

Final Year Design Project System Requirements Specification

E-Masjid System

---

Software Design Specification Document

by

**Dawood Ahmed**

**089264 (2022-KS-158)**

**Haris Ehsan**

**089301 (2022-KS-190)**

**(Evening)**

Project Advisor:

**Muhammad Kamran**

---

Faculty of Computing & Information Technology,  
University of the Punjab, Lahore, Pakistan.

(2025)

# **E-Masjid System**

## **Executive Summary**

We have made this design document which provide the technical design for our project E-Masjid system. It is a web based platform which help mosque to manage their daily operation digitally. This system is design in a way to solve the basic problems of manual record keeping and lack of transparency by giving features for donation tracking, prayer time management, event organization and online nikah booking service.

We are using modern web technologies including React.js for the user interface, Node.js for the server, and MongoDB for data storage. The design includes secure payment processing with Stripe, proper user authentication, and responsive design for mobile compatibility. This document serves as a complete guide for our development team to build the system correctly and efficiently.

## Table of Contents

<b><i>E-Masjid System</i></b> .....	<b>2</b>
Executive Summary .....	<b>2</b>
<b><i>System Design</i></b> .....	<b>5</b>
1. Design Considerations .....	<b>5</b>
2. Requirements Traceability Matrix .....	<b>6</b>
3. Design Models .....	<b>7</b>
3.1 Architectural Design .....	<b>7</b>
Class Diagram .....	8
Component diagram .....	9
3.2 Data Design .....	<b>9</b>
3.2.1 Data Dictionary .....	11
3.3 User Interface Design .....	<b>13</b>
3.3.1 Screen Images .....	14
3.3.2 Screen Objects and Actions .....	18
3.4 Behavioral Model .....	<b>19</b>
4. Design Decisions .....	<b>25</b>
5. Summary .....	<b>27</b>
<b><i>References</i></b> .....	<b>28</b>

## List of Tables

Table 1 Requirements Traceability Matrix .....	6
Table 2 Data Dictionary Table .....	11

## Tables of Figures

Figure 1 Class diagram .....	8
Figure 2 Component diagram .....	9
Figure 3 Home page design .....	15
Figure 4 Login page design .....	15
Figure 5 Donation Transparency page design .....	16
Figure 6 Admin Dashboard design .....	17
Figure 7 Scholar page design .....	17
Figure 8 Login Sequence Diagram .....	19
Figure 9 Online Donation Sequence Diagram .....	20

Figure 10 Nikah Booking Sequence Diagram .....	21
Figure 11 Admin Update Prayer Times Sequence Diagram .....	22
Figure 12 User Account State Diagram .....	23
Figure 13 Nikah Booking Status State Diagram.....	25
Figure 14 Event State Diagram.....	25

## System Design

### Product Perspective

Our E-Masjid System is a complete web based platform that will be accessible through any modern web browser. The system is designed to serve mosque administration for managing operations and community members can use these services.

### Dependencies

1. Stable internet connection for all users.
2. Stripe service availability for payment processing.
3. Modern web browsers supporting React.js features.
4. MongoDB database server for data storage.

### Interaction with Other Systems

1. **Stripe Payment Gateway:** It is used for making online donations securely.
2. **Email Service:** It is used for sending password reset links.
3. **Internet Connection:** It is required for all users to access the system.

### Design Constraints

1. **Performance Requirements:** Prayer times page loads within 3 seconds, handles 100+ users during Friday prayers.
2. **Usability Requirements:** Simple interface with large buttons, works on mobile and computer, elderly friendly design.
3. **Security Requirements:** Encrypted passwords, secure payments through Stripe, admin access protection.
4. **Technical Constraints:** MERN stack technology, responsive design, automatic weekly backups.

## 1. Design Considerations

### Assumptions

Following are the assumptions:

1. Mosque administrators have basic computer knowledge.
2. Users have internet access and email accounts.
3. The mosque has at least one computer for admin use.

4. Religious scholars can use basic web applications.
5. Community members can use web browsers on phones or computers.

### **Dependencies**

Following are the dependencies:

1. Stable internet connection for all users.
2. Stripe payment service available at any time.
3. Web browsers supporting modern JavaScript.
4. MongoDB database running properly.

### **Limitations**

Following are the limitations:

1. Cannot work without internet connection.
2. No SMS notifications for announcements.
3. No mobile app version.
4. Payment system requires card payments only.
5. Cannot handle offline data entry.

### **Risks**

Following are the risks:

1. Payment security issues.
2. System downtime during prayer times.
3. Elderly users finding the system difficult.
4. Data loss from system crashes.

## **2. Requirements Traceability Matrix**

Table 1 Requirements Traceability Matrix

<b>Requirement ID</b>	<b>Requirement Description</b>	<b>Design Specification</b>
FR-1	The system will allow users to register and login with email and password, with different access levels for admin and community members	Component "User Authentication Module"

FR-2	The system will show donation records, expense details, and financial reports so people can see both income and spending	Component "Financial Reporting Module"
FR-3	The mosque admin will be able to record cash donations with donor name, amount, date, and donation type	Component "Donation Management Module"
FR-4	The admin will be able to add, update, or remove events and announcements such as islamic classes, community programs, and eid prayers. Users can view them on the main page	Component "Event & Announcement Manager"
FR-5	Community members will be able to book nikah registrar for nikah ceremonies by selecting date and providing contact details	Component "Nikah Booking Module"
FR-6	The admin will be able to set and update daily prayer times including special timings for Jummah and Ramadan	Component "Prayer Times Manager"
FR-7	Community members will be able to make donations online through the website by entering amount and personal details and process real payments using Stripe and see payment confirmation with receipt	Component "Online Payment Processor"
FR-8	If users forget their password, they can reset it using their email address	Component "Password Recovery Module"
FR-9	Users can see if their Nikah service request is pending, accepted, or rejected	Component "Booking Status Tracker"
FR-10	Admin can add where mosque money is spend like for repairs, electricity etc	Component "Expense Management Module"
FR-11	Admin can create special accounts for religious scholars who perform Nikah	Component "User Account Manager"

### 3. Design Models

This section describes the various design models used to represent the E-Masjid System architecture. We use UML diagrams to visualize the system structure, data relationships, and component interactions to ensure clear understanding.

#### 3.1 Architectural Design

Our E-Masjid System follows the MVC architecture pattern. This separates the system into three main parts.

- **Model:** Handles data and business logic.
- **View:** User interface that users see.
- **Controller:** Processes user requests and connect Model and View.

## Class Diagram

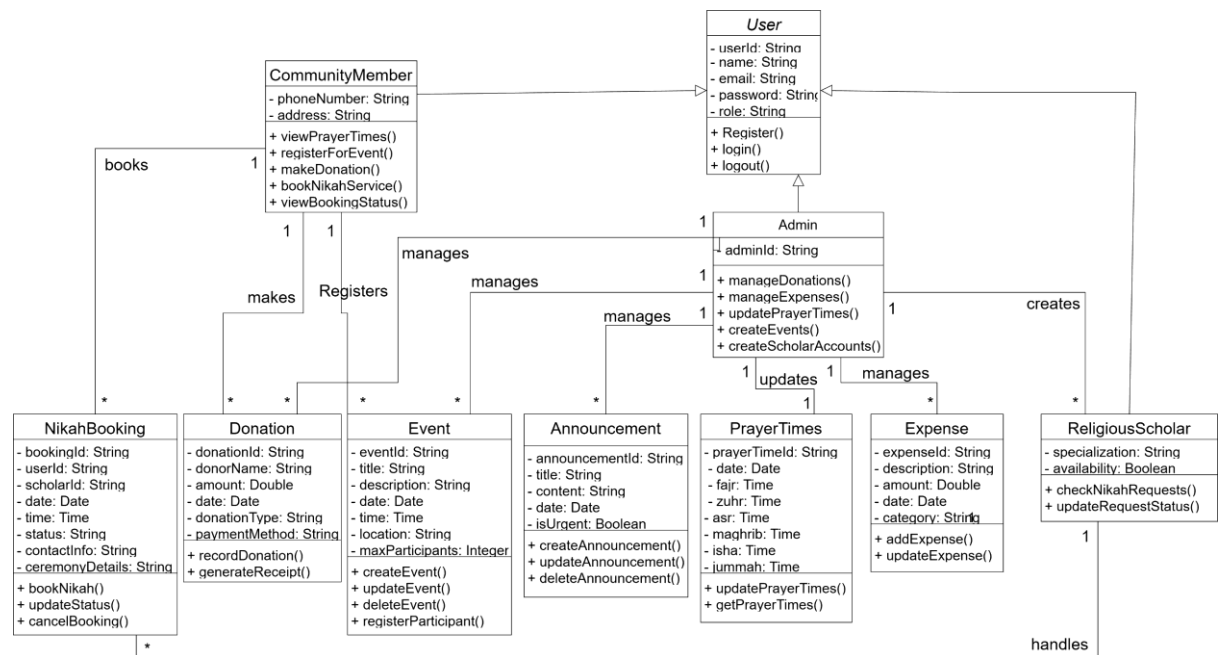


Figure 1 Class diagram

## System Architecture Components

### Frontend Layer:

- User Interface Components.
- Mobile-responsive design.
- Client side validation.
- Real-time updates.

### Backend Layer:

- API routes and controllers.
- Business logic processing.
- Authentication and authorization.
- Payment processing.

### Data Layer:

- User data and profiles.



- Donation and expense records.
- Prayer time schedules.
- Event and announcement data.
- Nikah booking requests.

#### External Services:

- Stripe Payment Gateway.
- Email Service for password resets.

## Component diagram

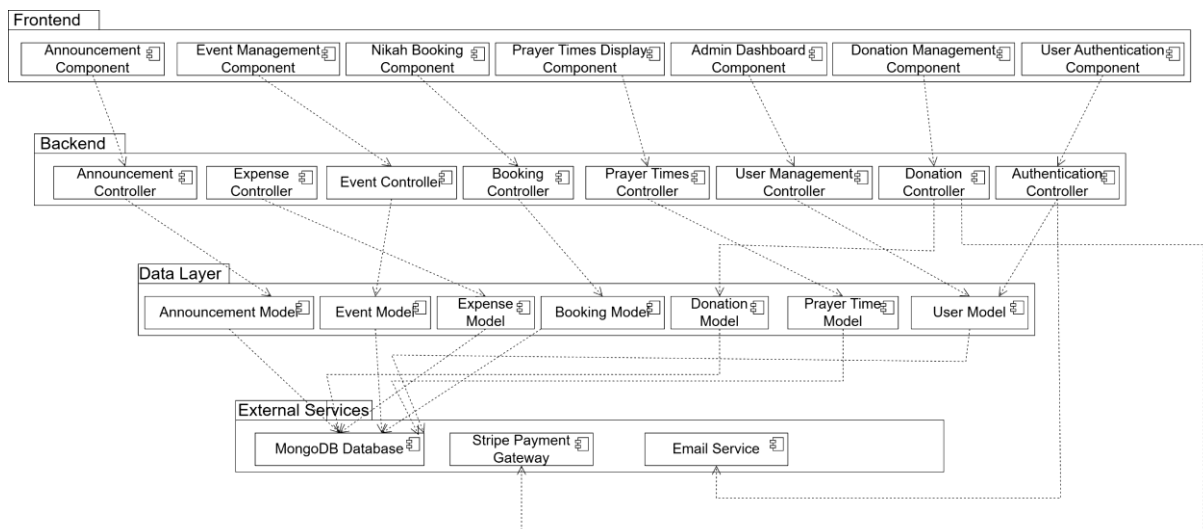


Figure 2 Component diagram

## 3.2 Data Design

The E-Masjid System transform real world mosque operations into structured data using MongoDB collections. Each main entity from the class diagram corresponds to a database collection with specific field to store and manage information efficiently.

#### Database Collections Structure:

##### 1. Users Collection

- Stores all user information including admin, community members, and religious scholars.
- Uses role based access control.
- Fields: userId, name, email, password, role, phone, address, specialization.

##### 2. Donations Collection

- Records all donation transactions.
  - Links to donor information for transparency.
  - Fields: donationId, donorId, amount, date, type, paymentMethod, receiptNumber.
3. **Expenses Collection**
- Tracks mosque expenditure for financial transparency.
  - Categorized expenses for better reporting.
  - Fields: expenseId, description, amount, date, category.
4. **Events Collection**
- Manages mosque events and programs.
  - Supports online registration.
  - Fields: eventId, title, description, date, time, location, maxParticipants, registeredUsers[ ].
5. **Announcements Collection**
- Stores important mosque announcements.
  - Supports urgent flag for important updates.
  - Fields: announcementId, title, content, date, isUrgent, publishedBy.
6. **Prayer Times Collection**
- Stores daily prayer schedules.
  - Special entries for Jummah and Ramadan.
  - Fields: prayerTimeId, date, fajr, zuhr, asr, maghrib, isha, jummah, isSpecial.
7. **Nikah Bookings Collection**
- Manages marriage service requests.
  - Tracks booking status.
  - Fields: bookingId, userId, scholarId, date, time, status, contactInfo, ceremonyDetails.

#### **Data Relationships:**

- One-to-many: One user can make multiple donations.
- One-to-many: One admin can create multiple events.
- Many-to-many: Many users can register for many events.
- One-to-one: Each day has one prayer time schedule.

### 3.2.1 Data Dictionary

Table 2 Data Dictionary Table

Terminology	Description
Users Collection	Stores all system user accounts
userId	String, Primary key for user identification
Name	String, Full name of the user
Email	String, User email address
Password	String, Encrypted password for security
Role	String, User role (admin/community/scholar)
Phone	String, Contact phone number.
Address	String, User residential address
specialization	String, For scholars area of expertise
Donations Collection	Records all financial donations
donationId	String, Unique donation identifier
donorId	String, Reference to user who donated
Amount	Number, Donation amount in rupees
Date	Date, When donation was made
Type	String, Donation type (Zakat/Sadaqah/Mosque Fund)
paymentMethod	String, Cash or Card payment
receiptNumber	String, Unique receipt number for donation confirmation
Expenses Collection	Tracks mosque spending
expenseId	String, Unique expense identifier

Description	String, What the money was spent on
Amount	Number, Expense amount in rupees
Date	Date, When expense occurred
Category	String, Expense category
Events Collection	Manages mosque events
EventId	String, Unique event identifier
Title	String, Event name/title
Description	String, Detailed event information
Date	Date, Event date
Time	Time, Event time
Location	String, Where event will be held
maxParticipants	Number, Maximum allowed attendees
registeredUsers	Array, List of user IDs who registered
Announcements Collection	Stores mosque announcements
announcementId	String, Unique announcement identifier
Title	String, Announcement headline
Content	String, Full announcement text
Date	Date, When announcement was posted
isUrgent	Boolean, Marks urgent announcements
publishedBy	String, Admin who posted the announcement
PrayerTimes Collection	Stores prayer schedules
prayerTimeId	String, Unique prayer time identifier
Date	Date, Date for prayer times

Fajr	Time, Fajr prayer time
Zuhr	Time, Zuhr prayer time
Asr	Time, Asr prayer time
Maghrib	Time, Maghrib prayer time
Isha	Time, Isha prayer time
Jummah	Time, Jummah prayer time
NikahBookings Collection	Manages marriage service requests
bookingId	String, Unique booking identifier
userId	String, Reference to user who booked
scholarId	String, Reference to assigned scholar
Date	Date, Requested ceremony date
Time	Time, Requested ceremony time
Status	String, Current status (Pending/Accepted/Rejected)
contactInfo	String, User contact details for ceremony
ceremonyDetails	String, Additional ceremony information

### 3.3 User Interface Design

The E-Masjid System will have a clean, simple, and easy to use interface designed for all types of users, including elderly people who may not be comfortable with complex technology.

#### User Experience

1. **Homepage:** Shows current prayer times, recent announcements, and quick access buttons for main features
2. **Navigation:** Simple menu at the top with clear labels
3. **Mobile Friendly:** All screens work perfectly on mobile phones and tablets
4. **Elderly Friendly:** Large buttons, clear text, and simple forms

## **How Different Users Will Use the System:**

### **For Community Members:**

1. Prayer times always visible on top of every page
2. Simple donation form with card payment option
3. One click registration for events
4. Easy booking form with date picker
5. View booking status in their profile

### **For Mosque Admin:**

1. Special admin panel accessible through /admin URL
2. Simple forms to add/update donations, expenses, events, announcements
3. Financial reports showing income and expenses
4. Create and manage religious scholar accounts

### **For Religious Scholars:**

1. View pending requests in simple list format
2. One click buttons to accept or reject bookings
3. See their booked ceremonies in calendar view

### **Feedback and Messages:**

1. Green popup messages for successful actions
2. Red popup messages with simple explanations
3. "Are you sure?" prompts for important actions
4. Spinner animation when system is processing

### **3.3.1 Screen Images**

We have created basic screen designs to show how the interface will look. These designs follow our guidelines of simplicity and ease of use.



Figure 3 Home page design

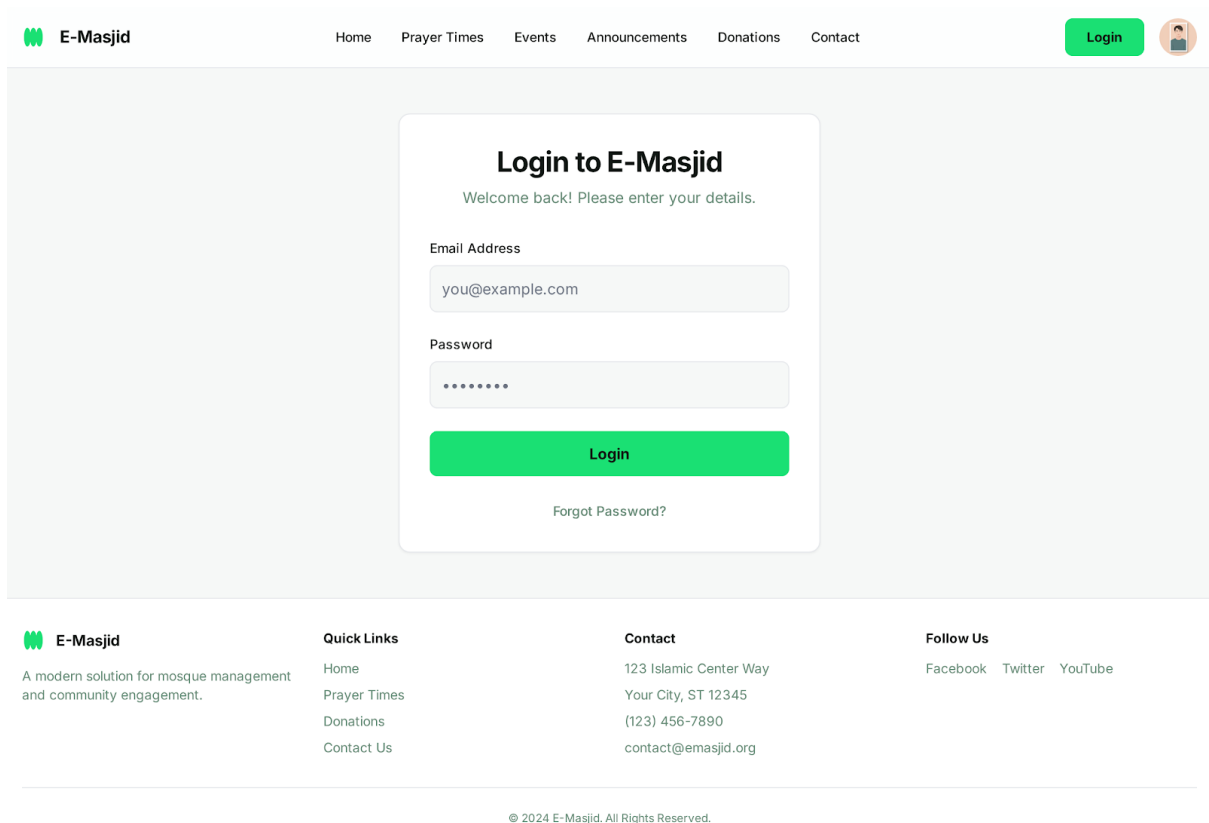


Figure 4 Login page design

## Donation Transparency Dashboard

Total Donations Received

**\$50,000**

Total Amount Spent

**\$35,000**

Remaining Balance

**\$15,000**

### Donation History

Donor Name	Amount	Date	Category
Abdullah Khan	\$250	15-Jul-2024	Zakat
Fatima Ahmed	\$100	14-Jul-2024	Sadaqah
Anonymous	\$500	12-Jul-2024	Building Fund
Yusuf Ali	\$50	11-Jul-2024	Sadaqah
Maryam Begum	\$1,000	10-Jul-2024	Zakat

### Spending History

Expense Title	Amount Used	Date	Description
Utility Bill - Electricity	\$850	13-Jul-2024	June electricity bill
Maintenance - Plumbing	\$300	09-Jul-2024	Wudu area pipe repair
Cleaning Supplies	\$150	05-Jul-2024	Monthly restock
Imam's Salary	\$4,500	01-Jul-2024	Monthly salary

Figure 5 Donation Transparency page design



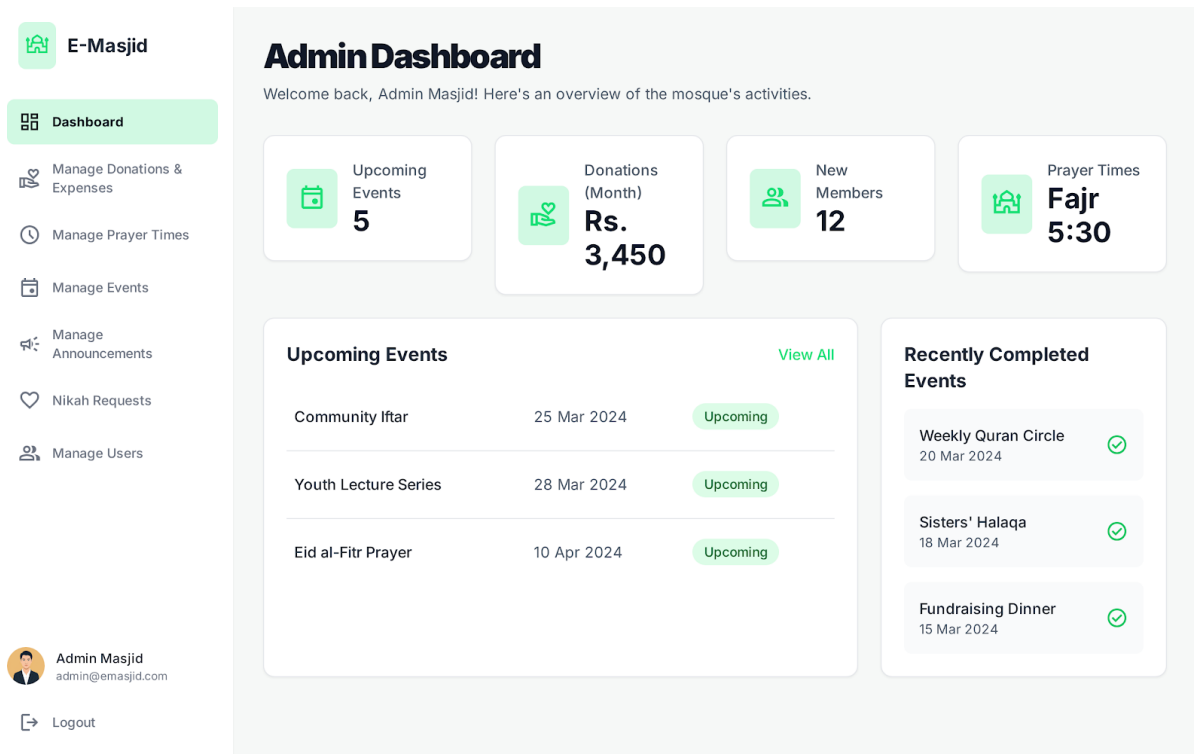


Figure 6 Admin Dashboard design

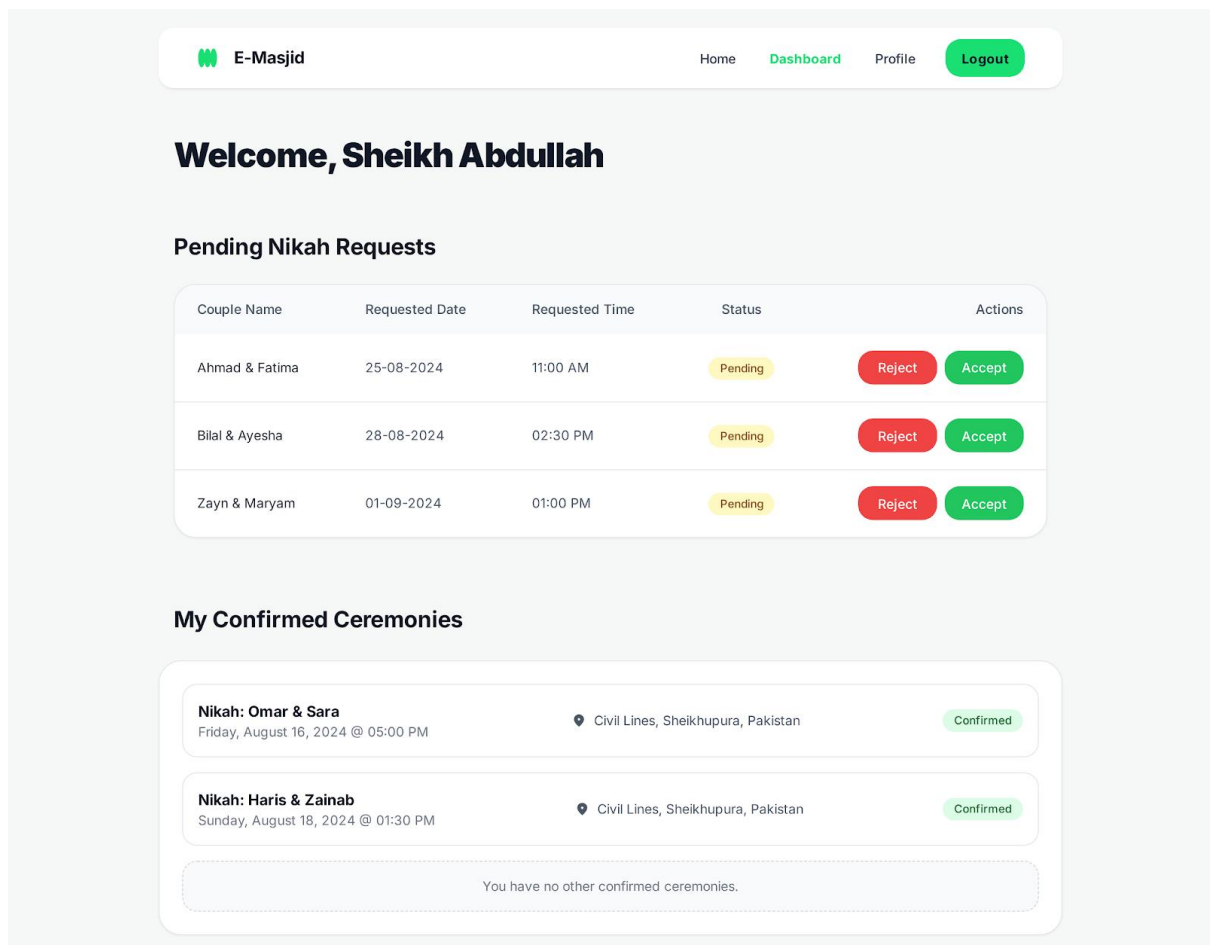


Figure 7 Scholar page design

### 3.3.2 Screen Objects and Actions

#### Use Case 1: Making an Online Donation

##### Screen Objects:

1. **Donation Amount Field:** Text box to enter donation amount
2. **Donation Type Dropdown:** Select Zakat, Sadaqah, or Mosque Fund
3. **Card Details Form:** Fields for card number, expiry, CVC
4. **Donate Now Button:** Green button to submit donation
5. **Cancel Button:** Red button to cancel the process

##### Actions:

1. **User enters amount:** System validates it's a positive number
2. **User selects donation type:** System shows description of that type
3. **User enters card details:** System validates card format
4. **User clicks Donate Now:** System processes payment via Stripe
5. **Payment successful:** Shows green "Donation Successful" message with receipt
6. **Payment failed:** Shows red "Payment Failed" message with retry option

#### Use Case 2: Admin Creating an Event

##### Screen Objects:

1. **Event Title Field:** Text box for event name
2. **Date and Time Pickers:** Calendar and time selectors
3. **Description Box:** Large text area for event details
4. **Location Field:** Text box for event location
5. **Max Participants Field:** Number field for attendance limit
6. **Publish Button:** Blue button to publish event
7. **Save Draft Button:** Gray button to save for later

##### Actions:

1. **Admin enters event details:** System validates all required fields
2. **Admin sets date/time:** System checks date is not in past
3. **Admin sets max participants:** System validates positive number
4. **Admin clicks Publish:** System creates event and shows on website
5. **Event published:** Shows green "Event Published Successfully"

6. **Validation error:** Shows red message next to incorrect field

### 3.4 Behavioral Model

Behavioral models show how the system components interact and how states change during different processes.

#### User Login Sequence Diagram

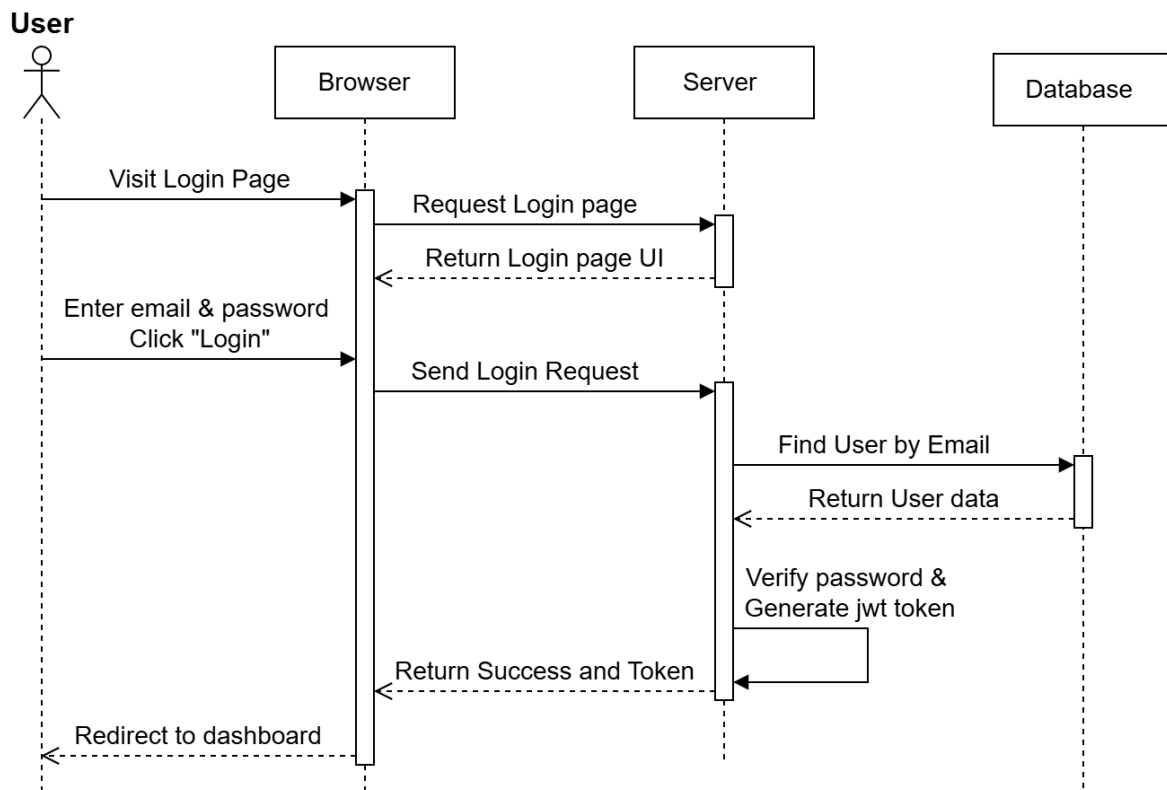


Figure 8 Login Sequence Diagram

This diagram shows the steps that happen when a user logs into the E-Masjid system. It explains how the system checks the user's details to ensure security.

#### Key Steps:

1. User enters email and password, clicks "Login"
2. Browser sends details to server
3. Server checks database for the email
4. Database finds user and sends info back
5. Server checks if password is correct
6. If correct, server creates security token
7. Token sent to browser, user taken to dashboard
8. If wrong, error message shown to use

## Online Donation Sequence Diagram

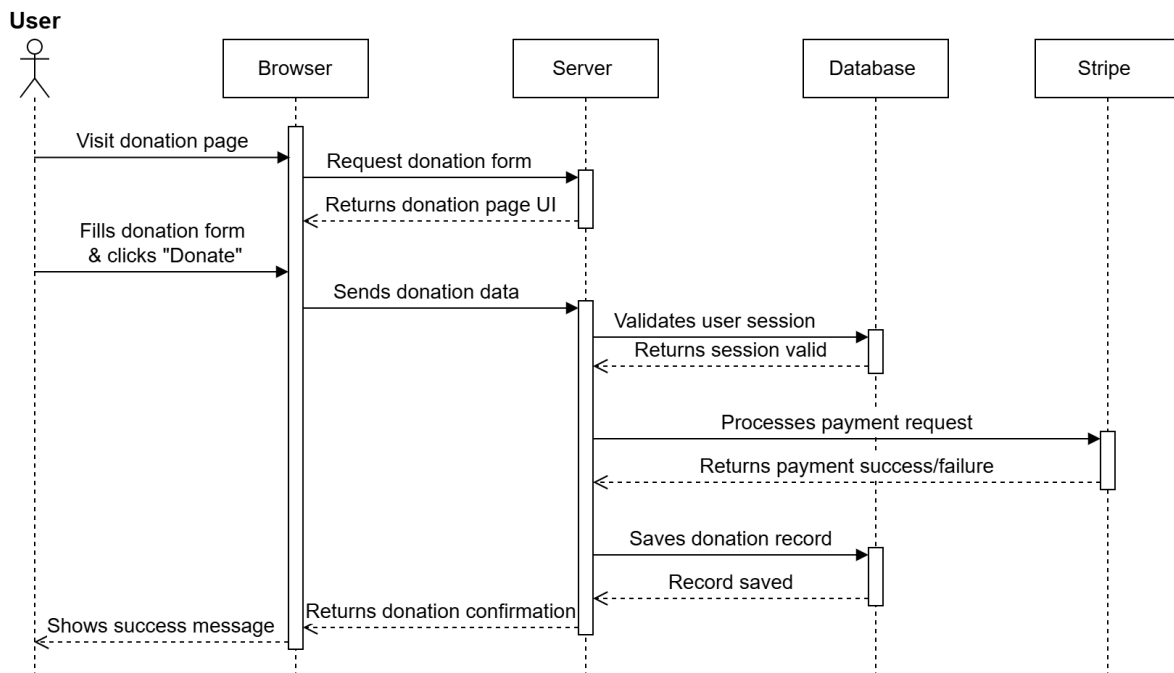


Figure 9 Online Donation Sequence Diagram

This diagram shows the complete flow when a community member makes an online donation. The process involves the user interface, backend server, database, and Stripe payment gateway working together to process the payment and record the transaction.

### Key Steps:

1. User submits donation form
2. Backend validates and processes payment via Stripe
3. Payment result recorded in database
4. Confirmation displayed to user

## Nikah Booking Sequence Diagram

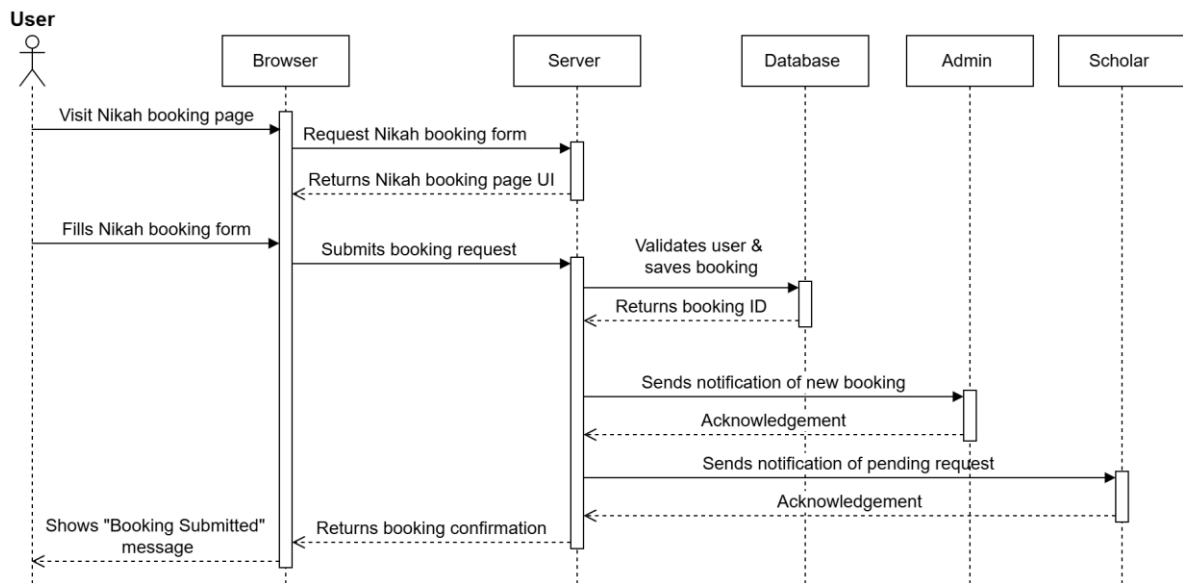


Figure 10 Nikah Booking Sequence Diagram

This diagram display all the steps of the process when a user books Nikah services. The booking request goes through validation, storage, and notification to both admin and religious scholar.

#### Key Steps:

1. User submits booking request
2. System validates and stores the request
3. Notifications sent to admin and scholar
4. User receives booking confirmation
5. Scholar can later update booking status

#### Prayer Times Update Sequence Diagram

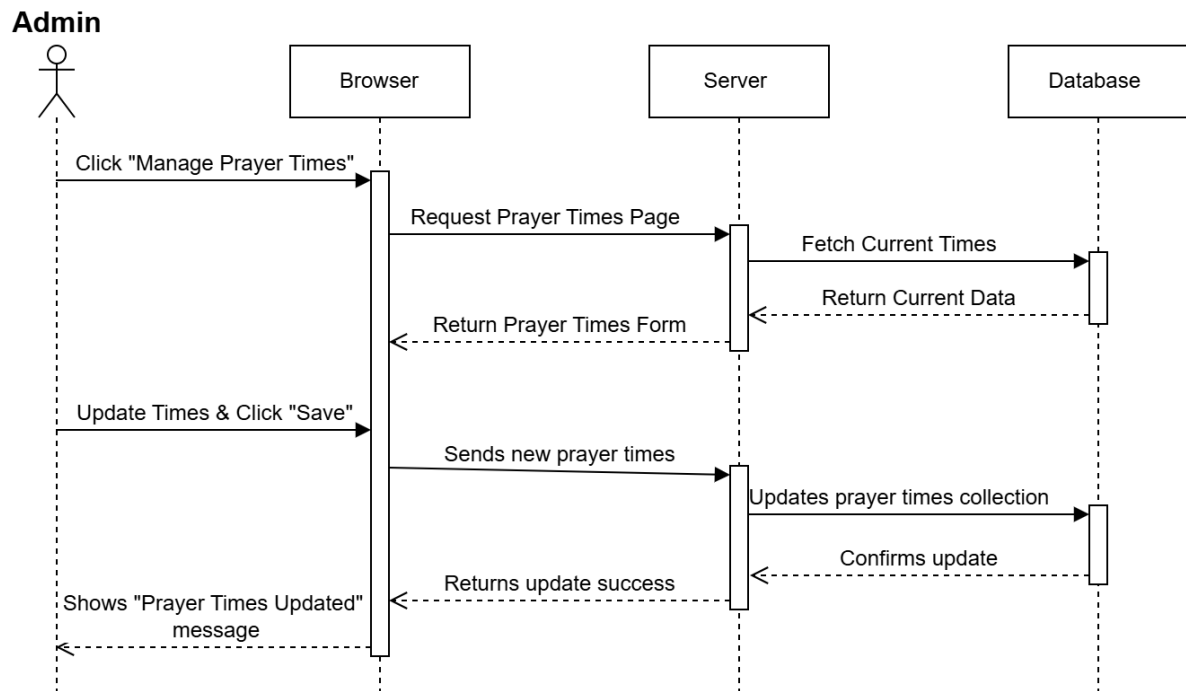


Figure 11 Admin Update Prayer Times Sequence Diagram

This diagram shows how mosque admin updates prayer times.

### Key Steps:

1. Admin updates times
2. Changes saved to database
3. Admin receives confirmation

### State Diagrams:

The following state diagrams display the dynamic behavior of key system entities. They show how entities transition between states in response to user actions and system events, ensuring proper workflow implementation.

## User Account State Diagram

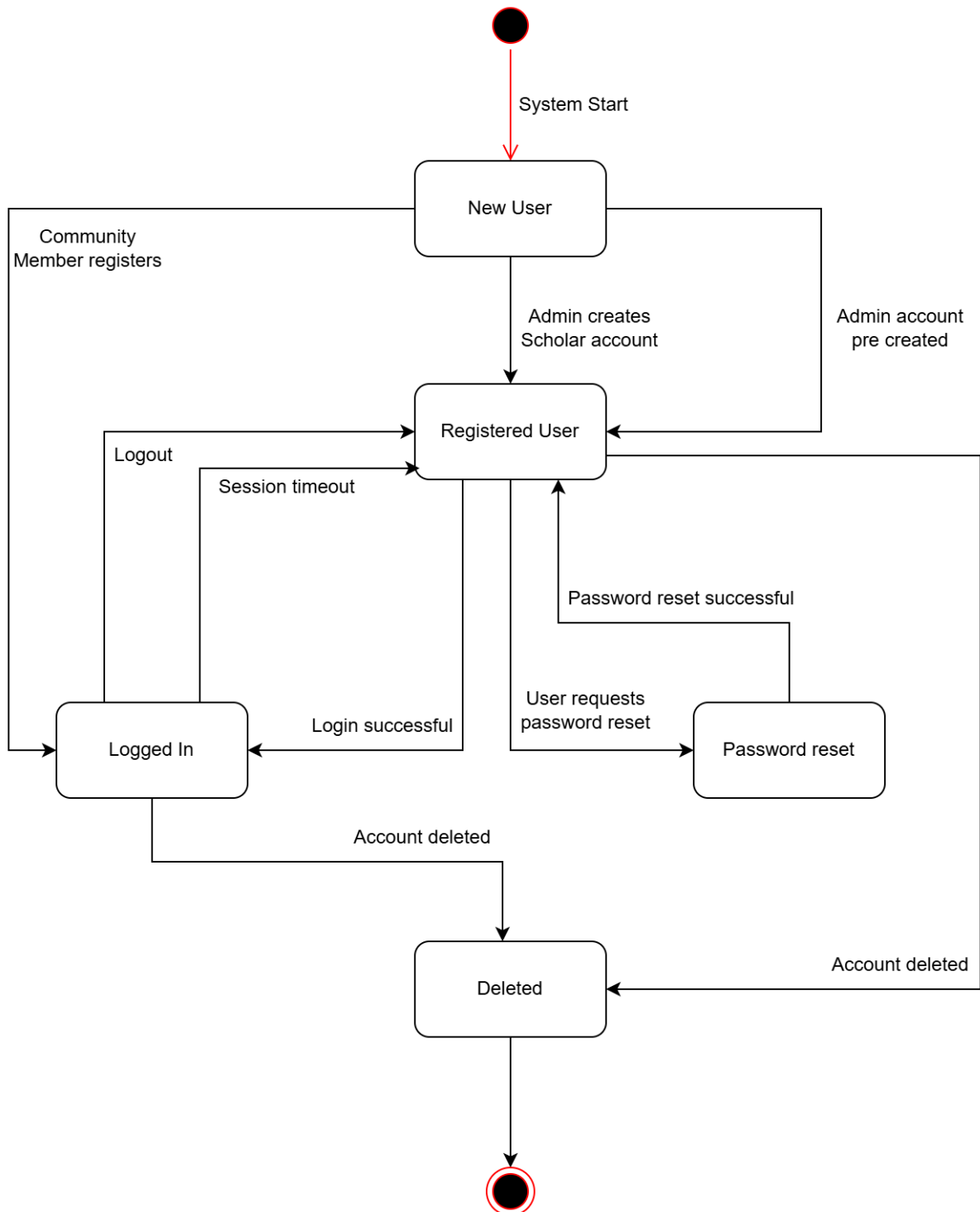


Figure 12 User Account State Diagram

Nikah Booking Status State Diagram

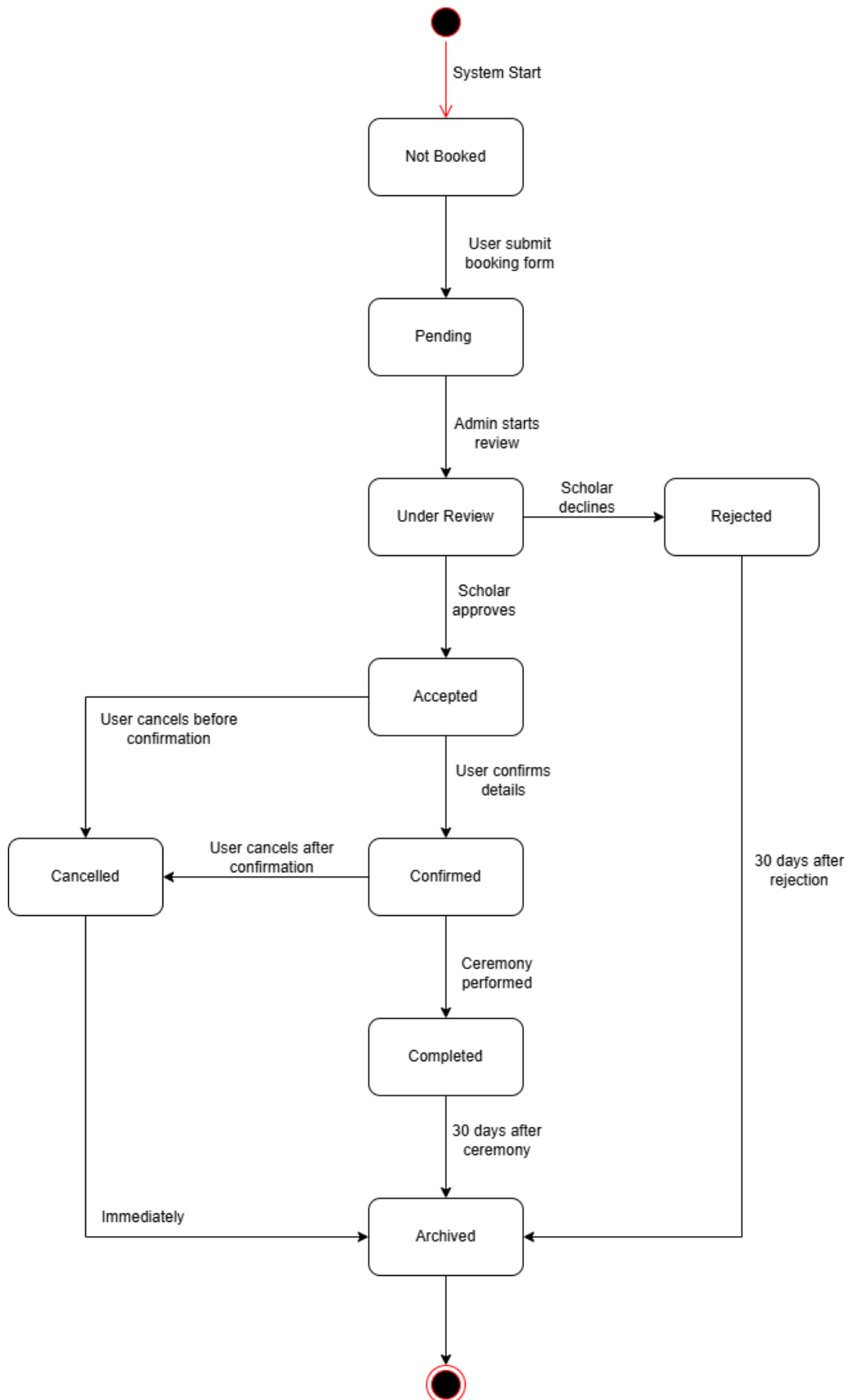




Figure 13 Nikah Booking Status State Diagram

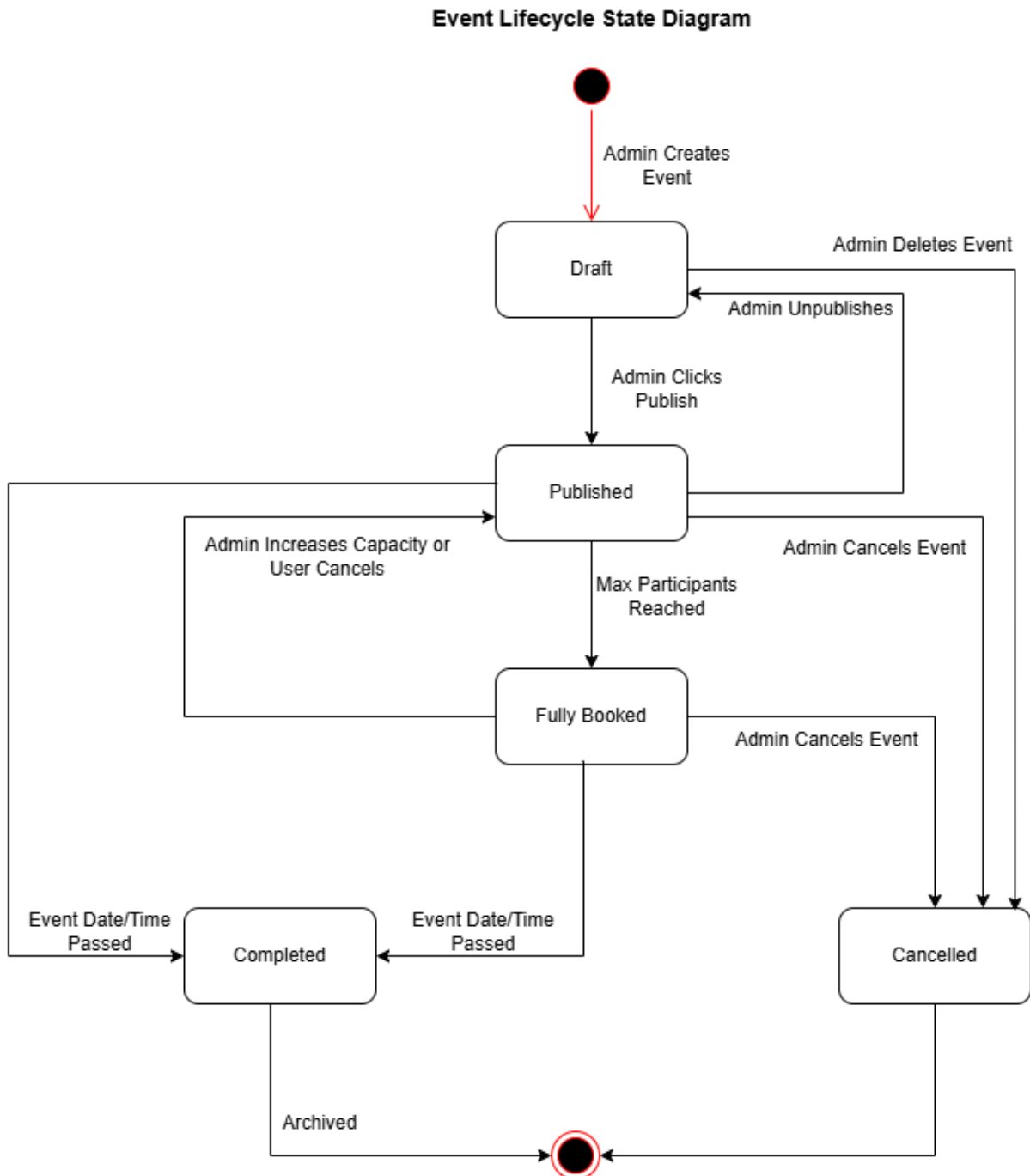


Figure 14 Event State Diagram

#### 4. Design Decisions

The following section will outline the key aspects of architecture and technology, which we have been using in the design of E-Masjid System and why we have done so.

### **Architecture Pattern Selection**

The MVC architecture was chosen for this web application. This pattern clearly separates business logic, user interface, and request handling. It makes the system easy to maintain and testable.

### **Technology Stack MERN**

The project was developed as MERN stack. The choice has a front and back end using a single JavaScript technology. The React architecture of components in the form of reusable UI components is supported however the node.js offers non-blocking I/O to handle the concurrent user requests during the most hectic time of prayers.

### **Database System**

MongoDB is a NoSQL document database that was chosen in lieu of the conventional SQL databases because of its flexibility of semi structured data. and native support with JavaScript/Node.js. It is non-schematic and dynamic data requirements, and horizontal. scaling capabilities ensure the performance during times of heavy traffic like the Friday prayers.

### **Pay Gateway Integration**

The choice of the payment processing solution was Stripe due to the high level of its security, the vast number of documents on the developers, and adherence to the PCI DSS. It is an external service that is concerned with sensitive payment information that does not require the system to bear the security-related cost and provides a secure transaction processing.

### **Authorization and Authentication**

Authentication of the user was performed through the JSON Web Tokens (JWT) that provides stateless session management, which is scaled to a distributed system. Permission of three groups of users that involved the community members, religious scholars and administrators, was controlled by the use of the role based access control (RBAC). one who has the right levels of access.

### **Responsive Design Approach**

CSS Grid and Flexbox layouts helped to take the mobile-first approach of responsive design. It will also be able to give the optimum viewing experience irrespective of the size of the device, conforming to the usability requirement of the outdated interfaces with huge buttons and simple navigation.

## **Deployment Strategy**

The system will be deployed on cloud platform (Heroku or AWS) to enjoy automatic scaling, managed services and high availability. This will lower the infrastructure maintenance and give a stable operation whenever there is an expected peak traffic.

## **5. Summary**

This Software Design Specification show the complete technical plan for our E-Masjid System. We have explained the MVC architecture, MERN technology, database design, and user interfaces. The design describes all functional requirements and provides clear guidance for development. Following this document we will ensure to build a reliable system that meets the mosque's needs.

## References

List any documents or other resources to which this SRS refers, if any. These might include user interface style guides, standards, system requirements specifications, interface specifications, or the SRS for a related product. The following are a few examples of different resources.

### **Book**

Author(s). Book title. Location: Publishing company, year, pp.

Example:

W.K. Chen. Linear Networks and Systems. Belmont, CA: Wadsworth, 1993, pp. 123-35.

### **Article in a Journal**

Author(s). "Article title". Journal title, vol., pp, date.

Example:

G. Pevere. "Infrared Nation." The International Journal of Infrared Design, vol. 33, pp. 56-99, Jan. 1979.

### **Articles from Conference Proceedings (published)**

Author(s). "Article title." Conference proceedings, year, pp.

Example:

D.B. Payne and H.G. Gunhold. "Digital sundials and broadband technology," in Proc. IOOC-ECOC, 1986, pp. 557-998.

### **World Wide Web**

Author(s)\*. "Title." Internet: complete URL, date updated\* [date accessed].

M. Duncan. "Engineering Concepts on Ice. Internet: [www.iceengg.edu/staff.html](http://www.iceengg.edu/staff.html), Oct. 25, 2000 [Nov. 29, 2003].