



Bangladesh Army University of Science and Technology

Saidpur, Nilphamari

Title: Community Center Booking System

Submitted By

Student Name	Student ID
1. Md Sayadul Islam	0802420405101103
2. Md Elias Ahmad	0802420405101139
3. Momita Jaman	0802420402101100

SESSIONAL PROJECT REPORT

This Report is Presented in Partial Fulfillment of the course "**CSE 2100: Software Development Project I**" in the Department of Computer Science and Engineering

Submitted To:

Shifa Tasmiah Tisha

Lecture, CSE

Department of Computer Science and Engineering

Bangladesh Army University of Science and Technology

Md. Maskur Al Shal Sabil

Lecture,CSE

Department of Computer Science and Engineering

Bangladesh Army University of Science and Technology

Submitted by

Md Sayadul Islam
0802420405101103
Dept. of CSE, BAUST

Md Elias Ahmad
0802420405101139
Dept. of CSE, BAUST

Momita Jaman
0802420402101100
Dept. of CSE, BAUST

1.1 Introduction

The BAUST Community Center Booking System is a specialized software application designed to centralize and automate the management of university facilities such as the Main Hall, Seminar Rooms, and multi-purpose rooms. It eliminates manual tracking processes, minimizes errors and scheduling conflicts (like double bookings), and provides secure, transparent access for both facility administrators and the user community (students, faculty, and staff) to view schedules and make reservation requests.

1.2 Motivation

Managing the booking of shared university resources like community centers and halls is often a key, yet time-consuming and error-prone, administrative task when handled manually. The primary motivation for developing this system is to replace the inefficient, paper-based, and human-dependent process with a fully automated, web-based platform. This automation saves administrative time, drastically reduces human errors, ensures secure and immediate access to facility schedules, and significantly improves transparency and efficiency in resource allocation across the university.

1.3 Objectives

1. Develop an online booking system for community centers.
2. Ensure real-time availability checking.
3. Provide admin panel for managing bookings and payments.
4. Provide an easy booking process for users.
5. Allow users to check availability before booking.
6. Enable admins to approve, cancel, or manage bookings.
7. Maintain a secure database of all bookings.

1.4 Project Function

The system provides distinct functionalities for different user roles:

User Role	Key Functions
User/Client	Search facility availability, view public schedule, submit a booking request.
Admin	Approve or reject booking requests, manage facility schedules, update facility details, manage user accounts.
General	Real-time facility schedule display (e.g., Main Hall, Seminar Room).

1.5 Feasibility Study

Traditional manual facility scheduling is cumbersome, prone to human error, and inefficient, especially for a large university community. A computerized Booking Management System offers automation, high accuracy, and easy, 24/7 access to reservation tools, making resource handling more reliable and efficient.

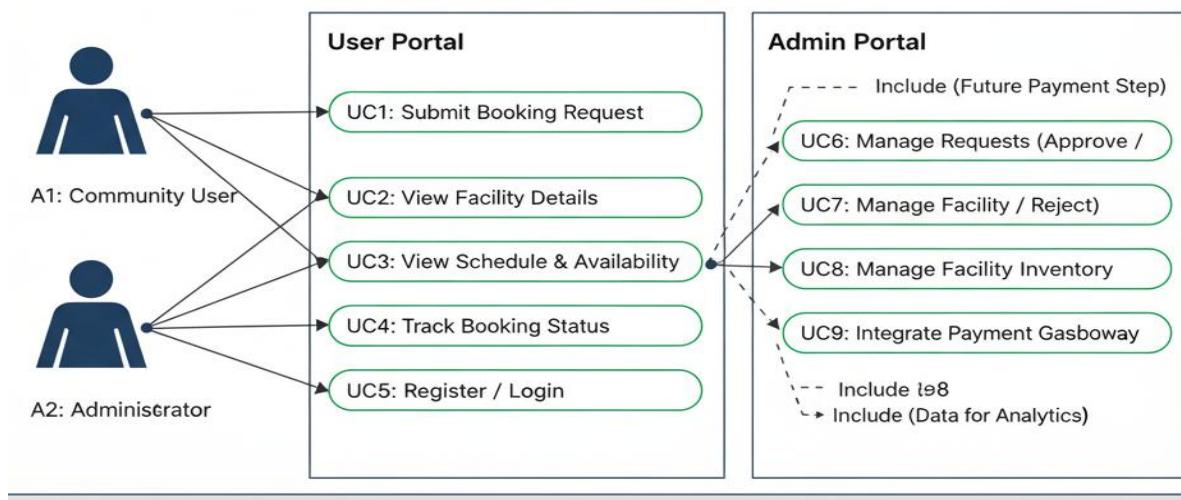
- 1 Operational Feasibility: The system is designed to be highly intuitive and easy to use for facility administrators and end-users alike, ensuring smooth adoption.
- 2 Technical Feasibility: The project utilizes established, modern web and database technologies (HTML5, Node.js, MySQL) that are readily available, well-supported, and scalable.
- 3 Economic Feasibility: The initial investment in development is quickly offset by saving administrative time, reducing paper usage, and preventing costly scheduling mistakes.

1.6 Technology Stack (Implementation Details)

The system is built using a robust, multi-tier architecture to ensure performance, scalability, and security.

Layer	Technology	Purpose
Frontend	HTML5, CSS3, JavaScript	For a responsive and interactive user interface.
Backend	Node.js with Express	For a fast, scalable, and efficient server-side application.
Database	MySQL	For reliable and secure data storage.

1.7 System Architecture



The system must support the core operations for three primary user groups: the User/Guest, the Staff/Receptionist, and the Administrator. The User/Guest functionality is centered on the reservation process. Users must be able to search facilities using multiple criteria, including location, size, cost, and critically, date/time availability and facility type (e.g., auditorium, conference room). They need to view complete facility details, including photos, ratings, and the specific cancellation policy, before proceeding to book a facility for a selected time slot. Upon booking, the system must seamlessly integrate with a Payment Gateway to securely process the booking fee and update the reservation status. Finally, users must have a dashboard to manage their bookings, allowing them to view, cancel, or modify existing reservations and download confirmation receipts.

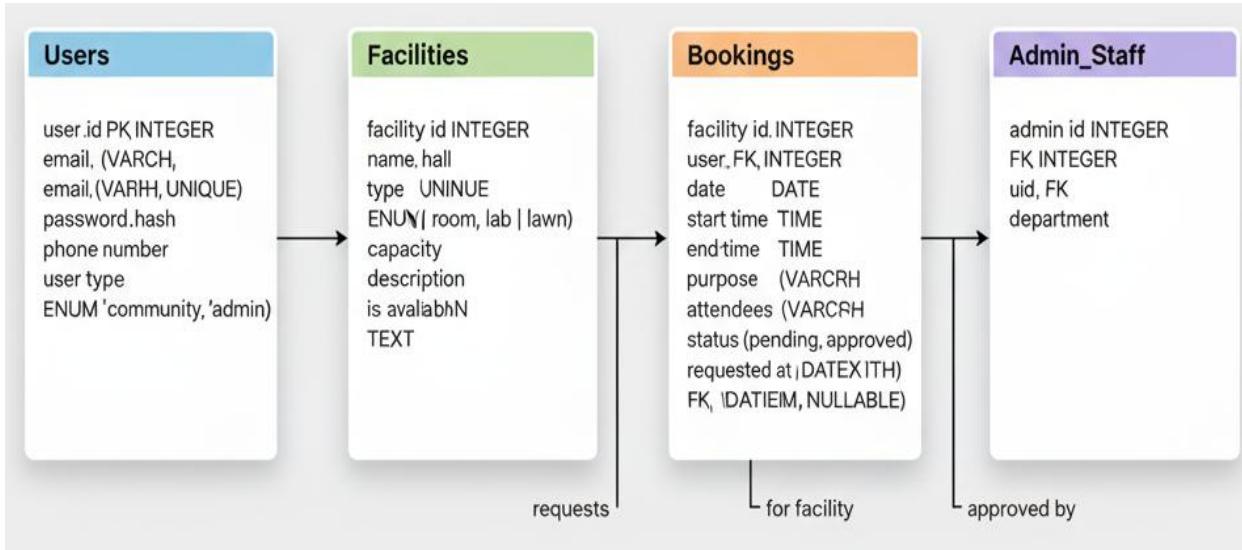
The Staff/Receptionist role is focused on day-to-day operational management and customer support. Staff must possess extensive booking management capabilities, enabling them to create bookings for walk-in guests, modify reservation details, and handle special requests. A key function is check-in/check-out tracking to monitor facility usage in real-time. They are also responsible for processing refunds and cancellations according to policy and tracking any associated financial adjustments. Crucially, the staff must have the ability to block out facilities from being booked due to necessary maintenance or repair schedules.

The Administrator holds the highest level of access and is responsible for overall system configuration and data integrity. This role includes managing all master data, which involves the creation, modification, and deletion of facility details, staff accounts, user profiles, and pricing rules. The system must support advanced facility management to define specific hourly/daily rates and set up peak/off-peak pricing logic. Furthermore, the Administrator must have access to comprehensive reporting tools to generate financial analytics, view facility utilization rates, and maintain an audit trail for all significant system transactions and user actions.

Non-Functional Requirements (NFR)

To ensure a high-quality user experience and maintain system integrity, several non-functional requirements must be met. Security is paramount: all data transmission must use SSL/TLS encryption, and the system must comply with relevant payment security standards (e.g., PCI-DSS) for handling transactions, while also being protected against common vulnerabilities like SQL injection and cross-site scripting (XSS). Performance must guarantee a responsive user experience, with critical functions like facility searches completing within two seconds, even when the system is under peak load. Usability demands an intuitive and mobile-responsive interface that allows guests to complete the entire booking process in a minimal number of steps (e.g., fewer than five). Finally, Reliability and Scalability are essential; the system must maintain an extremely high uptime (e.g., 99.9%), and the underlying architecture should be designed to handle significant future growth in both the number of facilities and transaction volume without requiring major overhauls.\

1.8 Relational Table



The system's functional scope is defined by the Use Case Diagram. This model illustrates the system's boundary, separating functionalities into a User Portal and an Admin Portal, and identifying the two key actors: the Community User (A1) and the Administrator (A2). The Community User interacts with the system to perform essential tasks such as UC5: Register/Login, UC2: View Facility Details, and UC3: View Schedule & Availability. The core user function is UC1: Submit Booking Request, and users can follow up on reservations using UC4: Track Booking Status. The Administrator, possessing a higher privilege level, manages the operational backbone of the system. This includes UC6: Manage Requests (Approving or Rejecting pending bookings), UC8: Manage Facility Inventory to update resource availability, and the management of facility details via UC7. An essential future capability outlined is UC9: Integrate Payment Gateway, indicating the plan for secure financial transaction processing. The relationships clearly depict which actor utilizes which function, with the Administrator having oversight and control over the core management use cases.

Entity-Relationship Model (Data View)

The Conceptual Database Schema forms the foundation for data persistence, organizing the system's information into four primary entities. The **Users** table acts as the central repository for all system participants, using **user.id** as the Primary Key (PK) and enforcing uniqueness on the **email** field. This table categorizes users via an **ENUM** type for access control, typically defining roles like 'community' and 'admin'. The **Facilities** table details all bookable resources, identified by its **facility id** PK, and includes descriptive attributes such as **capacity**, **type** (e.g., room, lab), and **availability status**. The crucial **Bookings** table is the transaction ledger, linking resources and users. It holds Foreign Keys (FK) referencing both the **Facilities** and **Users** tables, along with transactional data like **date**, **start time**, **end time**, **purpose**, and the current approval status (e.g.,

pending, approved). Finally, the Admin Staff table extends user details for internal personnel, linking back to the Users table via a Foreign Key to ensure proper system access and departmental assignment. This schema establishes the necessary one-to-many relationships, where one User can initiate multiple Bookings, and one Facility can be associated with multiple Bookings over time.

1.9 Facility Showcase

The BAUST Community Center Booking System is architecturally designed to support clear functional separation and data integrity. The system's scope, defined by the Use Case Model, is divided into a User Portal for functions like submitting booking requests and tracking status, and an Admin Portal for managing requests, facilities, and inventory. This logic is underpinned by a robust Database Schema centered on four core entities: Users, Facilities, Bookings (the central transactional record), and Admin Staff, ensuring data consistency and strict Role-Based Access Control between the staff and community members. Furthermore, the system prioritizes an intuitive User Experience (UX), showcased by the Available Facilities Interface, which presents campus resources like the Main Hall and Computer Lab in a clean, card-based layout, providing essential details like capacity and purpose to streamline the user's facility selection process. Collectively, these components form a cohesive system designed for efficient campus resource management.

1.10 The Problem

The current process of managing the community center facilities is hindered by inefficient manual tracking, leading to a high risk of scheduling conflicts (double bookings) and a lack of a central information hub for users to check facility availability in real-time.

1.11 The Solution

The proposed solution is a centralized, web-based platform designed to streamline the booking process, provide real-time schedules, and offer efficient administrator oversight.

1.12 Current Limitations

The project, in its current state, has the following known limitations:

- 1 Manual Approval: Bookings are not instant and require manual administrator approval, which creates a potential delay.
- 2 Web-Only Access: The system is currently only accessible via a web browser and lacks a dedicated mobile application.
- 3 No Payment Integration: The system lacks a built-in function for handling payments or deposits for paid facility use.
- 4 Basic Notifications: The current system does not feature automated email or SMS reminders for bookings or status changes.

1.13 Future Scope

Planned enhancements and future developments include:

- 1 Mobile App: Develop native Android and iOS apps for on-the-go booking and management.
- 2 Automated Notifications: Implement email and SMS reminders for upcoming bookings and status changes.
- 3 Online Payment Gateway: Integrate a secure payment system (e.g., local banking or international gateways) for paid events or facility deposits.
- 4 Analytics Dashboard: Add a reporting module for admins to track facility usage, popular time slots, and other key metrics.

1.14 Conclusion

The Community Center Booking System successfully automates the facility reservation process at BAUST, making it faster, more accurate, and secure. It significantly reduces administrative overhead and the potential for scheduling errors, while providing students and faculty with easy, transparent access to resource availability. This project improves overall efficiency in resource management and is a necessary step away from outdated manual procedures.

1.15 References

1. Welling, L., & Thomson, L. (2017). PHP and MySQL Web Development (5th ed.). Addison-Wesley Professional.
2. MySQL Documentation. (2024). MySQL Reference Manual. Oracle Corporation.
3. PHP Documentation. (2024). PHP Manual. The PHP Group.
4. W3Schools. (2024). PHP, MySQL, HTML, CSS Tutorials.
5. Robbins, J. N. (2018). Learning Web Design: HTML, CSS, JavaScript & Web Graphics (5th ed.). O'Reilly Media.
6. GeeksforGeeks. (2024). PHP, MySQL, and CSS Programming Tutorials.
7. TutorialsPoint. (2024). PHP, MySQL, and CSS Documentation & Guides.