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

List of Tables

List of Figure

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. We identified 6 core features, which are further divided into detailed use cases for better understanding.

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 7 main functional requirements 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 1

Table 1 Functional Requirement 1

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

Functional Requirement 2

Table 2 Functional Requirement 2

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

Functional Requirement 3

Table 3 Functional Requirement 3

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

Functional Requirement 4

Table 4 Functional Requirement 4

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

Functional Requirement 5

Table 5 Functional Requirement 5

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

Functional Requirement 6

Table 6 Functional Requirement 6

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

Functional Requirement 7

Table 7 Functional Requirement 7

Identifier	FR-7
Title	Online Donation System
Requirement	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.
Source	Community needs an easier way to donate
Rationale	People want to donate easily without visiting mosque
Business Rule	Each online donation must record donor information and amount
Dependencies	FR-6
Priority	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. We want:

- 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. We want:

- 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. We want:

- 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. We want:

- 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 one week 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 – Add Donor

Table 8 Use Case 1

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

Use Case #2 – Update Prayer Times

Table 9 Use Case 2

UC Identifier	UC2
Use Case Name	Update Prayer Times
Requirements Traceability	FR-3
Purpose	To set and display daily prayer times and special schedules for Jummah and Ramadan.
Priority	High
Preconditions	None

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

Use Case #3 – Book Nikah Services

Table 10 Use Case 3

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

Use Case #4 – Create Events

Table 11 Use Case 4

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

Use Case #5 – Post Announcement

Table 12 Use Case 5

UC Identifier	UC5
Use Case Name	Post Announcement
Requirements Traceability	FR-4
Purpose	To post and manage mosque announcements.
Priority	Medium
Preconditions	Admin must be logged in
Post conditions	Announcement shown on website.
Actors	Mosque Admin, Community Member

Extends	None
Main Success Scenario	1. Admin logs into system 2. Admin write new announcement. 3. System post it on the website. 4. User can view it anytime.
Alternate Flows	Admin edits or deletes announcement.
Exceptions	Internet issue while saving.
Includes	User authentication

Use Case #6 – User Registration

Table 13 Use Case 6

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

Use Case #7 – User Login

Table 14 Use Case 7

UC Identifier	UC7
Use Case Name	User Login
Requirements Traceability	FR-6

Purpose	To let registered users (admin or community member) log in using email and password.
Priority	High
Preconditions	User must be registered.
Post conditions	User successfully logged in
Actors	Mosque Admin, Community Member
Extends	None
Main Success Scenario	1. User enters email and password. 2. System checks credentials. 3. If correct, login successful. 4. website or admin dashboard opens.
Alternate Flows	Wrong credentials then system show “Invalid email or password.”
Exceptions	Server not responding
Includes	None

Use Case #8 – Online Donations

Table 15 Use Case 8

UC Identifier	UC8
Use Case Name	Online Donations
Requirements Traceability	FR-7
Purpose	Allow community members to donate online and track their donations
Priority	High
Preconditions	User must be logged in.
Post conditions	Donation recorded in database
Actors	Community Member, Donor
Extends	None
Main Success Scenario	1. User clicks "Donate Online" 2. Selects donation type and amount 3. Enters personal details 4. Clicks "Donate" 5. Sees confirmation

Alternate Flows	If server fails to save donation, system shows “Unable to save donation”.
Exceptions	Server not responding
Includes	User authentication

Use Case #9 – Check Nikah Requests

Table 16 Use Case 9

UC Identifier	UC9
Use Case Name	Check Nikah Requests
Requirements Traceability	FR-5
Purpose	To allow religious scholars to check new nikah booking requests and update their status as accepted or rejected.
Priority	Medium
Preconditions	The religious scholar account must already be created by the mosque admin, and the scholar must be logged into the system.
Post conditions	The nikah request status is updated to either “Accepted” or “Declined,” and the mosque admin is notified of the decision.
Actors	Religious Scholar, Mosque Admin
Extends	None
Main Success Scenario	1.Religious scholar logs into system 2.Goes to "Nikah Requests" page 3.Views list of pending booking requests 4.Clicks on a request to see details (date, time, contact info) 5.Clicks "Accept" or "Decline" button 6.System updates request status 7.Admin sees updated status in dashboard
Alternate Flows	If no requests available, system shows "No pending requests" If scholar tries to accept conflicting time, system shows "Time not available"
Exceptions	If system error occurs, shows "Unable to update request"
Includes	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.

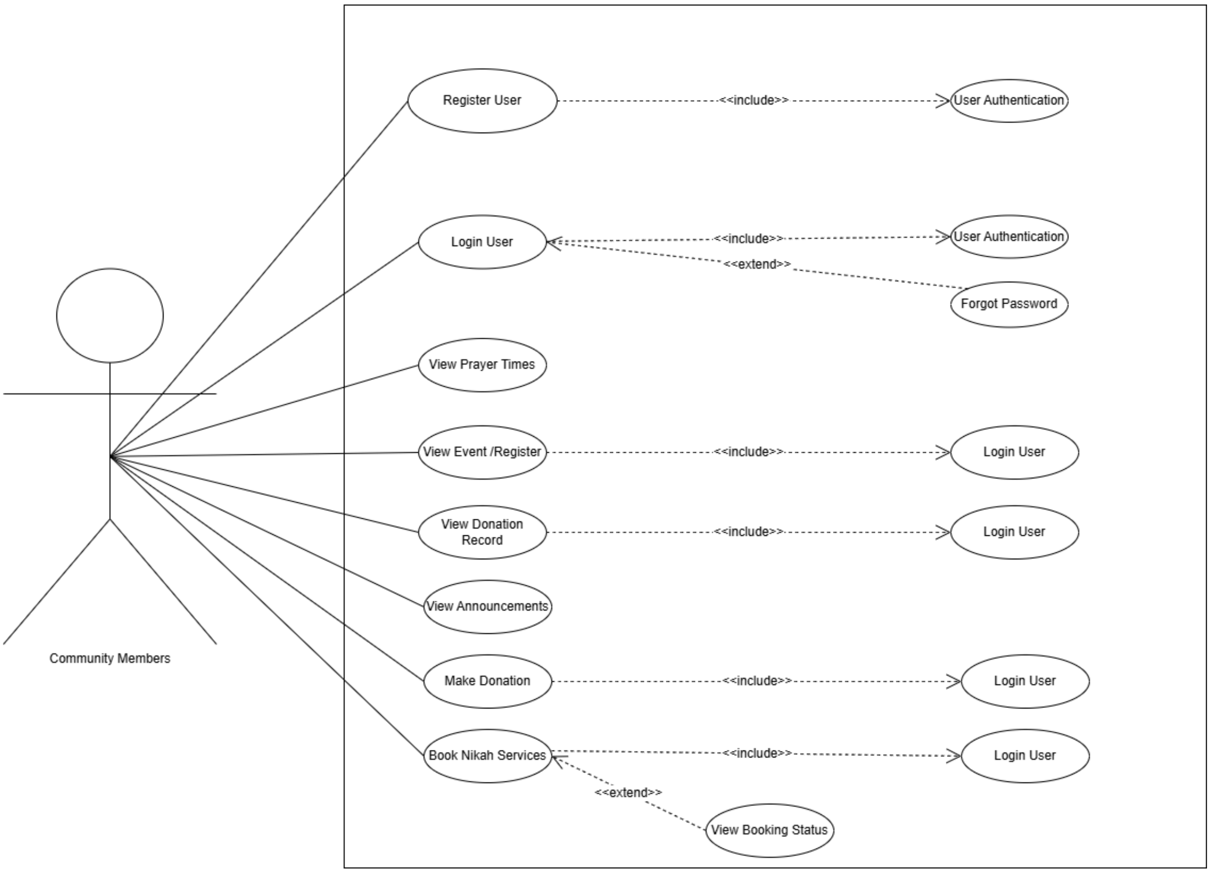

Figure 1 Use Case diagram

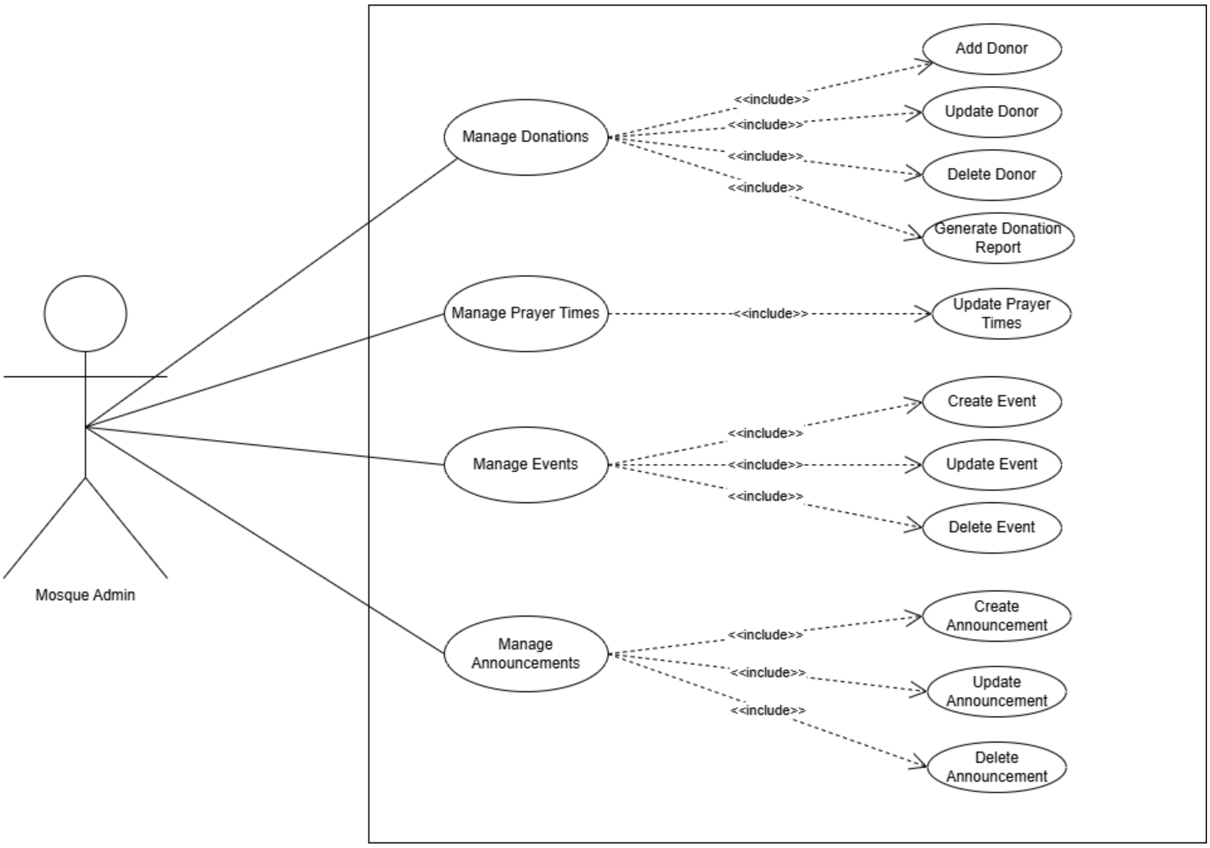

Figure 2 Use Case diagram

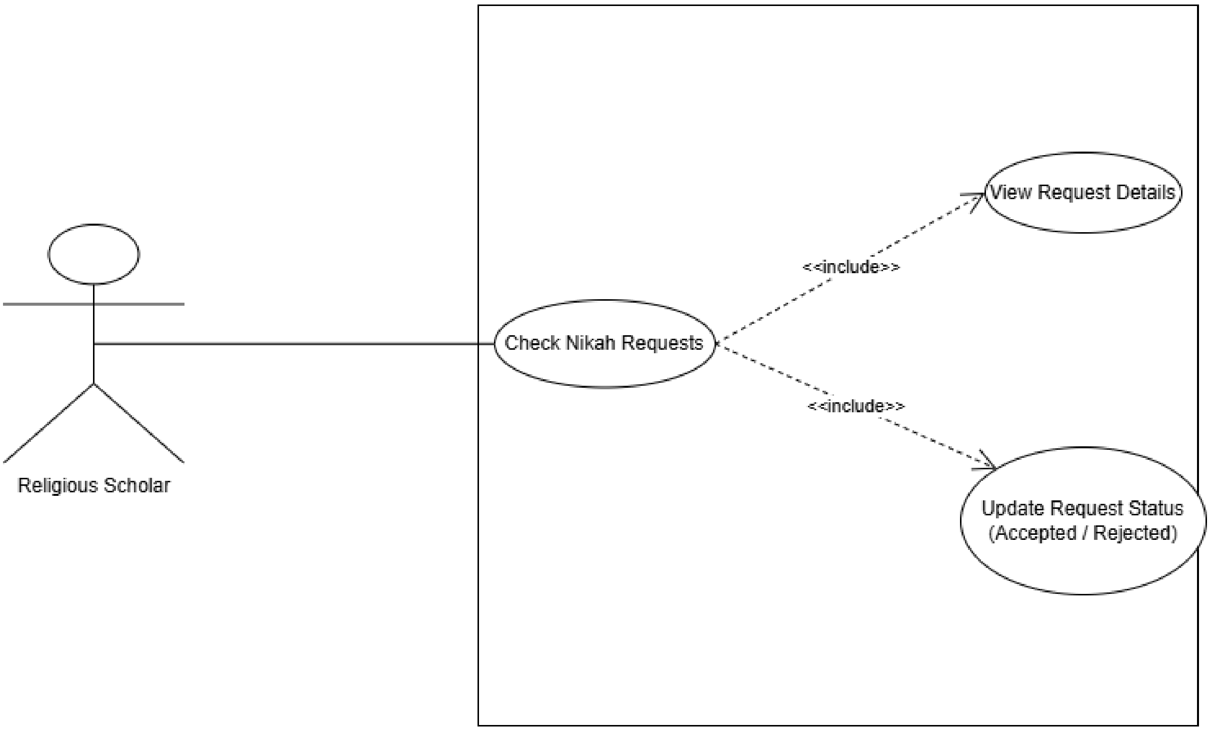

Figure 3 Use Case diagram

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