

**Final Year Design Project System Requirement Specification**

**E-Masjid System**

**Final Year Design Project SRS**

**Project ID:22-KS-BSIT-15**

**By**

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

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

**(Evening)**

**Project Advisor:**

**Muhammad Kamran**

**Govt. Graduate College, Civil Lines, Sheikhupura**

**University of the Punjab Lahore, Pakistan**

**(2025)**

## **E-Masjid System**

### **Executive Summary**

The E-Masjid System is a web based platform that helps mosques manage their daily operations in a digital way. It solves the problem of manual record keeping and lack of transparency in mosque management. The system allows mosque administration to manage donations with full transparency, display prayer times, organize events, and provide online nikah service booking. Community members can donate and see where their donations are used, check prayer schedules, register for events, and book nikah registrar for marriage ceremonies. We are building this system using MERN stack technology to create a responsive website that works on both mobile and computer. This system will make mosque management more efficient and build trust between mosque committees and community people.

## Table of Contents

<i>Executive Summary</i> .....	<b>2</b>
<i>Requirements Analysis</i> .....	<b>5</b>
User classes and characteristics .....	<b>5</b>
Requirement Identifying Technique .....	<b>5</b>
<b>Functional Requirements</b> .....	<b>6</b>
Functional Requirement X.....	6
<b>Non-Functional Requirements</b> .....	<b>9</b>
Reliability.....	9
Usability .....	9
Performance .....	10
Security .....	10
<b>External Interface Requirements</b> .....	<b>10</b>
User Interfaces Requirements .....	10
Software interfaces .....	11
Hardware interfaces.....	11
Communications interfaces.....	11
<i>Use case Analysis</i> .....	<b>12</b>
<b>Use Case #1</b> .....	<b>12</b>
<b>Use Case #2</b> .....	<b>12</b>
<b>Use Case #3</b> .....	<b>13</b>
<b>Use Case #4</b> .....	<b>14</b>
<b>Use Case #5</b> .....	<b>14</b>
<b>Use Case #6</b> .....	<b>15</b>
<b>Use Case #7</b> .....	<b>15</b>
<b>Use Case #8</b> .....	<b>16</b>
<b>Use Case Diagram</b> .....	<b>16</b>
<i>Storyboards</i> .....	<b>18</b>
<b>Summary</b> .....	<b>21</b>
<i>References</i> .....	<b>22</b>

## List of Tables

Table 1 Function Requirements Specification for Requirement Search Courses..... **Error! Bookmark not defined.**

Table 2 Use case Description for the Use case Recommend Course.... **Error! Bookmark not defined.**

# Requirements Analysis

This section explains the detailed requirements for the E-Masjid System. The main purpose of this SRS is to explain what our system will do and who will use it and what functions it will perform. We checked what the system needs to do and identify all the users who will interact with it. This analysis helps us to understand that what features to build and how they should work for different types of users.

## User classes and characteristics

User Class	User Characteristics
Mosque Administration	People who manage mosque operations like imam and committee members. They need full control over system and ability to manage all activities.
Donor	People who give donations to the mosque. They can see donation records, reports, and transparency details. They may donate online or in person.
Community Members	Local people who visit mosque regularly. They need to see prayer times, announcements, events and request services. They have limited access.
Religious Scholars	Islamic scholars who perform nikah ceremonies. They need to manage their availability and see their booking schedule.

## Requirement Identifying Technique

We are using Use Case Analysis technique because our system is a interactive web application with different user roles. This helps us understand how each user will interact with the system and what features they need.

### Use Cases we will analyze:

- UC1: Manage Donations
- UC2: Manage Prayer Times
- UC3: Manage Events
- UC4: Book Nikah Services
- UC5: Manage Announcements
- UC6: User Authentication

## Functional Requirements

We identified 6 main features for our system. Each feature has specific functional requirements that describe what the system should do. These requirements are written from user perspective to clearly explain the expected behavior.

### Functional Requirement X

<b>Identifier</b>	FR-1
<b>Title</b>	Record Donations
<b>Requirement</b>	The mosque admin will be able to record cash donations with donor name, amount, date, and donation type.
<b>Source</b>	Mosque committee discussion
<b>Rationale</b>	To maintain proper records and show transparency
<b>Business Rule</b>	Each donation must have at least donor name and amount
<b>Dependencies</b>	User authentication
<b>Priority</b>	High

<b>Identifier</b>	FR-2
<b>Title</b>	Show Donation Reports
<b>Requirement</b>	The system will show donation records and expenditure reports to community members in a transparent way.
<b>Source</b>	Community trust needs
<b>Rationale</b>	People want to see where their money is spent.
<b>Business Rule</b>	Reports should show income vs expenses clearly.
<b>Dependencies</b>	FR-1
<b>Priority</b>	High

<b>Identifier</b>	FR-3
<b>Title</b>	Manage Prayer Times
<b>Requirement</b>	The admin will be able to set and update daily prayer times including special timings for Jummah and Ramadan.
<b>Source</b>	Community feedback
<b>Rationale</b>	People need accurate prayer schedules.
<b>Business Rule</b>	Prayer times must be visible without login.
<b>Dependencies</b>	None
<b>Priority</b>	High

<b>Identifier</b>	FR-4
<b>Title</b>	Event & Announcement Management
<b>Requirement</b>	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.
<b>Source</b>	Community engagement needs
<b>Rationale</b>	Helps mosque communicate better with community
<b>Business Rule</b>	Events should show date, time and location clearly
<b>Dependencies</b>	User authentication
<b>Priority</b>	Medium

<b>Identifier</b>	FR-5
<b>Title</b>	Book Nikah Services
<b>Requirement</b>	Community members will be able to book nikah registrar for nikah ceremonies by selecting date and providing contact details
<b>Source</b>	Community service needs
<b>Rationale</b>	People need easy way to arrange marriage ceremonies
<b>Business Rule</b>	Booking requests must include confirm date and contact information
<b>Dependencies</b>	User authentication
<b>Priority</b>	Medium

<b>Identifier</b>	FR-6
<b>Title</b>	User Registration and Login
<b>Requirement</b>	The system will allow users to register and login with email and password, with different access levels for admin and community members.
<b>Source</b>	System security needs
<b>Rationale</b>	To protect sensitive information and manage permissions
<b>Business Rule</b>	Admin users have full access, community users have limited access
<b>Dependencies</b>	None
<b>Priority</b>	High

<b>Identifier</b>	FR-7
<b>Title</b>	Online Donation System
<b>Requirement</b>	Community members will be able to make donations online through the website by entering amount and personal details and see donation confirmation without real payment integration.
<b>Source</b>	Community needs an easier way to donate
<b>Rationale</b>	People want to donate easily without visiting mosque
<b>Business Rule</b>	Each online donation must record donor information and amount
<b>Dependencies</b>	FR-6
<b>Priority</b>	Medium

## Non-Functional Requirements

This section describes the quality requirements of our system that how it should perform and how easy it should be to use and how secure it should be.

### Reliability

The system should work reliably for daily mosque operations. It should not crash frequently and should recover quickly if any problems occur.

### Specific Requirements:

- The system should be available 95% of the time during prayer hours
- If system goes down, it should recover within 30 minutes
- Donation data should not be lost even if system has problems
- Backup of important data should happen automatically every week

### Usability

The system should be simple and easy for mosque admins and normal users to use. All buttons and forms will be clear and labeled properly.

### **Specific Requirements:**

- New users should be able to find prayer times within 2 clicks
- Donation recording process should take less than 3 minutes for admin
- Nikah booking form should be completable within 5 minutes
- Interface should use large fonts and clear buttons for elderly users
- All main features should be accessible from home page

### **Performance**

The system should work quickly and smoothly even when multiple people use it at the same time.

### **Specific Requirements:**

- Prayer times page should load within 3 seconds
- Donation reports should generate within 5 seconds
- System should handle up to 100 users at the same during Friday prayers
- Event registration should process within 2 seconds

### **Security**

The system should protect sensitive information like donor details and maintain privacy.

### **Specific Requirements:**

- User passwords should be stored encrypted in database
- Only admin should see personal donor information
- System should prevent unauthorized access to admin features
- Session should timeout after 30 minutes of inactivity

## **External Interface Requirements**

This section describes how our E-Masjid system will interact with users and other systems. It covers the user interface design, software connections, and communication methods.

### **User Interfaces Requirements**

Our system will have a clean and simple interface that works well for both mosque administrators and community members, including elderly users.

### **Design Guidelines:**

- Use simple colors that are common in Islamic design
- Large buttons and text for easy reading, especially for older users
- Consistent navigation menu on all pages
- Prayer times always visible on the header
- Mobile friendly design that works on smartphones and tablets
- Use common icons that people can easily understand
- Error messages in simple language, not technical terms

### **Layout Standards:**

- Homepage shows prayer times, announcements, and quick access to main features
- Admin dashboard with clear sections for donations, events, and services
- Forms should be simple with clear labels and instructions
- Use responsive design that adjusts to different screen sizes

## **Software interfaces**

The system will use the following software tools.

### **Frontend:**

- React.js web application running in modern browsers
- Compatible with mobile browsers on iOS and Android

### **Backend:**

- Node.js server with Express.js framework
- MongoDB database for storing all data
- JWT tokens for user authentication

### **External Services:**

- No real payment integration is implemented at this stage to keep the system simple.
- No complex API integrations to keep it simple

## **Hardware interfaces**

The system will run on any normal computer or smartphone that has an internet connection and a browser. No special hardware is required, but a basic server will host the system.

## **Communications interfaces**

Our system will use standard web communication:

### **Network Requirements:**

- Standard HTTP/HTTPS protocols for web access
- Internet connection required for using the system
- No special network configuration needed

#### **Communication Features:**

- Basic in-app notifications for new announcements
- No SMS integration initially
- No email marketing system
- Simple contact forms for communication

## **Use case Analysis**

### **Use Case #1**

<b>UC Identifier</b>	UC1
<b>Use Case Name</b>	Manage Donations
<b>Requirements Traceability</b>	FR-1, FR-2
<b>Purpose</b>	To allow admin to add, edit, and view all donations with clear records and receipts and maintain transparent records for community viewing
<b>Priority</b>	High
<b>Preconditions</b>	User must be logged in as admin,
<b>Post conditions</b>	Donation recorded in database
<b>Actors</b>	Mosque Admin, Community Member
<b>Extends</b>	None
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Admin logs into system</li> <li>2. Admin opens donation dashboard.</li> <li>3. Adds donor name, type of donation and amount.</li> <li>4. System saves data and shows success message.</li> <li>5. User can later see total donation and where it was used.</li> </ol>
<b>Alternate Flows</b>	If connection fails, system shows “Unable to save donation” message.
<b>Exceptions</b>	Invalid input like empty fields or wrong amount.
<b>Includes</b>	User authentication

### **Use Case #2**

<b>UC Identifier</b>	UC2
<b>Use Case Name</b>	Manage Prayer Times
<b>Requirements Traceability</b>	FR-3
<b>Purpose</b>	To set and display daily prayer times and special schedules for Jummah and Ramadan.

<b>Priority</b>	High
<b>Preconditions</b>	None
<b>Post conditions</b>	Updated prayer time table is shown to all users.
<b>Actors</b>	Mosque Admin, Community Member
<b>Extends</b>	None
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Admin logs into system</li> <li>2. Goes to Prayer Times section</li> <li>3. Updates prayer times for current day</li> <li>4. Marks special timings for Jummah/Ramadan</li> <li>5. Clicks Save</li> <li>6. System updates prayer display</li> <li>7. Community members view times without login</li> </ol>
<b>Alternate Flows</b>	Admin can set weekly schedule instead of daily
<b>Exceptions</b>	Missing data from admin input.
<b>Includes</b>	None

### Use Case #3

<b>UC Identifier</b>	UC3
<b>Use Case Name</b>	Book Nikah Services
<b>Requirements Traceability</b>	FR-5
<b>Purpose</b>	To allow community members to book nikah registrar for marriage ceremonies
<b>Priority</b>	Medium
<b>Preconditions</b>	User must be registered and logged in
<b>Post conditions</b>	Booking request submitted, admin notified
<b>Actors</b>	Community Member, Mosque Admin
<b>Extends</b>	None
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. User logs into system</li> <li>2. Clicks "Book Nikah Service"</li> <li>3. Selects preferred date and time</li> <li>4. Fills contact details and ceremony details</li> <li>5. Submits the request</li> <li>6. System sends confirmation to user</li> <li>7. Admin sees new booking in dashboard</li> </ol>
<b>Alternate Flows</b>	User can view their booking status
<b>Exceptions</b>	Missing date or invalid contact info.
<b>Includes</b>	User authentication

## Use Case #4

<b>UC Identifier</b>	UC4
<b>Use Case Name</b>	Manage Events
<b>Requirements Traceability</b>	FR-4
<b>Purpose</b>	To create and manage mosque events with online registration
<b>Priority</b>	Medium
<b>Preconditions</b>	Admin must be logged in
<b>Post conditions</b>	Event created and visible to community, registrations open
<b>Actors</b>	Mosque Admin, Community Member
<b>Extends</b>	None
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Admin logs into system</li> <li>2. Click Create Event</li> <li>3. Fills event title, date, time, description</li> <li>4. Set registration options</li> <li>5. Publishes event</li> <li>6. Community members see event on homepage</li> <li>7. Members register for event with their details</li> </ol>
<b>Alternate Flows</b>	Admin can cancel event or view registration list
<b>Exceptions</b>	If event date is past, show "Event completed" status
<b>Includes</b>	User authentication

## Use Case #5

<b>UC Identifier</b>	UC5
<b>Use Case Name</b>	Manage Announcements
<b>Requirements Traceability</b>	FR-4
<b>Purpose</b>	To post and manage mosque announcements.
<b>Priority</b>	Medium
<b>Preconditions</b>	Admin must be logged in
<b>Post conditions</b>	Announcement shown on website.
<b>Actors</b>	Mosque Admin, Community Member
<b>Extends</b>	None
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Admin logs into system</li> <li>2. Admin write new announcement.</li> <li>3. System post it on the website.</li> <li>4. User can view it anytime.</li> </ol>
<b>Alternate Flows</b>	Admin edits or deletes announcement.
<b>Exceptions</b>	Internet issue while saving.
<b>Includes</b>	User authentication

## Use Case #6

<b>UC Identifier</b>	UC6
<b>Use Case Name</b>	User Registration
<b>Requirements Traceability</b>	FR-6
<b>Purpose</b>	To allow new users to register in the system using email and password.
<b>Priority</b>	High
<b>Preconditions</b>	User is not already registered.
<b>Post conditions</b>	New user account created.
<b>Actors</b>	Community Member, Mosque Admin
<b>Extends</b>	None
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. User opens registration page.</li> <li>2. Enters name, email, password.</li> <li>3. System validates input.</li> <li>4. Account created successfully.</li> </ol>
<b>Alternate Flows</b>	User already exists then system shows “Email already registered.”
<b>Exceptions</b>	Invalid input fields or server error.
<b>Includes</b>	None

## Use Case #7

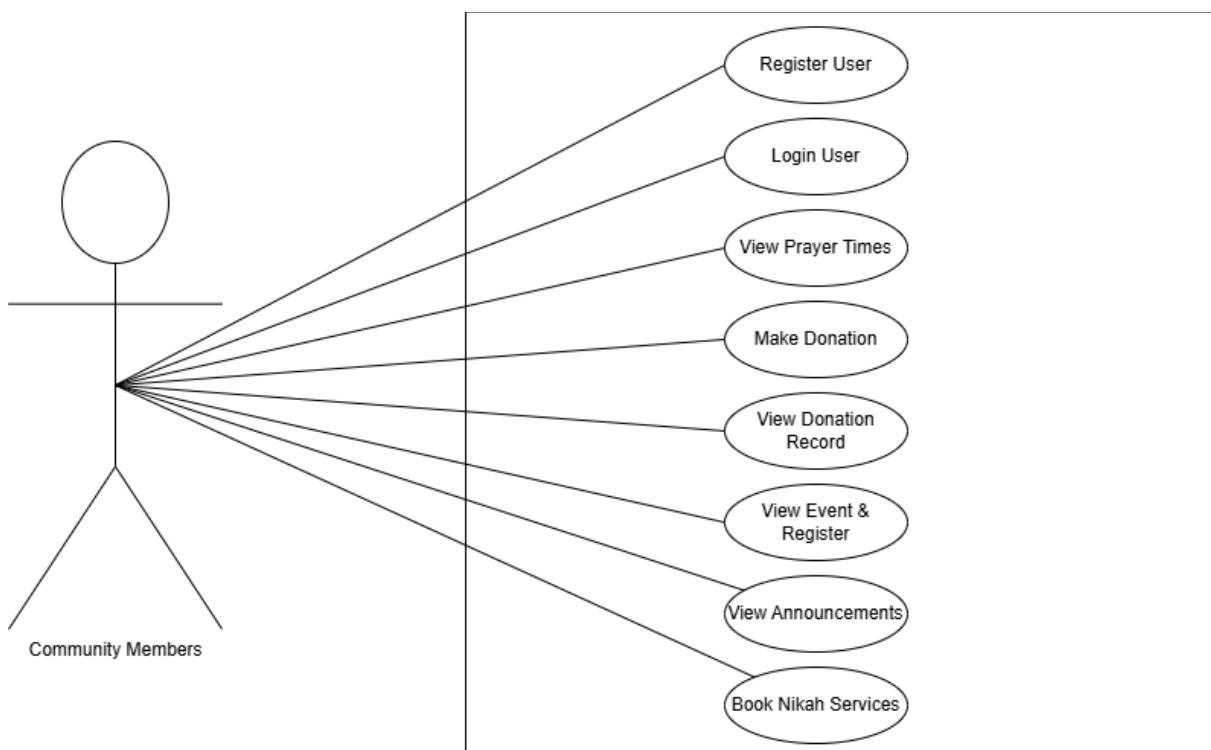
<b>UC Identifier</b>	UC7
<b>Use Case Name</b>	User Login
<b>Requirements Traceability</b>	FR-6
<b>Purpose</b>	To let registered users (admin or community member) log in using email and password.
<b>Priority</b>	High
<b>Preconditions</b>	User must be registered.
<b>Post conditions</b>	User successfully logged in
<b>Actors</b>	Mosque Admin, Community Member
<b>Extends</b>	None
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. User enters email and password.</li> <li>2. System checks credentials.</li> <li>3. If correct, login successful.</li> <li>4. website or admin dashboard opens.</li> </ol>
<b>Alternate Flows</b>	Wrong credentials then system show “Invalid email or password.”
<b>Exceptions</b>	Server not responding
<b>Includes</b>	None

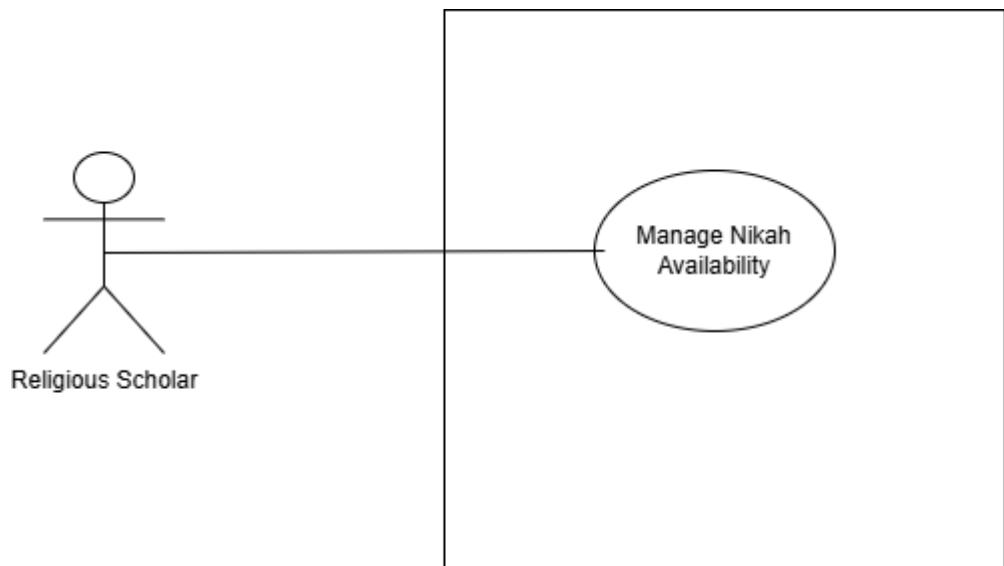
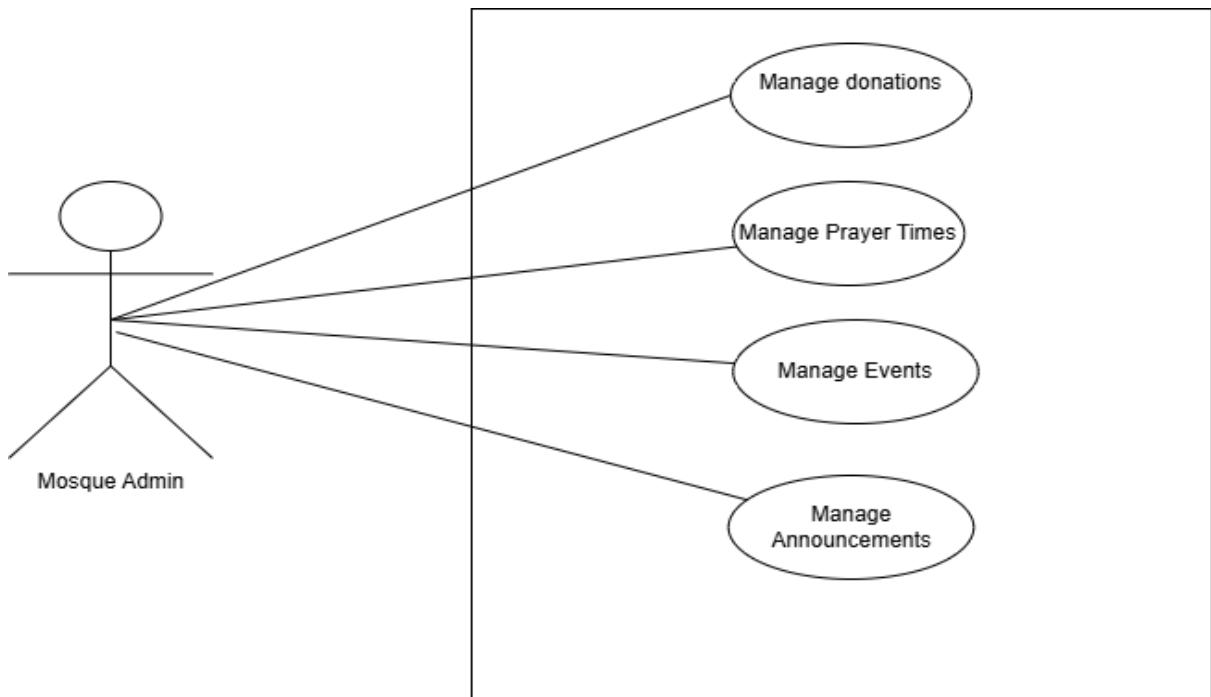
## Use Case #8

<b>UC Identifier</b>	UC8
<b>Use Case Name</b>	Online Donations
<b>Requirements Traceability</b>	FR-7
<b>Purpose</b>	Allow community members to donate online and track their donations
<b>Priority</b>	High
<b>Preconditions</b>	User must be logged in.
<b>Post conditions</b>	Donation recorded in database
<b>Actors</b>	Community Member
<b>Extends</b>	None
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. User clicks "Donate Online"</li> <li>2. Selects donation type and amount</li> <li>3. Enters personal details</li> <li>4. Clicks "Donate"</li> <li>5. Sees confirmation</li> </ol>
<b>Alternate Flows</b>	If server fails to save donation, system shows "Unable to save donation".
<b>Exceptions</b>	Server not responding
<b>Includes</b>	User authentication

## Use Case Diagram

These diagrams shows all the users and features of our E-Masjid system. It helps visualize how different users interact with different parts of the system.





# **Storyboards**

This section shows how users will use our system in real life. Each storyboard explains one main feature with simple steps that happen on screen.

## **Storyboard 1 – Online Donation System**

### **Scene Description:**

A community member wants to donate money to mosque through website without visiting in person. They want to see confirmation and later check where their money was used.

### **Steps:**

1. User goes to website and clicks "Donate Online" button.
2. Selects donation type from options (Zakat, Sadaqah, Mosque Fund).
3. Enters donation amount in rupees.
4. Fills in their name and contact details.
5. Clicks "Submit Donation" button.
6. System shows "Donation Successful!" message.
7. Admin sees new donation in dashboard with donor details.
8. When mosque spends money, admin adds expense record like "5000 for new fans".
9. All users can see transparency report showing donations and expenses.
10. Donor feels happy seeing how their money helped mosque.

## **Storyboard 2 – Event Management**

### **Scene Description:**

Mosque admin wants to organize a Quran learning class and community members want to register online for it.

### **Steps:**

1. Admin logs into system and goes to Events section.
2. Clicks "Create New Event" button.
3. Fills event details: "Quran Learning Class", date, time, description.
4. Sets maximum participants if needed.
5. Clicks "Publish Event" button.
6. Community member visits website and sees new event on homepage.
7. Clicks on event to see full details.
8. Clicks "Register for Event" button.
9. Fills simple form with name and phone number.
10. Gets confirmation message "Successfully Registered!".
11. Admin can see list of registered people for the event.

## **Storyboard 3 – Nikah Booking**

### **Scene Description:**

A community member wants to book a marriage (nikah) service online instead of visiting mosque in person.

### **Steps:**

1. User logs into system.
2. Goes to Nikah Booking page.
3. Fills required details like date, time, and contact info.
4. Submits the booking request.
5. System sends confirmation message.
6. Mosque admin sees new booking request in dashboard.
7. Admin checks available imam and confirms schedule.
8. User receives message “Your Nikah booking is confirmed.”

## **Storyboard 4 – Announcement System**

### **Scene Description:**

Mosque admin needs to share important announcements with entire community quickly.

### **Steps:**

1. Admin logs in and goes to Announcements section.
2. Clicks "Create New Announcement" button.
3. Writes announcement title and details.
4. Marks it as "Urgent" if important.
5. Clicks "Publish" button.
6. Announcement immediately appears on website homepage.

7. Community members visit website and see new announcement.
8. They read the important information.
9. No one misses important mosque updates anymore.

## **Summary**

In this SRS document, we explained all the main requirements and features of our E-Masjid System. We started by understanding the problems faced by mosque committees and community members, then used use cases and storyboards to identify the real needs of the system. We listed both functional and non-functional requirements like donation management, prayer time setup, service requests, and security. This document helped us to understand what our system will do, who will use it and how each feature will work. It will also help us in the next steps like design and development because all features and requirements are already clear. Overall this SRS gives a complete picture of the system before we start coding.

# References

## Books and Research Papers:

1. Ahmad, M., Hassan, A., & Khan, S. (2019). "Trust and Transparency in Religious Charitable Organizations in Pakistan." *Journal of Islamic Management Studies*, 12(3), 45-62.
2. Khan, F. (2020). "Digital Transformation of Religious Institutions: A Case Study of Mosques in Urban Pakistan." *Pakistan Journal of Information Technology*, 8(2), 112-128.

## Technology Documentation:

3. MongoDB Documentation - <https://www.mongodb.com/docs/>
4. React.js Official Documentation - <https://react.dev/>
5. Node.js Documentation - <https://nodejs.org/en/docs/>
6. Express.js Guide - <https://expressjs.com/>

## Design and Framework Resources:

7. Bootstrap Framework Documentation - <https://getbootstrap.com/docs/>
8. Chart.js Documentation - <https://www.chartjs.org/docs/>
9. JWT (JSON Web Tokens) - <https://jwt.io/introduction>

## Additional Resources:

10. Our original Final Year Project Proposal document
11. Discussions with mosque committee members
12. University guidelines for software requirements specification