalability and ease of maintenance:

- **Booking Service:** Handles all booking operations, including checking facility availability and confirming bookings.
- **Facility Management Service:** Maintains up-to-date information on facilities and their availability, pulling in updates from facility managers.
- **Payment Service:** Processes payments, interacting securely with third-party payment gateways.



- **User Management Service:** Manages user authentication, registration, and profile details.

A micro services approach allows each component to operate independently, ensuring that changes to one part do not affect the entire app's functionality.

3. **Data Layer:**

- A relational database stores essential data like user profiles, booking history, facility details, and payment records.
- Caching mechanisms (such as Redis) store frequently accessed data, like facility availability, to improve app performance and reduce load times for users.

4. **Integration Layer:**

- This layer manages the communication between the app and external services, such as payment gateways.
- Asynchronous processing ensures that any delays in communication with external systems do not impact the user experience within the app.

5. **Security Layer:**

- Security features, including data encryption, token-based authentication, and secure access controls, protect user data and payment details.
- SSL/TLS encryption secures data transfer between the mobile app and the backend services, maintaining data privacy and user trust.

6. **Monitoring and Logging:**

- Real-time monitoring tools track app performance, alerting the development team to potential issues before they impact users.
- Logging tools capture system activities and user interactions, aiding in troubleshooting and system improvements.

---

## 17 Proposed Software Architecture

### 17a Overview

For the Sports Facility Booking Management System, the most appropriate architecture style is a Layered Architecture. This design style offers a structured approach to organizing components, making it easier to manage, test, and scale. The layered structure separates responsibilities into distinct layers, allowing changes in one layer without impacting others, which is beneficial for a mobile application focused on real-time booking, payment processing, and user management.

The motivation for choosing a layered architecture is to create a well-organized and maintainable application that supports scalability, security, and efficient data flow. Since the application involves user interactions, real-time updates, secure payment processing, and facility management, a layered approach ensures these functionalities are isolated in separate, manageable layers. This architecture style aligns well with our mobile-only approach, enabling efficient data access and easy updates as the system evolves.

- **Scalability:** The layered structure allows each layer to be scaled independently as needed, optimizing resource usage without affecting the entire application.
- **Modularity and Maintainability:** Layered architecture divides the application into self-contained layers (e.g., presentation, application logic, and data), making it easier to maintain and update components individually.
- **Separation of Concerns:** Each layer has distinct responsibilities, ensuring that business logic, data access, and user interface concerns remain isolated, improving code quality and reducing errors.
- **Security:** Sensitive data handling and access control are centralized in the relevant layers (e.g., data and service layers), simplifying the implementation of security protocols and regulatory compliance.
- **Reliability:** The separation between layers allows the system to handle failures more gracefully. For instance, issues in the presentation layer do not affect data handling or business logic directly.

- **Flexibility and Extensibility:** Adding new features or integrating third-party services (e.g., additional payment gateways) is easier with this architecture, as each layer can be updated independently.

The proposed layers for the architecture will be:

1. **Presentation Layer (UI/UX):**

- The front-end mobile application where users interact with the system to search, view, and book sports facilities.
- Provides a responsive, user-friendly interface optimized for mobile devices, delivering a seamless booking experience.
- Interfaces with the business layer through RESTful APIs to access services like booking, facility details, and payments.

2. **Business Logic Layer:**

- Contains core business logic for handling bookings, payments, user management, and facility operations.
- Acts as the intermediary between the presentation layer and the data layer, ensuring that business rules are applied consistently.
- Responsible for tasks such as checking facility availability, confirming bookings, managing user accounts, and processing payments.

3. **Service Layer (Middleware):**

- Manages communication between internal layers and external services (e.g., payment gateways and notification services).
- Provides APIs for the business layer to interact with third-party services asynchronously, improving response times and handling failures gracefully.
- Helps with tasks such as validating payment information and handling booking confirmations from facility managers.

4. **Data Access Layer:**

- Handles all interactions with the database, including data retrieval, updates, and storage.
- Isolates data-related operations to ensure that data handling is consistent and optimized, supporting quick data retrieval for a smooth user experience.
- Caches frequently accessed data (e.g., facility availability) to improve performance and reduce database load.

---

### 5. Database Layer:

- Stores persistent data, including user profiles, booking history, facility details, and transaction records.
- Hosted on a cloud-based database, allowing scalability, security, and reliability for data storage.
- Supports real-time updates and ensures data integrity across transactions, particularly for booking and payment information.

This layered architecture provides a robust, scalable, and secure foundation for the Sports Facility Booking Management System. By separating functionalities into layers, the application becomes easier to manage, secure, and extend, meeting the needs of mobile users seeking a smooth, real-time booking experience. This architecture is ideal for supporting future growth, as each layer can be scaled or modified independently, ensuring adaptability as the system and user base expand.

## 17b Class Diagrams

The class diagram for the Sports Facility Booking Management System outlines the static structure of the application, showcasing the key classes, their attributes, methods, and relationships. This diagram provides a comprehensive view of the system's main components, including User, Booking, Facility, Payment, and Notification, each with its own specific functionality. The class diagram serves as a foundational guide for building the application's core functionalities, ensuring a cohesive design that efficiently supports the booking and management of sports facilities.

The purpose of this class diagram is to define the structure and interactions between the system's main classes, helping to streamline development and facilitate future updates. By clearly outlining the responsibilities and relationships of each class, this model promotes a modular and maintainable system structure. It allows for organized implementation, where each class encapsulates specific functions related to user actions, facility management, booking processing, and payment handling. This approach ensures that the system remains flexible, scalable, and easy to extend with additional features or services.

In designing this class diagram, several key considerations were taken into account to support the application's functionality, maintainability, and growth:

- 
- **Encapsulation:** Each class has defined attributes and methods that encapsulate its responsibilities, preventing external interference and enforcing data integrity. For example, User handles user-related operations, while Booking focuses on the booking management process.
  - **Relationships:** Clear relationships between classes are defined, including associations, composition, and inheritance. For instance, the Notification class has subclasses EmailNotification and SMSNotification to represent different types of notifications. The User and Booking classes have a one-to-many association, indicating that a user can have multiple bookings.
  - **Modularity:** Classes are organized by distinct functionalities, allowing the system to remain modular. This modularity simplifies debugging and testing, as each class operates independently, focusing on its own tasks within the system.
  - **Reusability:** Core components, such as Payment and Notification, are structured to be reusable across different areas of the system. This enables future features or integrations with minimal restructuring, as reusable components can be adapted to various contexts within the application.
  - **Scalability:** The diagram is designed with scalability in mind, allowing for easy addition of new features or modification of existing ones. For example, more subclasses could be added to Notification if additional notification methods are needed, or more attributes could be added to Facility if additional facility details are required.

This class diagram establishes a well-defined structure for implementing the sports facility booking management system, supporting efficient development, ease of maintenance, and future expansion.

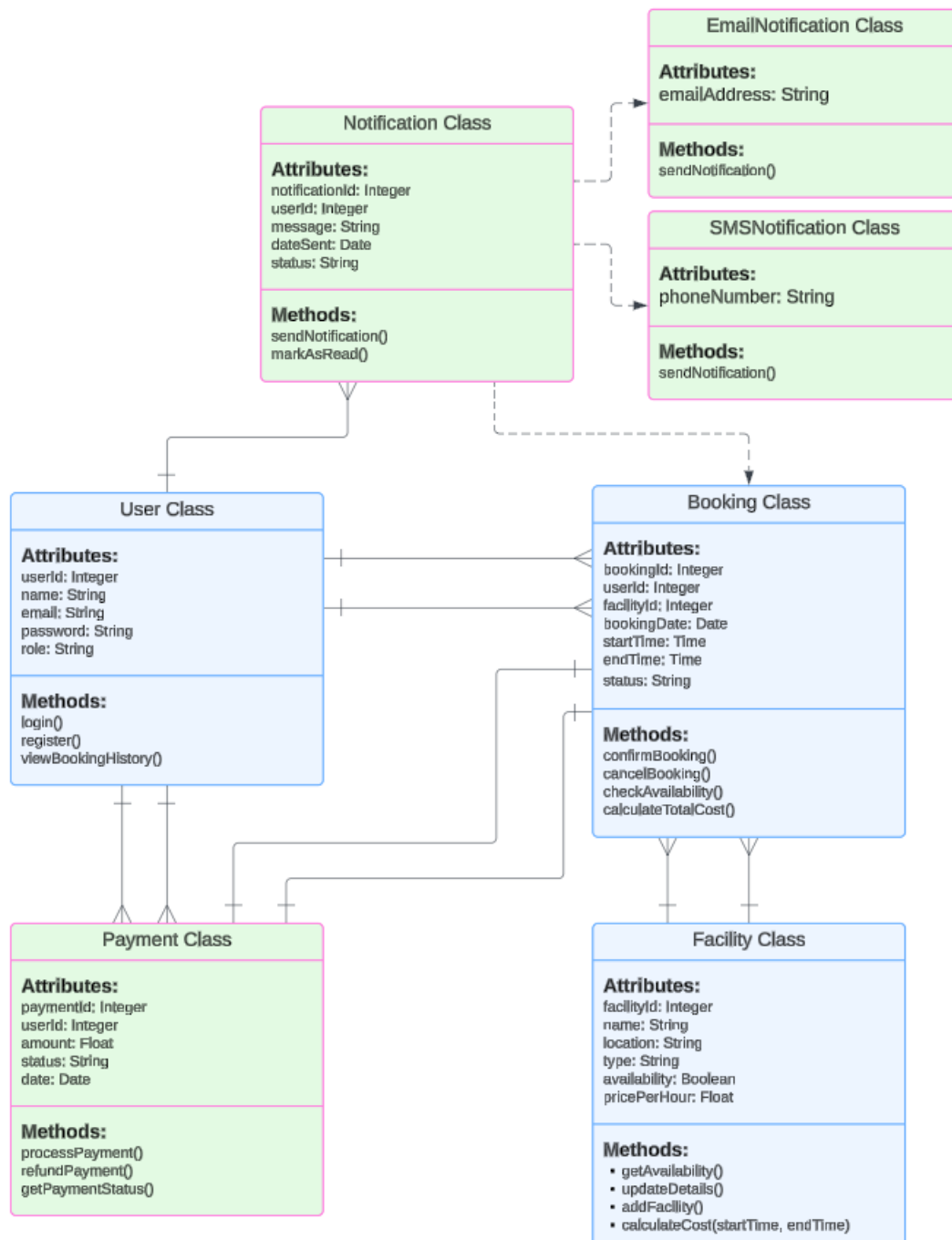


Figure 2: Class Diagram

---

## 17c Dynamic Model

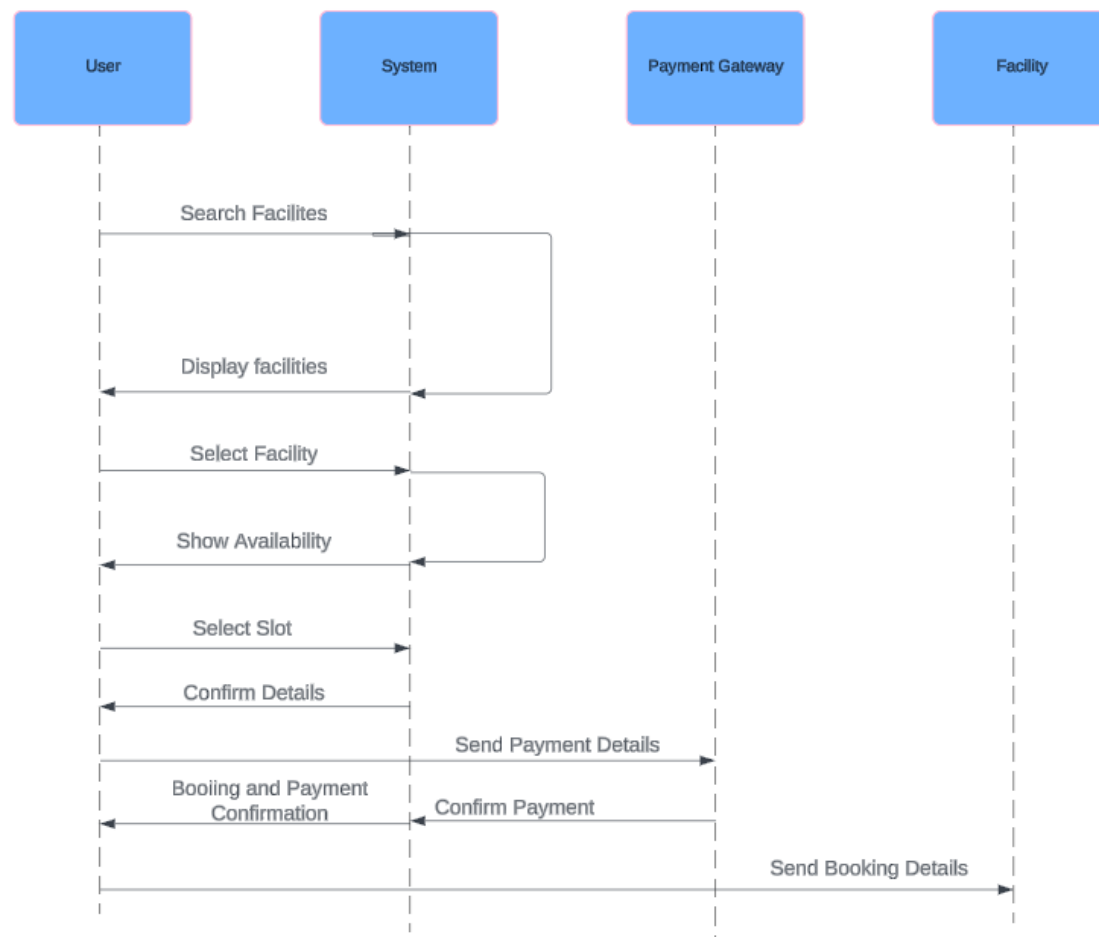
The dynamic model for the Sports Facility Booking Management System illustrates the system's behaviour over time, showing interactions among objects as they respond to user actions, such as searching for a facility, booking a slot, or processing payments. This model is represented with sequence diagrams, state diagrams, and activity diagrams, providing insight into the system's real-time functionality.

The dynamic model helps to visualize how various components in the system interact, focusing on the sequence of events, conditions, and dependencies among entities. This is critical for validating the flow of operations, optimizing response times, and ensuring seamless user interactions. It also enables us to spot and address potential bottlenecks in user-facing processes like booking and payments, where speed and accuracy are essential. When creating the dynamic model, the following factors were prioritized:

- **User Interaction Flow:** Mapping out the complete sequence from initial search to booking confirmation, including edge cases for cancelled or failed bookings.
- **Error Handling and Recovery:** Ensuring that the system handles issues such as failed payments, unavailable facilities, or incomplete bookings, and that users receive appropriate feedback.
- **Concurrency:** Managing concurrent access to facilities, especially in high-demand scenarios, ensuring data consistency and preventing overbooking.
- **Responsiveness:** Ensuring that user interactions are met with real-time responses, particularly in processes like payment confirmation or availability checks.
- **Security and Validation:** Maintaining secure interactions in processes, especially for sensitive actions like login, payment processing, and data retrieval.

The dynamic model of the sports facility booking system includes a detailed sequence and data flow diagrams (DFDs) at various levels, illustrating the end-to-end process of booking a facility. This model highlights the interactions between primary components, including the User, Sport Facility Booking System, Facility Owner/Manager, and Payment Gateway, showcasing the system's workflow and real-time data exchange. These diagrams provide a structured view of the user journey and the critical system interactions involved in searching, selecting, and booking a facility, as well as confirming payment.

This sequence diagram outlines the interaction between the User, the System, the Payment Gateway, and the Facility during the process of booking a sports facility. The diagram shows the chronological flow of events, starting from searching for a facility and ending with sending the booking details to the facility after successful payment confirmation. It highlights the key steps involved, including selecting a facility, checking availability, making a payment, and confirming the booking.

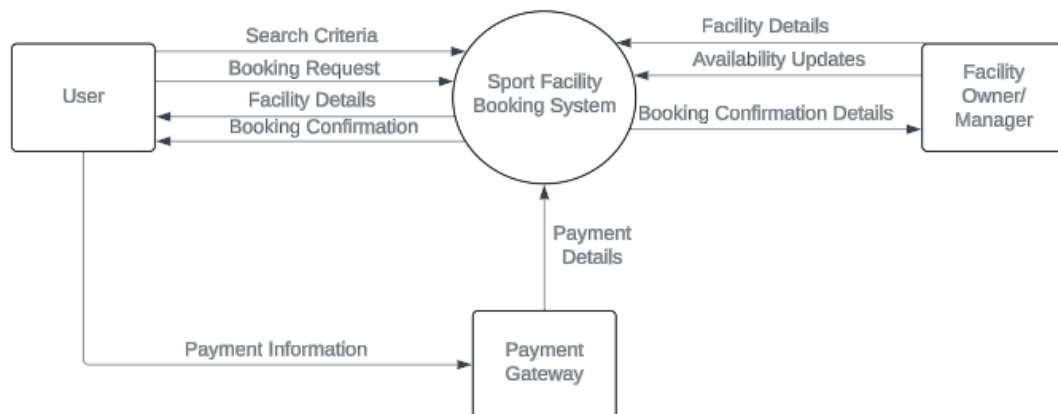


**Figure 3: Sequence Diagram**

The Level 0 DFD represents a high-level overview of the sports facility booking system. It identifies the primary components—the User, Sport Facility Booking System, Facility Owner/Manager, and Payment Gateway—and their interactions. Users initiate searches for available facilities and send booking requests to the system, which processes bookings and confirms availability with the Facility Owner/Manager. The Payment Gateway handles secure payment transactions, ensuring that bookings are completed efficiently. This

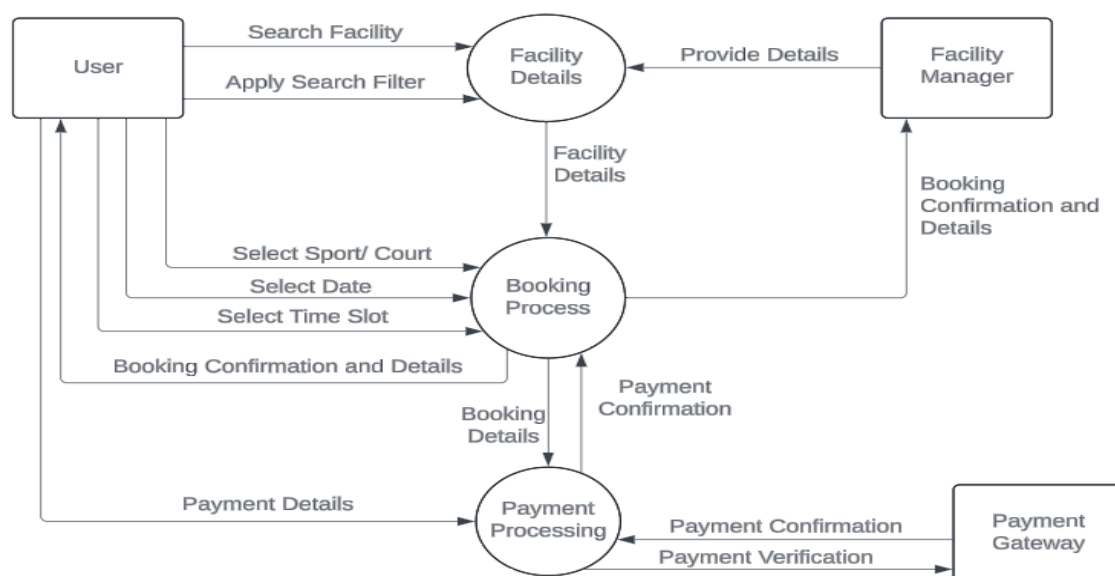


diagram shows the primary flows of information, giving a broad view of the system's external interactions and high-level functionality.



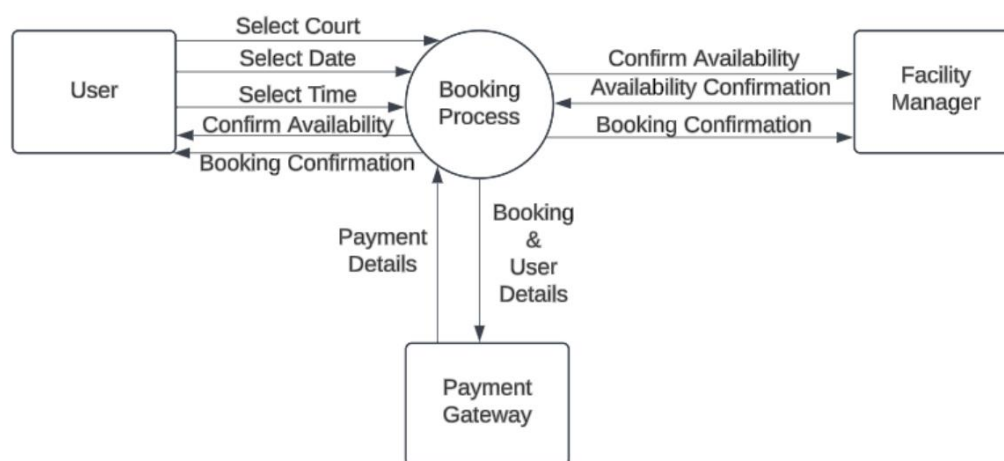
**Figure 4: Data Flow Diagram Level 0**

In Level 1, the system's core functionalities are expanded to show more specific interactions, breaking down each step in the booking process. Users can search and filter facilities, view facility details, and select specific booking criteria such as date and time. The system then processes this information, verifies booking availability with the Facility Manager, and initiates payment processing. The Payment Processing component handles transactions through the Payment Gateway, confirming successful payments back to the system. This level reveals additional detail on how each user action is processed and how subsystems work together to complete a booking.



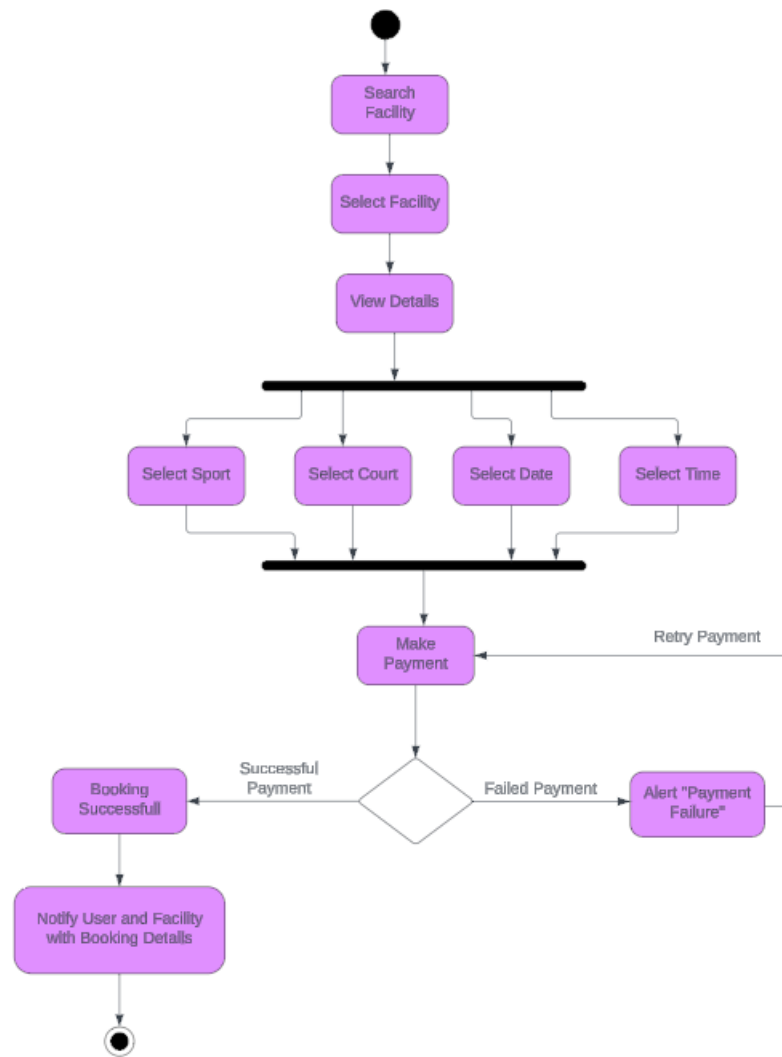
**Figure 5: Data Flow Diagram Level 1**

The Level 2 DFD offers a detailed view of the booking process, highlighting each specific interaction needed to complete a booking transaction. Users specify their booking preferences, which the system verifies by communicating with the Facility Manager. Once availability is confirmed, the system sends booking and user details to the Payment Gateway, ensuring secure payment. Confirmation messages are sent to both the User and Facility Manager upon a successful transaction. This detailed flow reduces ambiguity in the booking process, emphasizing real-time availability checks and secure payment integration to deliver a seamless booking experience.



**Figure 6: Data Flow Diagram Level 2**

This diagram illustrates the process flow for booking a sports facility, starting from searching for a facility to completing the payment. The diagram highlights key decision points, parallel actions (such as selecting the court, sport, date, and time), and the outcomes based on the payment status.

**Figure 7: Activity Diagram**

The sequence and data flow diagrams in this dynamic model offer a comprehensive breakdown of the booking process. By representing interactions at multiple levels, this model captures the complexity and precision required for an optimal user experience. It not only facilitates smooth booking transactions but also ensures secure, real-time communication between the User, Facility Manager, and Payment Gateway, laying the groundwork for a reliable and adaptable booking system.

---

## 17d Subsystem Decomposition

Subsystem decomposition outlines the distinct areas of responsibility within the Sports Facility Booking Management System, organizing the system into focused subsystems. This decomposition is crucial for establishing clear boundaries for development and maintenance, enabling different teams to work on specific parts of the system without overlap.

Subsystem decomposition enables a well-organized, maintainable, and scalable system by breaking down complex functionality into smaller parts. This structure promotes modular development, where each subsystem focuses on a particular responsibility, such as managing user data or processing payments. Subsystem decomposition also enhances flexibility, allowing subsystems to be updated, replaced, or extended without major disruptions to the overall architecture.

- **Independence:** Each subsystem is designed to operate independently, making it easier to test and deploy individual parts of the system.
- **Efficient Communication:** Clear interfaces between subsystems ensure smooth communication, reducing dependencies and minimizing the risk of issues spilling over from one subsystem to another.
- **Specialized Functionality:** By isolating specific functions within subsystems, we can tailor each to its role, optimizing performance and efficiency in each area without affecting the system as a whole.

In the Sports Facility Booking Management System, subsystems include User Management, Facility Management, Booking Management, Payment Processing, and Notification Handling. Each subsystem is designed with its own focused purpose, allowing it to handle specific tasks effectively while integrating seamlessly with other parts of the system. This modular structure is crucial for a scalable, adaptable, and maintainable solution.

## 17e Hardware/Software Mapping

This section provides an outline of how the software components of the Sports Facility Booking Management System will be assigned to different hardware resources. A clear mapping ensures the system runs efficiently, handling bookings, payments, and notifications smoothly while being secure, scalable, and reliable.

Mapping software to specific hardware helps optimize the system's performance and use of resources, delivering a better experience for users. This setup allows the system to handle high loads smoothly, supports growth, and keeps the system resilient and secure. For a platform that handles real-time bookings and sensitive payment information, strategic hardware mapping is essential for speed, availability, and security.

- **Performance:** Key components like the booking and payment modules need fast, reliable infrastructure to keep the system responsive and process transactions efficiently.
- **Scalability:** The hardware setup should allow the system to grow easily as the number of users and booking volume increase.
- **Reliability:** To minimize downtime, critical components should have backup and failover options.
- **Security:** Sensitive information, such as payment data and user credentials, should be processed on secure hardware to ensure data privacy and meet compliance standards.
- **Network Efficiency:** Servers should be positioned to reduce delays for users, especially in areas with many users, to keep the system fast.

The setup requirements would be like:

- **Web and Application Servers:** These servers host the main application, handling user interactions, searches, and bookings. Cloud-based options like AWS Services are ideal as they allow flexibility, scaling, and cost management. Load balancers distribute traffic across servers, helping the system handle more users smoothly.
- **Database Server:** This server stores essential data, like user profiles, booking records, facility details, and transactions. It needs a high-capacity setup with features for data security and backup, such as RAID configurations and daily backups. Managed cloud databases are recommended, as they provide automated scaling, high availability, and disaster recovery.
- **Payment Gateway:** To securely handle payments, this component is kept separate from the main application. Third-party payment processors like Razorpay, which is PCI-compliant, can manage this aspect. This setup allows for easy payment management, including refunds and transaction tracking, directly within the platform.

- **Notification Server:** This handles outgoing messages like booking confirmations and reminders. Services like AWS are suitable for managing high volumes of notifications and ensuring reliable delivery.
- **Load Balancer:** This distributes user requests across multiple web and application servers, preventing overload and reducing the risk of any single point of failure, especially during busy times.

This setup ensures each part of the system is running on the right type of infrastructure, keeping it secure, scalable, and responsive for users.

## 17f Data Dictionary

The Data Dictionary defines all data elements within the Sports Facility Booking Management System. It includes details for each data attribute, their data types, constraints, and descriptions, serving as a central reference to ensure consistent data handling and accurate implementation of database interactions. The dictionary is structured to align with software engineering best practices, facilitating clear communication among stakeholders and supporting system maintainability.

The Data Dictionary is essential for standardizing data definitions, ensuring all team members understand each data element's purpose, constraints, and structure. This clarity minimizes errors, enhances data integrity, and aids in debugging. Moreover, it acts as a critical tool for developers during implementation and for future maintainers, promoting long-term scalability and extensibility in the system's architecture.

In creating the Data Dictionary, the following considerations were applied:

- **Data Integrity:** Constraints and validation rules are defined to prevent data anomalies, ensuring each attribute aligns with real-world requirements.
- **Performance:** Choosing efficient data types helps optimize system performance, especially for frequently accessed data.
- **Scalability:** The dictionary is designed to allow easy expansion to support additional data elements or modifications as needed.
- **Security:** Sensitive data attributes, such as user passwords, are marked for encryption or hashing to ensure data protection.

**Table 2: Users Table**

<b>Table Name: Users Table</b>			
<b>Field Name</b>	<b>Data Type</b>	<b>Description</b>	<b>Constraints</b>
userID	INT	Unique identifier for each user.	Primary Key, Auto-incremented
name	VARCHAR(50)	Full name of the user.	Required, max length 50 characters
email	VARCHAR(50)	User's email address.	Required, Unique, valid email format
password	VARCHAR(255)	Hashed password for user authentication.	Required, encrypted or hashed
role	ENUM(Admin, Customer)	Role of the user in the system.	Enum
dateCreated	TIMESTAMP	Account creation date and time.	Auto-generated
status	BOOLEAN	User account status (active/inactive).	Default: active

**Table 3: Facilities Table**

<b>Table Name: Facilities Table</b>			
<b>Field Name</b>	<b>Data Type</b>	<b>Description</b>	<b>Constraints</b>
facilityID	INT	Unique identifier for each facility.	Primary Key, Auto-incremented
name	VARCHAR(100)	Facility's name.	Required, max length 100 characters
location	VARCHAR(100)	Address/location of the facility.	Required, max length 100 characters
type	VARCHAR(50)	Type of facility (e.g., court, field).	Required
description	VARCHAR(255)	Short description of the facility.	Optional, max length 255 characters
availability	BOOLEAN	Current availability status of the facility.	Default: available
pricePerHour	DECIMAL(10, 2)	Hourly rental price of the facility.	Required, non-negative

**Table 4: Staff Table**

Table Name: Staff Table			
Field Name	Data Type	Description	Constraints
staffID	INT	Unique identifier for each staff member.	Primary Key, Auto-incremented
name	VARCHAR(50)	Staff member's name.	Required, max length 50 characters
role	VARCHAR(30)	Staff member's role (e.g., Manager).	Required
contactInfo	VARCHAR(50)	Staff contact information.	Required
availability	BOOLEAN	Staff availability status.	Default: available

**Table 5: Bookings Table**

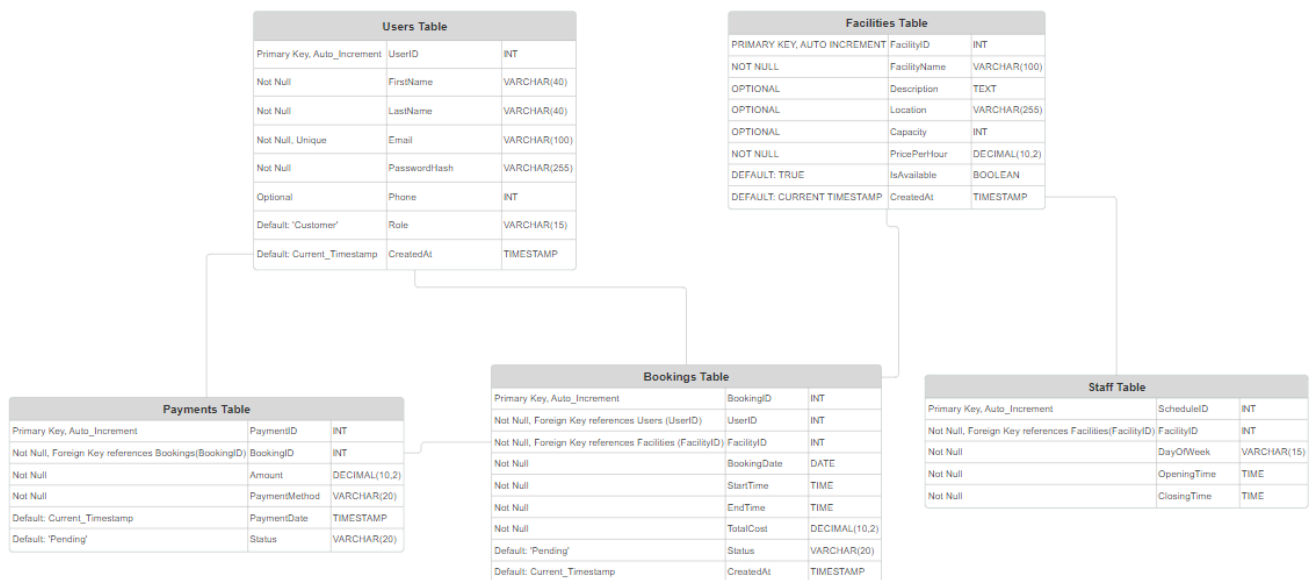
Table Name: Bookings Table			
Field Name	Data Type	Description	Constraints
bookingID	INT	Unique identifier for each booking.	Primary Key, Auto-incremented
userID	INT	ID of the user who made the booking.	Foreign Key referencing Users
facilityID	INT	ID of the facility booked.	Foreign Key referencing Facilities
bookingDate	DATE	Date of the booking.	Required
startTime	TIME	Start time of the booking.	Required
endTime	TIME	End time of the booking.	Required
status	ENUM(Confirmed, Cancelled)	Booking status.	Enum
dateCreated	TIMESTAMP	Booking creation timestamp.	Auto-generated

**Table 6: Payments Table**

Table Name: Payments Table			
Field Name	Data Type	Description	Constraints
paymentID	INT	Unique identifier for each payment.	Primary Key, Auto-incremented
bookingID	INT	Associated booking ID.	Foreign Key referencing Bookings



userID	INT	User who made the payment.	Foreign Key referencing Users
amount	DECIMAL(10, 2)	Amount paid for the booking.	Required, non-negative
status	ENUM(Paid, Refunded)	Payment status.	Enum
date	DATE	Date of the payment transaction.	Required
dateCreated	TIMESTAMP	Payment record creation timestamp.	Auto-generated



**Figure 8: ER Diagram**

This Data Dictionary provides a thorough definition of the data elements in the Sports Facility Booking Management System, supporting both the technical team and stakeholders with a unified understanding of the data requirements. It ensures data consistency, integrity, and security throughout the application.

## 17g Persistent Data Management

Persistent Data Management refers to the strategy and techniques used to store, retrieve, and manage data that needs to be saved across sessions in the Sports Facility Booking Management System. This ensures that key information, such as user details, booking records, and payment transactions, is consistently available even after the system is restarted. Properly managing persistent data is crucial for maintaining data integrity, performance, and security within the application.

The purpose of Persistent Data Management is to guarantee that essential data is securely stored and easily accessible, supporting system functionalities such as user authentication, booking history retrieval, and payment verification. Reliable data management underpins the application's ability to deliver a seamless user experience and maintain trust, especially in areas like payment processing and booking records where accuracy is critical.

Key considerations in Persistent Data Management for the Sports Facility Booking Management System include:

- **Data Integrity:** Ensuring that data is accurate, consistent, and free from corruption, especially during updates, deletions, and transactions.
- **Data Security:** Protecting sensitive data, such as user credentials and payment information, through encryption and controlled access to prevent unauthorized access and breaches.
- **Scalability:** Designing the data management system to handle growing amounts of data as more users, bookings, and facilities are added over time.
- **Performance:** Optimizing data storage and retrieval processes to minimize load times and ensure the application runs smoothly, even with large data volumes.
- **Backup and Recovery:** Establishing a robust backup and recovery plan to prevent data loss due to unforeseen incidents, ensuring business continuity.
- **Compliance:** Adhering to regulatory requirements, such as data protection laws, to ensure legal compliance in handling personal data.

In the Sports Facility Booking Management System, Persistent Data Management can be implemented as follows:

- **Database Selection:** A relational database management system is chosen to store structured data, including user profiles, facilities, bookings, payments, and notifications. This ensures efficient data storage and supports complex queries required for booking and payment management.
- **Entity Relationships:** Defining and enforcing foreign key relationships between tables (e.g., Users and Bookings, Bookings and Payments) ensures referential integrity across the database, preventing orphaned records.
- **Data Access Layer:** Implementing a Data Access Layer (DAL) in the application's backend abstracts direct database interactions. The DAL handles all Create, Read,

Update, and Delete (CRUD) operations, maintaining consistent data handling and reducing redundant code across the application.

- **Caching Strategy:** For frequently accessed data, such as facility details and user session information, a caching layer (e.g., Redis) can be employed to reduce database load and improve response times.
- **Security Practices:** Sensitive data fields, like passwords, are stored in hashed formats, and payment data is encrypted. Access to database resources is controlled through role-based permissions, and sensitive operations are audited to ensure compliance with security policies.
- **Backup and Disaster Recovery:** Daily automated backups of the database are scheduled, with recovery procedures documented and tested periodically to ensure quick restoration in case of data loss.

By adopting these practices, the Sports Facility Booking Management System ensures that data is effectively managed, secure, and consistently available to support application functionality and meet user needs.

## **17h Access Control and Security**

Access control and security mechanisms are fundamental components of the Sports Facility Booking Management System, ensuring that only authorized users can access, modify, or delete sensitive information. This includes user data, booking details, payment information, and facility management data. Implementing stringent access controls not only protects against unauthorized access but also helps in maintaining data integrity and compliance with relevant regulations.

The motivation behind establishing effective access control and security measures is multifaceted:

- **Data Protection:** Protect sensitive data, such as user information and financial details, from unauthorized access or breaches, ensuring privacy and compliance with data protection laws.
- **User Trust:** Foster user confidence in the system by demonstrating a commitment to security, which can enhance user engagement and loyalty.

- **Operational Integrity:** Maintain the integrity of the booking system by preventing unauthorized modifications to bookings, facilities, or user accounts, thus ensuring reliable and accurate operations.
- **Regulatory Compliance:** Adhere to industry regulations and standards, such as GDPR or PCI DSS, which often require strict access controls and data handling protocols.

When developing access control and security measures for the Sports Facility Booking Management System, the following considerations are paramount:

- **Authentication Mechanisms:** Implement strong user authentication methods, such as passwords and OTP verification, to verify user identities.
- **Role-Based Access Control (RBAC):** Define user roles (e.g., Admin, Customer, and Staff) and assign permissions based on these roles to ensure that users only access the data necessary for their functions.
- **Data Encryption:** Encrypt sensitive data at rest and in transit to protect it from unauthorized access and ensure secure communication channels.
- **Audit Logging:** Maintain comprehensive audit logs of user activities, including login attempts, data access, and modifications, to detect potential security incidents and support investigations.
- **Regular Security Assessments:** Conduct periodic security assessments and penetration testing to identify vulnerabilities in the system and ensure compliance with security policies.
- **User Training:** Provide ongoing security awareness training to users to help them recognize security threats, such as phishing attacks, and understand best practices for data handling.

Consider a scenario in the Sports Facility Booking Management System where different users have distinct access needs:

- **Admin User:**
  1. **Access Rights:** Full access to all system features, including user management, facility settings, booking modifications, and payment processing.
  2. **Authentication:** Required to use MFA for logging in.

3. **Logging:** All administrative actions are logged for auditing.
- **Customer User:**
    1. **Access Rights:** Can view available facilities, make bookings, and view their booking history. Cannot access or modify other users' data.
    2. **Authentication:** Standard password authentication with options for OTP login.
    3. **Logging:** User activities, such as booking and payment transactions, are logged.
  - **Staff User:**
    1. **Access Rights:** Can manage facility information, check availability, and assist customers with bookings. Restricted from accessing payment details.
    2. **Authentication:** Required to log in with a unique username and password.
    3. **Logging:** Staff actions are logged to ensure accountability.

By implementing these access control measures, the Sports Facility Booking Management System ensures that each user interacts with the system according to their role, thereby enhancing security while maintaining usability.

## **17i Global Software Control**

Global software control encompasses the strategies, tools, and processes used to manage software development, deployment, and operation across various regions. For the Sports Facility Booking Management System tailored for the Indian market, global software control ensures consistency, reliability, and adherence to local regulations. This involves overseeing software versions, configurations, deployments, and operational performance while catering to the specific needs of Indian users. The primary motivations for implementing global software control include:

- **Consistency:** To deliver a uniform user experience across the application, reinforcing brand identity and trust among users.
  - **Regulatory Compliance:** To meet Indian laws and regulations, particularly regarding data protection and privacy, which are critical in today's digital landscape.
  - **Risk Management:** To identify and mitigate potential risks associated with software deployment, including security vulnerabilities and operational failures.
-

- **Resource Optimization:** To leverage best practices in software development, enhancing efficiency and aligning efforts with organizational objectives.
- **Scalability:** To ensure the software can be effectively scaled to accommodate future growth and enhancements in the Indian market.

Key considerations for establishing global software control in the Indian context include:

- **Version Control:** Utilize a robust version control system to manage software iterations, ensuring all team members work with the latest updates.
- **Configuration Management:** Establish consistent processes for managing software configurations across development, testing, and production environments.
- **Deployment Automation:** Employ automation tools for deployments to minimize errors and ensure consistent performance across various environments.
- **Monitoring and Performance Metrics:** Implement monitoring solutions to track application performance, enabling timely identification of issues and optimization opportunities.
- **Localization:** Ensure the app is tailored to local languages, currencies, and cultural nuances to enhance user engagement and satisfaction.
- **Security Protocols:** Develop and maintain security standards that safeguard sensitive user data, complying with Indian regulations.
- **Collaboration Tools:** Use effective communication and collaboration platforms to facilitate seamless teamwork among development and operational teams.

In the context of the Sports Facility Booking Management System, here's how global software control can be implemented:

- **Version Control:** The development team employs Git for managing code changes, allowing for collaborative work while maintaining a comprehensive change history.
- **Deployment Automation:** CI/CD pipelines are established with tools like Jenkins to automate deployment processes, ensuring swift and error-free releases.
- **Localization:** The application is localized to support multiple Indian languages, ensuring users can navigate the platform comfortably in their preferred language.
- **Monitoring:** Tools such as New Relic are utilized to monitor application performance, providing insights that help optimize user experience and system reliability.

- **Compliance:** The app is designed to comply with Indian data protection regulations, incorporating features such as user consent management and data encryption.

By implementing these global software control measures, the Sports Facility Booking Management System can effectively meet the needs of Indian users, ensuring a seamless, secure, and reliable experience.

## 17j Boundary Conditions

Boundary conditions define the limits and constraints within which the Sports Facility Booking Management System operates. These conditions outline both functional and non-functional requirements, ensuring that the system adheres to defined standards and expectations. They delineate interactions between the system and external entities, such as users and third-party services, providing a clear framework for system functionality. Establishing boundary conditions serves several critical purposes:

- **System Integrity:** Ensures that the application operates within specified limits, thereby minimizing risks of failures and ensuring reliable performance.
- **User Expectations:** Sets clear guidelines for users regarding the capabilities and limitations of the system, enhancing user satisfaction and trust.
- **Performance Optimization:** Identifies performance thresholds and limitations, facilitating effective resource allocation and responsive system behaviour.
- **Compliance and Security:** Ensures adherence to regulatory requirements and security protocols, protecting user data and maintaining privacy.
- **Development Clarity:** Provides developers with a clear understanding of the system's constraints, aiding in accurate implementation and testing.

When defining boundary conditions for the Sports Facility Booking Management System, the following key considerations must be addressed:

- **Functional Boundaries:** Clearly specify core functionalities, including user registration, facility booking, payment processing, and reporting features.
- **User Roles:** Define the capabilities and limitations associated with various user roles (e.g., Admin, Customer, Staff) to control access to specific features and data.

- 
- **Performance Limits:** Establish thresholds for system performance, including maximum concurrent users, transaction processing times, and acceptable response times for user interactions.
  - **Integration Limits:** Identify the boundaries for integration with external systems, such as payment gateways and notification services, ensuring reliable and secure data exchange.
  - **Data Handling Constraints:** Specify limits on data storage, retention policies, and data access protocols to comply with legal requirements and protect user privacy.
  - **Error Handling:** Define procedures for how the system will respond to errors or exceptions, including user notifications and fall-back mechanisms.

For the Sports Facility Booking Management System, the following boundary conditions are proposed:

- **Functional Boundaries:**
    - Users may book facilities for a maximum duration of 8 hours per reservation.
    - Admin users have the ability to access and modify all bookings and facility information.
  - **User Roles:**
    - Customer users can view only their own booking history and details, with no access to other users' information.
    - Staff members can update facility availability but do not have permission to modify payment-related information.
  - **Performance Limits:**
    - The system shall support a minimum of 500 concurrent users without degradation of performance.
    - The maximum acceptable response time for user actions (e.g., booking a facility) shall not exceed 2 seconds.
  - **Integration Limits:**
    - Payment processing transactions must be completed within 30 seconds to enhance user experience.
    - Integration shall only occur with authorized payment gateways that adhere to PCI DSS standards.
-



- **Data Handling Constraints:**
  - User data shall be retained for a maximum of 2 years following account deletion, in compliance with applicable data protection regulations.
  - Passwords must be securely stored using strong hashing algorithms, with a minimum length of 8 characters and a requirement for a mix of character types.
- **Error Handling:**
  - Users shall receive informative error messages when a booking fails, including suggestions for corrective actions (e.g., selecting an alternative time slot).
  - The system shall log all errors for analysis by the development team to enhance reliability and user satisfaction.

By rigorously defining and adhering to these boundary conditions, the Sports Facility Booking Management System can ensure a robust, secure, and user-friendly experience tailored to the needs of its users in the market.

## 18 Subsystem Services

Subsystem services are the functional components within the Sports Facility Booking Management System, designed to operate semi-independently while contributing to the application's overall functionality. Each service addresses specific tasks, enhancing modularity, maintainability, and scalability, ultimately improving system robustness and performance.

Defining subsystem services offers several key advantages:

- **Modularity:** Facilitates efficient development and maintenance by allowing teams to focus on distinct functionalities.
- **Scalability:** Enables independent development and deployment, accommodating varying user demands.
- **Maintainability:** Simplifies updates and modifications, minimizing the impact on the overall system.
- **Performance:** Optimizes individual services for improved overall system performance.

- Responsibility Clarity: Clarifies roles within the development team, enhancing collaboration.

Key considerations in designing subsystem services include:

- Service Identification: Determine functionalities suitable for encapsulation as separate services.
- Inter-Service Communication: Define protocols for seamless data exchange between services.
- Data Management: Establish rules for data ownership and access to ensure consistency.
- Error Handling: Implement robust mechanisms to manage failures and provide user feedback.
- Security: Enforce security measures to protect sensitive data and prevent unauthorized access.
- Performance Metrics: Identify indicators to monitor service health and efficiency.

For the Sports Facility Booking Management System, the following subsystem services are defined:

1. User Management Service:

- Functionality: Handles user registration, authentication, and profile management.
- Inter-Service Communication: Validates user permissions with the Booking Service.
- Performance Metric: Monitors successful logins and registration failures.

2. Facility Management Service:

- Functionality: Manages facility information and availability.
- Inter-Service Communication: Provides availability data to the Booking Service.
- Performance Metric: Tracks facility updates and inquiries.

3. Booking Service:

- Functionality: Manages the booking process, including reservations.
- Inter-Service Communication: Processes transactions with the Payment Service.
- Performance Metric: Monitors booking success rates.

#### 4. Payment Service:

- **Functionality:** Handles payment processing and transaction tracking.
- **Inter-Service Communication:** Confirms bookings after payment with the Booking Service.
- **Performance Metric:** Tracks transaction success rates.

#### 5. Notification Service:

- **Functionality:** Sends notifications about bookings and promotions.
- **Inter-Service Communication:** Triggers notifications based on events from other services.
- **Performance Metric:** Monitors delivery success rates and user engagement.

By implementing these subsystem services, the Sports Facility Booking Management System can achieve enhanced modularity and performance, effectively meeting the needs of users in the Indian market.

## 19 User Interface

The user interface (UI) of the Sports Facility Booking Management System serves as the primary interaction point between users and the application. It includes design elements, layouts, and navigation features that facilitate user engagement and provide a smooth booking experience. A well-designed UI is crucial for enhancing usability, accessibility, and user satisfaction, allowing users to navigate the system efficiently and accomplish their tasks effectively.

Prioritizing an effective user interface is essential for several reasons:

- **User Engagement:** A visually appealing and intuitive UI encourages interaction and fosters a positive user experience.
  - **Usability:** A clear and organized layout simplifies navigation, enabling users to quickly locate information and complete tasks.
  - **Accessibility:** An inclusive UI accommodates users with diverse abilities, broadening the user base.
  - **Brand Image:** A cohesive and professional UI reinforces the brand's identity and credibility.
  - **User Retention:** A positive experience increases satisfaction and loyalty, encouraging repeat usage of the system.
-

---

When designing the user interface for the Sports Facility Booking Management System, several key considerations should be made:

- **User-Centric Design:** Focus on the needs and preferences of target users to create an intuitive and engaging experience.
- **Consistency:** Maintain uniformity in design elements, including colours, fonts, and button styles, to ensure a cohesive experience.
- **Responsiveness:** Design the UI to adapt seamlessly across various screen sizes and devices, providing a consistent user experience.
- **Feedback Mechanisms:** Incorporate visual and auditory feedback for user actions (e.g., confirmations, error messages) to enhance usability.
- **Accessibility Standards:** Adhere to WCAG (Web Content Accessibility Guidelines) to ensure compatibility for users with disabilities, including keyboard navigation and screen reader support.
- **Visual Hierarchy:** Utilize design principles like contrast, size, and spacing to guide user attention to important elements and actions.

### **Selected Model: User Profile Model**

The User Profile Model is particularly significant for the Sports Facility Booking Management System. This model, established by a software engineer, captures the diverse characteristics of end-users, which include:

- **Age:** Understanding the age demographics helps tailor the interface and features to suit different user groups.
- **Gender:** Recognizing gender-specific preferences can guide design decisions and personalization options.
- **Physical Abilities:** Accommodating varying physical abilities ensures accessibility and enhances usability for all users.
- **Education:** Considering the educational background allows for appropriate complexity in language and functionality.
- **Cultural Background:** Acknowledging cultural differences can inform design choices that respect user diversity.
- **Motivation and Goals:** Understanding users' motivations and goals enables the system to align features and functionalities with user expectations.

- **Personality:** Recognizing different personality types helps in creating a more engaging and user-friendly interface.

By leveraging the User Profile Model, the Sports Facility Booking Management System can tailor the UI to meet the specific needs and preferences of its users, ultimately leading to enhanced user satisfaction and engagement.

For the Sports Facility Booking Management System, the user interface may incorporate the following elements:

- **Dashboard:** A clean, organized layout displaying key information such as upcoming bookings, facility availability, and notifications.
- **Booking Page:** A streamlined, step-by-step booking process with clear instructions, allowing users to select facilities, dates, time slots, and payment options.
- **User Profile:** An easily navigable profile management section where users can update their information, view booking history, and manage preferences.
- **Search and Filter Options:** An intuitive search bar with filter options (e.g., facility type, location, availability) to help users quickly find suitable venues.
- **Mobile Responsiveness:** A responsive design that optimizes the UI for mobile devices, ensuring easy navigation and booking on-the-go.
- **Notifications and Alerts:** Visual notifications for booking confirmations, cancellations, and promotional offers that are easily accessible within the UI.

By emphasizing the User Profile Model and focusing on a well-designed user interface, the Sports Facility Booking Management System can enhance user satisfaction, streamline booking processes, and foster a positive interaction between users and the application.

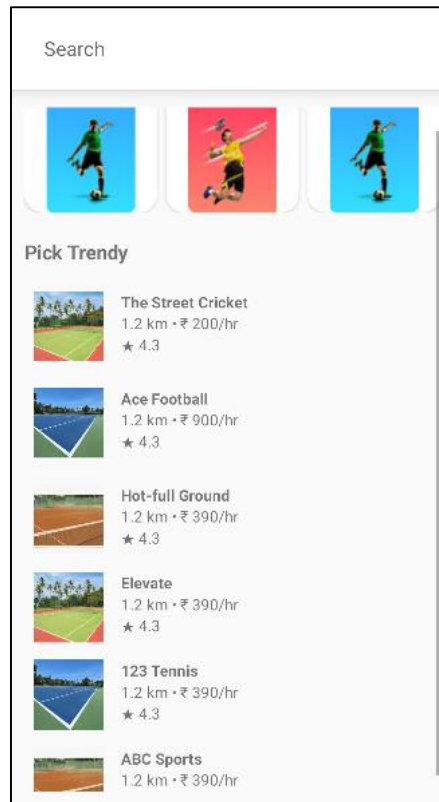


Figure 9: UI Design for Home Page

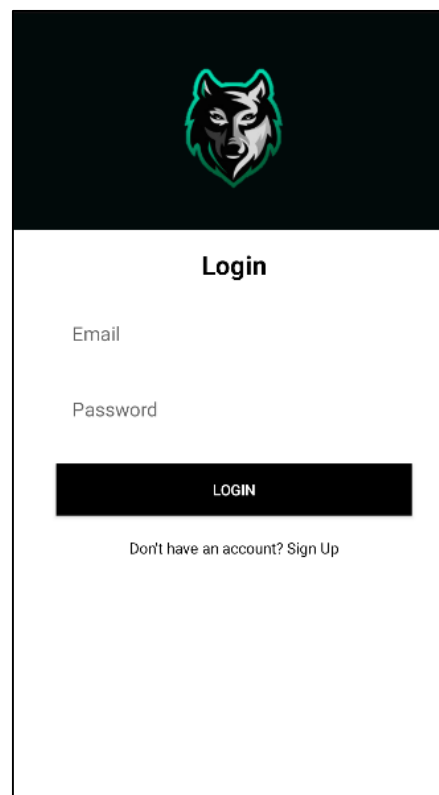
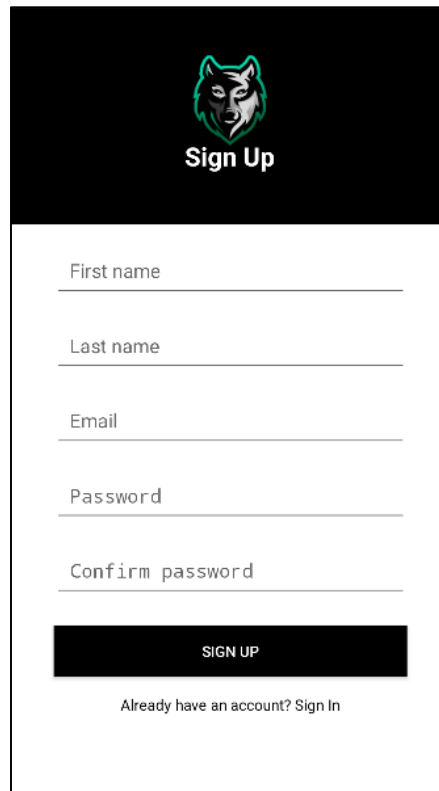


Figure 10: UI Design for Login Page



The Sign Up page features a dark header with a green wolf head logo and the text "Sign Up". Below the header, there are five input fields for "First name", "Last name", "Email", "Password", and "Confirm password". A black "SIGN UP" button is positioned below the fields. At the bottom, a link reads "Already have an account? Sign In".

Figure 11: UI Design for Sign Up Page



The Facility Details page for "ABC Sports" includes a photo of a green tennis court with a red border. The text below the photo provides details: "37K users booked this venue till date.", "Near Navrachana University, Bhayli, Vadodara - 390000", "₹700 Onwards · 12:00 AM - 11:59 PM", and "4.5 41 ratings". Under "Available Sports:", there are icons for soccer and tennis. The "Offers" section lists two discounts: "Flat 40% off upto ₹1000 Code: NEW40" and "Flat 10% off upto ₹175 Code: K10", each with a "COPY" button. There is also a "Split Payment" option with a "SPLIT" button. A "Facilities" section is at the bottom, followed by a black "BOOK MY SLOT" button.

Figure 12: UI Design for Facility Details Page

---

## IV Test Plans

### 20 Features to be Tested/ not to be Tested

This section defines the features of the Sports Facility Booking Management System that will be included in testing and those that will be excluded. It establishes a clear scope for testing activities, ensuring that critical components are thoroughly validated while non-essential or out-of-scope areas are clearly identified to avoid unnecessary testing. Identifying features to be tested (and those not to be) is essential for the following reasons:

- **Resource Allocation:** Ensures testing resources are focused on areas critical to functionality, performance, and security.
- **Risk Management:** Helps prioritize testing of high-risk areas while excluding low-risk or stable areas.
- **Efficiency:** Prevents redundant or low-impact testing, allowing for streamlined testing cycles.
- **Clarity:** Provides testers and stakeholders with a well-defined testing boundary, avoiding confusion and ensuring that expectations are aligned.

In selecting features to be tested or excluded, consider the following:

- **Criticality:** Features that are essential to core functionality (e.g., booking process, payment integration) should be prioritized.
- **User Impact:** High-impact areas that affect user experience and satisfaction (e.g., facility search, user profiles).
- **Complexity:** Complex components with multiple dependencies may require additional testing.
- **Known Stability:** Features or components with stable history and low rates of modification may be excluded from intensive testing.
- **Regulatory Requirements:** Any features tied to compliance or security standards should be tested to meet regulatory expectations.



**Table 7: Feature Testing Scope**

<b>Feature</b>	<b>Tested / Not Tested</b>	<b>Rationale</b>
Facility Search and Filter	Tested	Core functionality for locating venues; crucial to user satisfaction.
User Registration & Login	Tested	Essential for user access; includes security validation and session management.
Booking Management	Tested	Primary feature for scheduling bookings; includes slot availability and conflict resolution.
Payment Processing	Tested	Core feature requiring validation for payment accuracy, security, and compliance.
Notifications	Tested	Key for user engagement, especially for booking confirmations and reminders.
Facility Manager Reports	Tested	Important for venue analytics, impacting business decisions for managers.
Customer Support Chat	Tested	Ensures users can seek help effectively; critical for user satisfaction.
Weather Integration	Not Tested	Excluded per project scope; third-party service not essential to core functionality.
Social Media Sharing	Not Tested	Non-essential feature; low impact on core functionality.
System Admin Settings	Tested	Ensures configuration options work as expected; crucial for maintaining system integrity.

This testing scope allows for a strategic allocation of resources, focusing on critical user-facing and functional components while deprioritizing areas outside of core functionality.

## 21 Pass Fail Criteria

The Pass/Fail Criteria section defines the standards used to evaluate whether a test case or feature in the Sports Facility Booking Management System has met the expected results. By establishing clear pass/fail criteria, testers can assess system functionality against defined benchmarks, ensuring that only components that meet quality expectations are approved for deployment. Clear pass/fail criteria are essential for:

- **Objective Assessment:** Providing a standardized way to determine if features meet functional and performance requirements.
- **Quality Assurance:** Ensuring that only components that meet the quality standards move forward, reducing potential issues post-deployment.
- **Consistency:** Enabling all testers to evaluate features consistently, which helps maintain high-quality outcomes across all testing phases.
- **Risk Management:** Reducing the risk of defects reaching production by setting thresholds for passing or failing test scenarios.

Key factors for defining pass/fail criteria include:

- **Requirement Fulfilment:** Ensuring each feature operates according to specifications outlined in the requirements document.
- **Error-Free Execution:** A feature passes if it completes without errors or exceptions; critical defects result in a fail.
- **Performance Benchmarks:** Testing response times, loading times, and overall performance against established standards.
- **User Experience:** Assessing usability elements, ensuring that the feature meets accessibility standards and provides a seamless experience.
- **Security Compliance:** Verifying that sensitive components, such as login and payment processing, adhere to security protocols and data protection standards.

**Table 8: Pass/Fail Criteria**

<b>Feature/Test Case</b>	<b>Pass Criteria</b>	<b>Fail Criteria</b>
Facility Search and Filter	Returns accurate, relevant results within 2 seconds.	Search results are inaccurate, incomplete, or exceed 2 seconds.
User Registration & Login	Registers/logs in users successfully, validating credentials.	Fails to authenticate or validate credentials correctly.
Booking Management	Books, modifies, and cancels slots without conflicts or errors.	Booking conflicts, errors, or failure to update slot availability.
Payment Processing	Processes payments successfully with correct amount and receipt.	Payment fails, incorrect amounts, or transaction errors.
Notifications	Sends timely, accurate notifications for booking confirmations.	Fails to send or sends incorrect notifications.
Facility Manager Reports	Generates reports accurately, meeting performance benchmarks.	Report data is inaccurate or takes excessive time to generate.
Customer Support Chat	Provides functional chat interface with real-time response.	Chat function fails or experiences significant delays in response.
System Admin Settings	Applies configuration changes accurately without affecting other features.	Configuration changes lead to system errors or data corruption.

By applying these pass/fail criteria, the testing team can ensure that each component meets functional, performance, and usability standards, allowing the system to provide a reliable and user-friendly experience.

---

## 22 Test Scenarios

### 1. Search Facilities

- Test search with a valid facility name/location (positive).
- Test search with an invalid facility name/location that yields no results (negative).
- Test search with empty input (negative).
- Test search with special characters or long input exceeding character limits (boundary values).

### 2. View Facility Details

- Test viewing details of a facility with a valid facility ID (positive).
- Test attempting to view facility details with an invalid facility ID (negative).
- Test viewing facility details without proper user authentication (negative).

### 3. Check Facility Availability

- Test checking availability for a future date for a valid facility (positive).
- Test checking availability for a past date, which should not allow booking (negative).
- Test checking availability for a non-existent facility ID (negative).
- Test for boundary dates, such as the first and last day a facility is open for booking.

### 4. Select Booking Slot

- Test selecting an available slot (positive).
- Test attempting to select a slot that is already booked (negative).
- Test selecting a slot outside the facility's operating hours (negative).
- Test selecting a slot on a fully-booked date (boundary values).

### 5. Payment Processing

- Test making a successful payment with valid card details (positive).
- Test payment with an invalid card number or expired card (negative).
- Test payment with an unsupported payment method (negative).
- Test handling network issues during payment (boundary/edge case).
- Test payment with a low balance or declined transaction (negative).

### 6. Booking Confirmation

- Test receiving a booking confirmation after successful payment (positive).

- Test attempting to confirm a booking when payment fails (negative).
- Test booking confirmation with delayed processing or network latency (boundary/edge case).

#### 7. Cancel Booking

- Test successfully cancelling a booking before the start time (positive).
- Test attempting to cancel a booking after the start time (negative).
- Test cancelling a booking that does not exist or has already been cancelled (negative).
- Test cancelling a booking during peak load times to check system response (boundary/edge case).

These scenarios will ensure that the Facility Booking Module undergoes rigorous testing to confirm that it functions as expected under different conditions, including positive cases, negative cases, and boundary cases. This analysis will aid in designing comprehensive test cases.

## 23 Test Cases

### Test Case 1: Search Facilities

#### 1. Test Case ID: TC001

- Description: Test search with a valid facility name.
- Preconditions: User is on the search page.
- Test Steps:
  1. Enter a valid facility name.
  2. Click the "Search" button.
- Expected Result: The system displays a list of matching facilities.
- Status: Pass/Fail

#### 2. Test Case ID: TC002

- Description: Test search with an invalid facility name.
- Preconditions: User is on the search page.
- Test Steps:
  1. Enter an invalid facility name.
  2. Click the "Search" button.
- Expected Result: The system displays a message, "No facilities found."

- Status: Pass/Fail

### 3. Test Case ID: TC003

- Description: Test search with an empty input.
- Preconditions: User is on the search page.
- Test Steps:
  1. Leave the search field empty.
  2. Click the "Search" button.
- Expected Result: The system prompts the user to enter a search term.
- Status: Pass/Fail

## Test Case 2: View Facility Details

### 1. Test Case ID: TC004

- Description: View details of a facility with a valid ID.
- Preconditions: Facility is available and searchable.
- Test Steps:
  1. Select a facility from the search results.
  2. Click "View Details."
- Expected Result: Facility details page displays the facility's amenities, pricing, and available time slots.
- Status: Pass/Fail

### 2. Test Case ID: TC005

- Description: Attempt to view facility details with an invalid facility ID.
- Preconditions: Facility ID is incorrect.
- Test Steps:
  1. Attempt to view details of a non-existent facility.
- Expected Result: The system displays an error message, "Facility not found."
- Status: Pass/Fail

## Test Case 3: Check Facility Availability

### 1. Test Case ID: TC006

- Description: Check availability for a future date.
  - Preconditions: Facility is available and has open slots on the future date.
-

- Test Steps:
    1. Select a future date.
    2. Click "Check Availability."
  - Expected Result: The system displays available slots for the selected date.
  - Status: Pass/Fail
2. Test Case ID: TC007
- Description: Check availability for a past date.
  - Preconditions: The system should not allow booking for past dates.
  - Test Steps:
    1. Select a past date.
    2. Click "Check Availability."
  - Expected Result: The system displays a message, "Booking not allowed for past dates."
  - Status: Pass/Fail

#### Test Case 4: Select Booking Slot

1. Test Case ID: TC008
- Description: Select an available booking slot.
  - Preconditions: The selected slot is open and within facility hours.
  - Test Steps:
    1. Choose an available slot.
    2. Confirm slot selection.
  - Expected Result: The selected slot is locked in for booking.
  - Status: Pass/Fail
2. Test Case ID: TC009
- Description: Attempt to select an already booked slot.
  - Preconditions: The slot has already been reserved.
  - Test Steps:
    1. Attempt to select a booked slot.
  - Expected Result: The system displays an error, "Slot not available."
  - Status: Pass/Fail
-

---

### Test Case 5: Payment Processing

#### 1. Test Case ID: TC010

- Description: Make payment with valid card details.
- Preconditions: User has valid payment information.
- Test Steps:
  1. Enter valid card information.
  2. Click "Pay Now."
- Expected Result: Payment is successful, and the booking is confirmed.
- Status: Pass/Fail

#### 2. Test Case ID: TC011

- Description: Make payment with an invalid card.
- Preconditions: Card information is incorrect or expired.
- Test Steps:
  1. Enter invalid card details.
  2. Click "Pay Now."
- Expected Result: The system displays a payment failure message.
- Status: Pass/Fail

### Test Case 6: Booking Confirmation

#### 1. Test Case ID: TC012

- Description: Receive booking confirmation after successful payment.
- Preconditions: Payment is successful.
- Test Steps:
  1. Complete payment.
- Expected Result: A confirmation message displays, and the user receives a confirmation email or SMS.
- Status: Pass/Fail

#### 2. Test Case ID: TC013

- Description: Attempt to confirm booking with a failed payment.
- Preconditions: Payment has failed.
- Test Steps:
  1. Attempt booking confirmation after a failed payment.



- 
- Expected Result: System prompts a message, "Payment required to confirm booking."
  - Status: Pass/Fail

### Test Case 7: Cancel Booking

#### 1. Test Case ID: TC014

- Description: Cancel booking before the start time.
- Preconditions: User has an active booking.
- Test Steps:
  1. Go to "My Bookings."
  2. Select the booking.
  3. Click "Cancel."
- Expected Result: Booking is successfully cancelled, and user receives cancellation confirmation.
- Status: Pass/Fail

#### 2. Test Case ID: TC015

- Description: Attempt to cancel booking after the start time.
- Preconditions: Booking start time has passed.
- Test Steps:
  1. Go to "My Bookings."
  2. Select the booking.
  3. Click "Cancel."
- Expected Result: The system displays a message, "Cannot cancel after booking start time."
- Status: Pass/Fail

These structured test cases enable thorough and repeatable testing of core functionalities, ensuring the system meets both functional and non-functional requirements before launch.

---

## 24 Testing Schedule

The Testing Schedule section outlines the timeline for executing all test phases for the Sports Facility Booking Management System. It includes estimated durations, start and end dates, resource allocation, and dependencies for each test phase, ensuring a structured and timely testing process. A well-defined testing schedule is essential to:

- **Timely Project Delivery:** Aligns testing milestones with the project timeline to avoid delays and ensure an on-time release.
- **Resource Management:** Allocates the right resources at the right time, optimizing availability and productivity.
- **Dependency Management:** Ensures that testing phases follow necessary development and deployment steps, reducing blockers.
- **Stakeholder Alignment:** Provides visibility for stakeholders, keeping everyone informed of the testing progress and any adjustments.

The following table outlines the structured testing schedule designed to validate the core functionalities and ensure reliability for the Sports Facility Booking Application. Each test case is strategically defined to cover a wide range of functionalities, from searching for facilities and checking availability to making payments, receiving confirmations, and handling cancellations. This testing phase is scheduled from December 1, 2024, to December 6, 2024, and involves various testing stages: Functional Testing, Payment Testing, Booking Confirmation, and Cancellation Testing. Each test case includes specific descriptions, anticipated durations, and expected outcomes, helping us identify and resolve potential issues early in the process. This structured approach ensures that the application meets performance, usability, and security standards before deployment.

**Table 9: Testing Schedule**

Test Case ID	Description	Testing Phase	Start Date	End Date	Duration	Status
TC001	Test search with a valid facility name.	Functional Testing	01-12-2024	01-12-2024	1 day	Pass/Fail
TC002	Test search with an invalid facility name.	Functional Testing	01-12-2024	01-12-2024	1 day	Pass/Fail
TC003	Check availability for a future date.	Functional Testing	02-12-2024	02-12-2024	1 day	Pass/Fail
TC004	Check availability for a past date.	Functional Testing	02-12-2024	02-12-2024	1 day	Pass/Fail
TC005	Select an available booking slot.	Functional Testing	03-12-2024	03-12-2024	1 day	Pass/Fail
TC006	Attempt to select an already booked slot.	Functional Testing	03-12-2024	03-12-2024	1 day	Pass/Fail
TC007	Make payment with valid card details.	Payment Testing	04-12-2024	05-12-2024	2 day	Pass/Fail
TC008	Make payment with an invalid card.	Payment Testing	04-12-2024	05-12-2024	2 day	Pass/Fail

TC009	Receive booking confirmation after successful payment.	Booking Confirmation	04-12-2024	05-12-2024	2 day	Pass/Fail
TC010	Attempt to confirm booking with a failed payment.	Booking Confirmation	04-12-2024	05-12-2024	2 day	Pass/Fail
TC011	Cancel booking before the start time.	Cancellation Testing	05-12-2024	05-12-2024	1 day	Pass/Fail
TC012	Attempt to cancel booking after the start time.	Cancellation Testing	06-12-2024	06-12-2024	1 day	Pass/Fail

These structured test cases enable thorough and repeatable testing of core functionalities, ensuring the system meets both functional and non-functional requirements before launch.

## V Project Issues

### 25 Project Planning

#### 25a Project Planning

For the Sports Facility Booking Management System, we will use the Agile model as the software development life cycle (SDLC). The Agile approach emphasizes iterative development, continuous feedback, and close collaboration with stakeholders, allowing us to adapt to changes and deliver incremental value. The high-level process diagram below illustrates the flow of tasks and their interactions in each Agile sprint cycle.

The Agile model was chosen to ensure flexibility in meeting the evolving needs of users, including facility managers and sports enthusiasts. Agile allows for faster delivery of incremental functionality, frequent reassessment of project goals, and close involvement of stakeholders, ensuring the product aligns with user requirements throughout the development process. Using Agile for this project requires attention to the following factors:

- **Iteration and Feedback:** Regular sprints enable feedback from stakeholders and iterative improvements to refine features based on user needs and market demands.
- **Collaboration:** Continuous collaboration between developers, designers, and stakeholders is essential to ensure alignment with the project's goals and adaptability to changing requirements.
- **Incremental Delivery:** Each sprint will produce a working increment of the application, enabling testing, validation, and user feedback at each stage.
- **Scope Management:** Agile projects can encounter scope changes frequently, so managing and prioritizing features effectively is critical.

#### Life Cycle Stages and Tasks (Agile Sprint Cycle)

##### 1. Sprint Planning

- Define sprint goals and deliverables
- Prioritize user stories and tasks for the sprint
- Allocate resources and estimate time

---

## 2. Development & Implementation

- Develop features and functionalities (e.g., user registration, booking, payment)
- Write code for frontend and backend, including integration with Razorpay for payments
- Conduct peer code reviews for quality assurance

## 3. Testing

- Perform unit and integration testing for new features
- Conduct user acceptance testing (UAT) with stakeholders
- Address defects and improvements from testing feedback

## 4. Integration & Deployment

- Integrate developed features into the main codebase
- Deploy the updated application to a staging or testing environment for review
- Implement security measures and ensure data integrity

## 5. Review & Retrospective

- Review sprint outcomes and gather feedback from stakeholders
- Conduct retrospective with the development team to identify improvements
- Document lessons learned and adjustments for the next sprint

## 6. Release Increment

- Release completed features to the production environment
- Ensure that the release includes stable and well-tested functionality
- Update documentation and release notes for users and stakeholders

### **Time and Resource Estimates (for each Sprint)**

- Sprint Planning: 1 day, product owner and development team
- Development & Implementation: 7-10 days, 4 developers
- Testing: 3-4 days, 2 testers
- Integration & Deployment: 1-2 days, 1 DevOps engineer
- Review & Retrospective: 1 day, development team and stakeholders
- Release Increment: 1 day, product owner and release manager

---

## Considerations for Data Conversion, User Training, and Cutover

- **Data Conversion:** Ensure accurate data migration with rollback options in case of errors, especially during major updates or releases.
- **User Training:** Develop training resources incrementally, keeping users updated on new features after each release.
- **Cutover:** Plan for minimal downtime, especially during high-usage periods. Coordinate cutover tasks with user communication to ensure a smooth transition.

By leveraging Agile, we ensure that the Sports Facility Booking Management System remains flexible, user-focused, and capable of adapting to new requirements or feedback, leading to a successful and timely delivery.

## 25b Planning of Development Phases

For the development of the Sports Facility Booking Management System, we have divided the project into multiple phases. Each phase covers specific functional and non-functional requirements, operating environment components, and ensures a structured approach to achieve the final product. Below is a high-level outline of each development phase, with details on the timeline, objectives, and the components required to support development and deployment.

Breaking down the development into phases allows for effective project management and clear milestones. Each phase is designed to progressively build and refine the system, ensuring that we meet both functional and non-functional requirements while preparing the necessary operating environment. By structuring the development into phases, we can track progress, address any challenges early, and manage resources efficiently.

During each phase, specific hardware, software, and other devices may be necessary to support development and testing. The exact list of components may evolve as we progress through design and implementation. Additionally, we need to ensure that development aligns with the operating environment requirements to minimize compatibility issues and streamline the deployment process.

---

## Development Phases for the Sports Facility Booking Management System

### 1. Phase 1: Core Functionality Development

- Required Operational Date: 31 December 2024
- Operating Environment Components Included:
  - Basic cloud infrastructure for development (e.g., AWS for server hosting)
  - Development environment setup (IDE, version control, CI/CD pipeline)
- Functional Requirements Included:
  - User Registration & Authentication
  - Facility Listing Management
  - Initial Booking Functionality
- Non-functional Requirements Included:
  - Basic security protocols for user data handling
  - System scalability to handle initial user load

### 2. Phase 2: Payment Integration & Advanced Booking Features

- Required Operational Date: 28 February 2025
- Operating Environment Components Included:
  - Integration of Razorpay payment gateway for handling transactions
  - Database configuration for secure transaction data storage
- Functional Requirements Included:
  - Payment Processing
  - Enhanced Booking Management (cancellation, rescheduling)
  - Facility Availability Updates
- Non-functional Requirements Included:
  - Compliance with payment security standards (e.g., PCI DSS)
  - High availability for payment-related services to minimize transaction errors

### 3. Phase 3: Notification System & User Interface Enhancements

- Required Operational Date: 15 April 2025
  - Operating Environment Components Included:
    - Integration of notification services (e.g., SMS and email)
-



- 
- Frontend development environment for UI/UX improvements
  - Functional Requirements Included:
    - Notification System (email and SMS notifications for bookings)
    - User Profile Management
    - Improved Search & Filter for Facility Listings
  - Non-functional Requirements Included:
    - Usability and accessibility compliance
    - Notification system reliability and timeliness
4. Phase 4: Reporting, Analytics, and Admin Dashboard
- Required Operational Date: 15 June 2025
  - Operating Environment Components Included:
    - Setup of analytics tools (e.g., Google Analytics, custom reporting)
    - Database updates for logging and analytics
  - Functional Requirements Included:
    - Admin Dashboard for Facility Managers
    - Reporting & Analytics on Bookings and Payments
  - Non-functional Requirements Included:
    - Data consistency and integrity for accurate reporting
    - Performance optimization for real-time data analytics
5. Phase 5: Final Testing, Deployment, and User Training
- Required Operational Date: 31 July 2025
  - Operating Environment Components Included:
    - Production server setup and deployment tools
    - User training environment and resources
  - Functional Requirements Included:
    - Comprehensive system testing
    - Final bug fixes and feature adjustments
    - Deployment and launch readiness
  - Non-functional Requirements Included:
    - System reliability and stability for launch
    - User training materials and on-boarding support
-

Each phase will ensure that specific components of the Sports Facility Booking Management System are built, tested, and refined before progressing. This structured approach will help us manage project complexities and deliver a high-quality, reliable application to our users.

## **26 Migration to New Product**

### **26a Requirements for Migration to the New Product**

This section details the activities required to migrate from the current booking methods to the new Sports Facility Booking Management System. It outlines the phases of implementation, data conversion needs, backup requirements, and a proposed timeline for each step.

The objective is to clearly define all tasks and requirements necessary for a seamless transition from existing manual, phone-based, and spreadsheet methods, as well as any basic software currently in use. Identifying these tasks will assist in planning and ensure an efficient migration that minimizes disruptions for users.

#### **1. Phased Implementation**

- Phase 1: Initial rollout to selected facilities currently using manual logs and phone reservations. This phase will introduce digital booking features and data entry tools.
- Phase 2: Expansion to facilities using spreadsheets and basic software, integrating their existing data into the new system.
- Phase 3: Full-scale deployment, opening access to all facilities, allowing direct user comparison of facilities, and supporting a broad search functionality for end-users.

#### **2. Data Conversion**

- Facilities using spreadsheets or basic software systems will need their existing data transferred to the new system. This may require custom scripts to import spreadsheet data directly into the system's database.
- Conversion scripts will be tested for accuracy and will run during off-peak hours to reduce impact on daily operations.

### 3. Manual Backup During Installation

- Facilities will maintain manual logs for reservations during the initial deployment of the new system to ensure no data loss or booking disruption. After successful implementation, these logs can be phased out.

### 4. Component Timeline

- Phase 1: Initial installation, estimated completion within three months from the start of deployment.
- Phase 2: Data migration and broader system access, estimated completion by six months.
- Phase 3: Full deployment with live user access and real-time booking features across all facilities, estimated completion by twelve months.

### 5. Parallel Operation Requirement

- During migration, a parallel run of the old and new systems may be necessary to ensure users have a reliable backup in case of issues. This will allow time for training staff and users to adopt the new system.

### 6. Additional Staff and Training

- Temporary IT support and trainers may be required during each phase to assist facilities and users with the transition.

### 7. Decommissioning the Old System

- Facilities will phase out their old logs, phone booking systems, and spreadsheets once the new system is fully operational.

**Table 10: Timetable for Migration to New Product**

Phase	Activities	Timeline
<b>Phase 1</b>	Initial setup and pilot rollout to selected facilities transitioning from manual logs and phone reservations.	Weeks 1 - 4
	Tasks: Configure core booking features, provide initial data entry tools, and train staff.	
<b>Phase 2</b>	Data migration from spreadsheets and integration with basic software.	Weeks 5 - 8
	Tasks: Convert booking data, ensure data accuracy, and implement real-time update integrations.	

<b>Phase 3</b>	Full deployment with live user access, including facility comparison and booking features.	Weeks 9 - 12
	Tasks: Enable full user access, finalize real-time booking functionalities, and eliminate manual backups.	
<b>Final Review</b>	Monitor system performance, finalize training, and decommission old systems.	Weeks 13 - 14

This timeline ensures the system is implemented and fully operational in under four months, allowing for faster adoption and earlier user access.

## 26b Data That Has to Be Modified or Translated for the New System

This section outlines data translation tasks necessary to transition from current methods to the Sports Facility Booking Management System, ensuring data accuracy and usability in the new environment.

This data translation ensures that legacy data from existing formats (e.g., manual logs, spreadsheets, or basic software) will be compatible with the new system, avoiding data loss and reducing inconsistencies.

- **Current Data Technology**
  - **Manual Logs:** Physical booking records and phone-based reservations with handwritten notes.
  - **Spreadsheets:** Digital records in applications like Excel, updated manually by staff.
  - **Basic Software Systems:** Proprietary or facility-owned booking software with minimal interactivity or real-time capabilities.
- **New Data Technology**
  - **Centralized Booking Database:** A structured database with tables for user profiles, facilities, bookings, and payments, supporting real-time updates and queries.
  - **Cloud Storage and Backup:** Ensures secure data handling, accessibility, and backups for essential booking and user data.
- **Data Translation Tasks**
  - **Manual Log Data Entry:** Digitize booking records from physical logs to the centralized database, ensuring accurate input of past booking records.

- 
- Spreadsheet Import: Automate data import from spreadsheets into the new system, converting records into a format that aligns with the new booking database tables.
  - Software Data Migration: Extract and transform data from existing software to the new database, preserving fields like user data, booking history, and facility details.
  - Foreseeable Problems
    - Data Inconsistencies: Errors may arise due to manual entry, requiring validation.
    - Loss of Historical Data: Legacy software might lack standardized data formats, complicating data migration.
    - Duplicate Entries: Multiple booking records for the same event or user may need merging to avoid redundancy.

When adding new entities or updating the data dictionary, ensure alignment with data currently held in manual, digital, or software formats.

## 27 Risks

### 27a Risk Identification

For the Sports Facility Booking System project, a thorough risk identification process is essential to ensure project success and reduce potential disruptions.

#### 1. Inaccurate Cost Estimation

- Content: The risk of underestimating or overestimating project costs.
- Motivation: Accurate cost estimation is crucial to avoid budget shortfalls or overspending.
- Details: Developing a booking system for sports facilities involves resources such as software licenses, developer hours, and infrastructure costs (e.g., servers, databases). If costs are underestimated, we risk budget overruns that may require additional funding or lead to project delays. Overestimation, however, could result in unnecessary resource allocation, impacting overall ROI. To mitigate this, ongoing financial assessments are essential throughout the project.

## 2. Schedule Pressure

- Content: The risk of time constraints causing rushed development.
- Motivation: Schedule pressure could result in shortcuts or reduced quality in development.
- Details: The demand for the application in the market may drive tight deadlines, particularly during key sports seasons when facility bookings are highest. If milestones are not met, the development team may be pressured to meet deadlines, leading to potential quality issues, such as increased bugs or technical debt. Implementing phased rollouts and timeline buffers will be key to addressing this risk.

## 3. Dependency on Third-Party Services

- Content: The risk of reliance on external services, such as payment gateways and location services.
- Motivation: Ensuring continuous service availability and user experience.
- Details: The application relies on third-party services for payment processing, map and location features, and potentially for user authentication. Interruptions or performance issues with these services could lead to booking failures, user dissatisfaction, and negative feedback, impacting app credibility. Developing fall-back protocols or alternative service providers is essential to prevent these disruptions from affecting the user experience.

## 4. Creeping User Requirements

- Content: The risk of continuous changes in user requirements beyond the initial scope.
- Motivation: Scope creep can disrupt timelines and increase costs.
- Details: As user feedback is incorporated, especially during UAT, there may be requests for additional features (e.g., expanded payment options or enhanced filtering). While feature enhancements can improve the product, uncontrolled additions can lead to extended timelines, increased costs, and complexity. A formal change control process will help ensure that only essential features are included during each development phase.

#### 5. Inadequate User Acceptance Testing (UAT)

- Content: The risk that insufficient UAT leads to a system that does not fully meet user needs or has undetected defects.
- Motivation: Ensuring that the final product meets functional requirements and provides a satisfactory user experience.
- Details: User acceptance testing is critical in identifying real-world usability and functionality issues. If UAT is rushed or inadequately conducted, there is a risk that key usability issues or defects remain unresolved, leading to poor user satisfaction. Engaging a diverse group of users and allocating adequate time for feedback and fixes are vital for the project's success.

#### 6. Data Security and Privacy Risks

- Content: Risks associated with data breaches, unauthorized access, or data leaks.
- Motivation: Protecting user data is essential to comply with regulations and maintain trust.
- Details: The system handles sensitive user data, such as payment information, contact details, and booking histories. There is a risk of data breaches or unauthorized access, which could lead to reputational damage, legal consequences, and loss of user trust. To mitigate this, strict data security protocols, encryption, and regular security audits will be enforced.

#### 7. Legislative and Compliance Changes

- Content: The risk that future regulatory changes could impact the system's operations.
- Motivation: Ensuring compliance with relevant legal standards and avoiding penalties.
- Details: Regulations around data privacy (e.g., India's Data Protection Bill) and online payments may change, impacting how user data is managed and protected. Compliance with these changes will be necessary to avoid legal issues. Ongoing monitoring of relevant regulatory developments and flexibility in system design will help in adapting to these changes if they arise.

## 8. Low User Adoption and Engagement

- **Content:** The risk that the app does not achieve the desired level of user engagement or adoption.
- **Motivation:** User adoption is critical to the system's success and market viability.
- **Details:** If the application fails to attract facility managers or end-users (sports enthusiasts), the project's ROI and market position may suffer. User engagement may be low due to usability issues, inadequate marketing, or lack of perceived value. Implementing user-friendly design, strong marketing campaigns, and addressing core user needs will be key strategies to mitigate this risk.

## 9. Inadequate Quality Assurance Standards

- **Content:** The risk that inadequate QA results in a product that fails to meet quality expectations.
- **Motivation:** Maintaining high quality is essential for user satisfaction and product credibility.
- **Details:** If quality assurance processes are insufficient, the application may be released with bugs or poor performance, leading to negative user feedback. Adopting a stringent QA process with regular testing cycles and performance monitoring will ensure that the product meets quality standards and is reliable upon release.

These risks form the basis for further assessment and contingency planning within the project, ensuring a proactive approach to any potential challenges that may arise.

## 27b Risk Assessment

In assessing the risks for the Sports Facility Booking System, we evaluate each risk based on its Likelihood (probability of occurrence) and Impact (potential effect on the project if the risk materializes). This approach enables us to prioritize risk management efforts effectively. The following risk assessment provides details regarding the importance of the following risk:

1. **Inaccurate Cost Estimation:** Misjudging the project's costs could strain the budget, lead to delays, and force compromises on essential features. Accurate initial cost estimates with regular financial evaluations are crucial to mitigating this risk. If not



addressed, resource reallocation may be required, potentially impacting quality or timelines.

2. **Schedule Pressure:** Given the peak demand periods (e.g., sports seasons), the project faces considerable pressure to meet deadlines. If delays occur, they could impact the system's market readiness, leading to lost opportunities. Implementing buffer time, phased rollouts, and adjusting expectations with stakeholders are crucial in managing this risk.
3. **Dependency on Third-Party Services:** Reliance on external services such as payment processing, location services, and maps can cause severe issues if these services experience downtime or change their APIs. Backup service providers or fall-back options and robust integration testing will be necessary to mitigate potential service interruptions.
4. **Creeping User Requirements:** Scope creep can extend the timeline and increase costs if not managed effectively. Establishing a change control process and enforcing clear boundaries on feature inclusion will help ensure that only essential changes are incorporated, preserving project scope and timeline.
5. **Inadequate User Acceptance Testing (UAT):** Poor UAT could lead to undetected usability or functionality issues, resulting in user dissatisfaction post-launch. Involving a comprehensive user group for testing, and allocating ample time for issue resolution, is critical for the project's success.
6. **Data Security and Privacy Risks:** Security vulnerabilities can lead to data breaches, loss of trust, legal consequences, and financial penalties. Comprehensive security audits, encryption, data handling protocols, and compliance checks with relevant data protection laws will be essential to mitigate this risk.
7. **Legislative and Compliance Changes:** Potential future changes in data privacy or financial regulations could impact how the system handles user data and transactions. Regularly monitoring regulatory changes and building flexibility into the system will help address this risk without compromising compliance.
8. **Low User Adoption and Engagement:** Insufficient user engagement could jeopardize the app's market viability. Conducting thorough market research, creating an intuitive user experience, and using targeted marketing are strategies to ensure effective user engagement and adoption.

9. **Inadequate Quality Assurance Standards:** A lack of sufficient QA can lead to bugs, performance issues, and poor user experience. Implementing rigorous testing processes and continuous integration (CI) practices will ensure the system maintains high quality and performance standards, reducing the risk of user dissatisfaction.

**Table 11: Risk Assessment Summary**

<b>Risk</b>	<b>Likelihood</b>	<b>Impact</b>	<b>Severity Level</b>
Inaccurate Cost Estimation	Medium	High	High
Schedule Pressure	High	High	Very High
Dependency on Third-Party Services	High	High	Very High
Creeping User Requirements	High	Medium	High
Inadequate User Acceptance Testing	Medium	High	High
Data Security and Privacy Risks	Medium	Very High	Very High
Legislative and Compliance Changes	Low	High	Medium
Low User Adoption and Engagement	Medium	High	High
Inadequate Quality Assurance	Medium	High	High

This risk assessment provides a foundation for prioritizing risk management activities, with a focus on the most critical risks that could impact the project's success, budget, schedule, and user satisfaction.

## **27c Risk Projection**

This risk projection outlines the possible impact on the project and its outcomes if identified risks were to materialize. By understanding these impacts, the project team can be prepared with contingency actions to minimize disruptions.

Assessing the real-world implications of risks allows the team to prioritize resources effectively, ensure continuity, and protect the project's timelines, quality, and cost structures.

**Table 12: Risk Projection with Potential Impacts and Mitigation Strategies**

<b>Risk ID</b>	<b>Risk Description</b>	<b>Timeframe</b>	<b>Potential Impact if Realized</b>	<b>Mitigation Strategy</b>
R1	<b>Feature Creep:</b> Continuous addition of new requirements.	Throughout development	Project timelines and budget would be extended, resulting in delays and possible quality issues due to rushed changes.	Set strict scope boundaries, conduct regular scope reviews.
R2	<b>Data Privacy &amp; Compliance:</b> Risks around data handling standards (GDPR, CCPA).	Ongoing	Non-compliance fines, reputational damage, user trust issues, or app suspension by regulatory authorities.	Implement secure storage/encryption, conduct regular audits.
R3	<b>System Downtime:</b> Unexpected server downtimes.	Production	User dissatisfaction, revenue loss, negative ratings, and potential legal liabilities if frequent.	Use redundant servers, regular load testing, and quick escalation protocols.
R4	<b>User Adoption:</b> Lower-than-expected user engagement.	Post-launch	Reduced revenue, decreased market reach, and missed business goals.	Conduct targeted marketing campaigns, enhance user experience.
R5	<b>Integration Issues:</b> Challenges integrating with payment gateways or external APIs.	Integration phase	Payment failures, revenue interruptions, or user frustration, which may lead to increased churn.	Perform pre-integration testing, maintain close communication with API providers.
R6	<b>Cyber security Threats:</b> Potential for data breaches and hacks.	Ongoing	Severe reputational damage, possible lawsuits, user attrition, and compliance penalties.	Employ multi-layered security, conduct regular security assessments.
R7	<b>Inadequate Performance on Older Devices:</b> Users experience slow or unresponsive UI on older devices.	Development	Negative user feedback, increased app uninstalls, or a reputation for poor accessibility.	Optimize code, offer a lightweight app version if feasible.

R8	<b>Vendor Dependency:</b> Reliability on third-party vendors, particularly for hosting or payment services.	Throughout project	Service interruptions, affecting user experience and potentially delaying financial transactions.	Establish fall-back providers and service level agreements (SLAs).
R9	<b>Inaccurate Budget Estimations:</b> Potential cost overruns.	Entire project lifecycle	Insufficient funds for project completion, requiring cuts to functionality or quality to stay within budget.	Regular budget reviews and forecasts, set up contingency funds.
R10	<b>Legislative Changes:</b> New regulations impacting data storage or user management requirements in India.	Ongoing	Costly re-engineering, compliance adaptation, or legal liabilities if the app fails to meet new standards.	Monitor regulatory updates, build compliance adaptability.

This risk projection underscores the consequences if each risk were to occur, emphasizing the importance of proactive management to safeguard the project's success.

## 27d Risk Mitigation, Monitoring and Management (RMMM)

In managing risks for the Sports Facility Booking Management System, RMMM involves creating strategies to handle anticipated issues effectively. This process ensures proactive control of risks, continuous monitoring, and efficient handling of any emerging issues. The RMMM outlines specific actions to:

- Mitigate identified risks by reducing their likelihood or impact.
- Monitor risks to detect early signs of occurrence.
- Manage the response if a risk becomes a reality, minimizing adverse effects on the project.

Effective RMMM strategies ensure project stability, reduce disruptions, and support timely responses to any issues, ultimately safeguarding the app's quality and timely delivery. This approach enables the team to keep risks in check, maintain operational efficiency, and protect the user experience.

---

### 1. Technical Challenges in Real-Time Booking

- Mitigation: Use advanced load balancing and caching to maintain system performance.
- Monitoring: Continuously track system load and booking response times.
- Management Action: Increase server resources or capacity immediately if challenges arise.

### 2. Data Privacy and Compliance Issues

- Mitigation: Regularly audit security measures and update encryption protocols.
- Monitoring: Conduct periodic data security audits and track compliance.
- Management Action: Conduct a full system review and take immediate action if any breach occurs.

### 3. Unforeseen Infrastructure Costs

- Mitigation: Maintain a budget buffer and adopt scalable cloud solutions to manage costs.
- Monitoring: Review financials monthly against budget projections.
- Management Action: Adjust project scope or seek cost-effective solutions if costs exceed budget.

### 4. Schedule Delays Due to Feature Changes

- Mitigation: Enforce a strict change management process to control new feature introductions.
- Monitoring: Conduct weekly progress reviews and assess milestones.
- Management Action: Re-prioritize features, focusing on core functionalities if delays occur.

### 5. Low User Adoption Rates

- Mitigation: Implement an engaging marketing strategy and establish user feedback loops.
- Monitoring: Track user analytics, including downloads and active usage rates.
- Management Action: Refine the app's UX/UI and boost targeted marketing efforts if adoption rates are low.

### 6. Third-Party Integration Failures

- Mitigation: Conduct thorough testing and select reliable third-party vendors.
-

- Monitoring: Routinely test integrations and monitor vendor performance.
- Management Action: Temporarily remove or replace the integration if performance issues arise.

This approach ensures proactive steps for risk mitigation, continuous monitoring to catch early signs of issues, and a clear management response to minimize impact.

## 28 Costs

The Sports Facility Booking Management Application is designed to simplify and enhance the process of booking sports facilities. This proposal outlines the initial cost estimation required for the successful development and implementation of the application, taking into account various parameters, expected efforts, and overall project management.

### 1. Effort Calculation

- The effort required for developing the application is estimated using the COCOMO model, a well-established method for software cost estimation. The formula used for this calculation is:

$$\text{Effort} = 2.94 \times (\text{Size})^{1.12} \times \text{Product of Effort Multipliers}$$

- Estimated Size: The application is expected to be approximately 1000 function points, a unit of measurement that represents the functional requirements of the software.
- Effort Multipliers: Various factors, including team experience, project complexity, and other environmental considerations, lead to adjustments in the effort multipliers. After analysing these factors, the overall effort has been adjusted to reflect a reduction of about 30%, leading to an estimated effort of:

$$\text{Adjusted Effort} = 43.97 \times 0.7 = 30.78 \text{ person-months}$$

### 2. Development Time

- The estimated development time required for the project is calculated using the following equation:

$$T_{\text{DEV}} = 2.5 \times (\text{Effort})^{0.38}$$

- Substituting the estimated effort:

$$T_{\text{DEV}} = 2.5 \times (30.78)^{0.38} \approx 8 \text{ months}$$

- This estimation includes time allocated for analysis, design, development, testing, and deployment.

### 3. Cost per Developer

- The cost of employing developers for the project is a critical component of the overall budget. The estimated cost per developer per month is set at: ₹1,00,000
- This figure encompasses salary, benefits, and overhead costs associated with the development team.

### 4. Total Cost Calculation

- The total cost of the project is derived from the calculated effort and the cost per developer. The formula is as follows:  
$$\text{Total Cost} = \text{Effort} \times \text{Cost per Developer}$$
- Calculating the total cost:  
$$\text{Total Cost} = 30.78 \times ₹1,00,000 = ₹30,78,000$$
- This total cost represents the anticipated budget required to cover all aspects of the project, including development, project management, and any potential unforeseen expenses.

### 5. Summary of Costs

- Estimated Effort: ~31 person-months
- Estimated Development Time: ~8 months
- Total Estimated Cost: ₹30,78,000 INR
- To accommodate potential additional requirements or unforeseen circumstances, a budget of approximately ₹32,00,000 INR

This initial cost estimation outlines the resources required for the development of the Sports Facility Booking Management Application. The calculated effort, time, and costs are designed to ensure a comprehensive understanding of the financial requirements and to facilitate effective project planning. By implementing these projections, we aim to deliver a robust and user-friendly application that meets the needs of sports facility managers and users while ensuring financial accountability and resource optimization.

The proposed budget will be crucial for securing necessary funding and guiding the project through its various phases, allowing for adjustments as needed based on project developments and stakeholder feedback.

---

## 29 Project Retrospective

At the conclusion of the Sports Facility Booking Management Application project, a retrospective review will be conducted to assess the methodologies and practices utilized throughout the project lifecycle. This reflection will identify effective methods that should be replicated and ineffective ones that should be avoided in future projects. Key components of the retrospective will include:

1. Successful Practices:

- **Example: Agile Methodology:** The adoption of Agile practices allowed the team to respond to changes in user requirements quickly. For instance, during a mid-project review, feedback from users led to the addition of a new feature for real-time facility availability. This flexibility contributed to a more user-centred design and improved overall satisfaction.
- **Justification:** Agile methodology fosters adaptability and continuous feedback, essential for projects with evolving requirements, ensuring the final product aligns with user needs.

2. Ineffective Practices:

- **Example: Insufficient Communication:** Regular meetings were scheduled, but key stakeholders often could not attend due to scheduling conflicts. As a result, important updates were missed, and decisions were delayed.
- **Justification:** Effective communication is crucial for project success; establishing alternative channels for updates (e.g., email summaries, recorded meetings) can ensure all stakeholders remain informed.
- **Example: Ignoring User Feedback During Development:** While the team conducted user testing at the end of the development cycle, they missed opportunities to gather feedback throughout the process. This resulted in the late identification of critical usability issues.
- **Justification:** Incorporating user feedback iteratively ensures that the final product is aligned with user needs and enhances usability.

3. Recommendations for Improvement:

- **Example: Enhanced Training Sessions:** Based on feedback, it became clear that team members would benefit from more targeted training on the chosen development frameworks and tools.



- Justification: Providing adequate training ensures that all team members are equipped with the necessary skills to utilize tools effectively, thereby improving overall efficiency and reducing errors.
- Example: Establishing Regular Checkpoints: Implementing bi-weekly checkpoints can provide opportunities for team members to align on project progress and address any roadblocks promptly.
- Justification: Regular checkpoints can help maintain momentum and ensure that the project remains on track, addressing issues before they escalate.

The primary motivation for conducting this project retrospective is to facilitate organizational learning and promote a culture of continuous improvement. By reflecting on both the successes and challenges experienced during the project, the team can enhance performance in future projects, optimize resource allocation, and contribute to the organization's growth.

#### 1. Implementation of Practices:

- Example: While we adopted regular stand-up meetings to enhance communication, attendance fluctuated, leading to some team members missing critical updates. This inconsistency caused misalignments on project progress and priorities.
- Consideration: It is important to evaluate if the implementation of these meetings was done effectively. Could we have established clearer expectations for attendance or used alternative communication methods (like recorded updates) to ensure everyone remained informed?

#### 2. Team Dynamics and Skills:

- Example: Some team members struggled with using the project management tool effectively, leading to gaps in task tracking and accountability. This challenge was exacerbated by the tool's complexity, which overwhelmed new users.
- Consideration: Assessing whether the chosen tool was appropriate for the team's skill level is vital. Should we have provided additional training or selected a more user-friendly tool that aligns better with the team's capabilities?

### 3. User Feedback Integration:

- Example: Feedback from initial user testing was not gathered until late in the development cycle, resulting in the late identification of usability issues that required significant rework.
- Consideration: We must determine whether our approach to gathering user feedback was flawed or if we simply did not prioritize user involvement throughout the development process. Establishing regular check-ins with users for feedback could enhance future projects.

By analysing these considerations, we can better understand the complexities behind the challenges faced during the project. This reflection will guide us in refining our methodologies and practices for future projects, ultimately leading to more effective project outcomes.

---

## VII References

- [1] Li, Can & Li, Junjie & Cao, Hongxiang & Meng, Zhan. (2017). Design and Implementation of Online Booking System of University Sports Venues. MATEC Web of Conferences. 100. 02024. 10.1051/mateconf/201710002024.
- [2] Hasan, Syed. (2014). DOCUMENTATION OF ONLINE BOOKING SYSTEM. 10.13140/RG.2.1.2545.4160.
- [3] Robertson, S., & Robertson, J. (2012). Mastering the Requirements Process: Getting Requirements Right (3rd Edition). Addison-Wesley.
- [4] Eric C. Schwarz, Stacey A. Hall and Simon Shibli, Sport Facility Operations Management.