

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>	2
<i>Requirements Analysis</i>	6
User classes and characteristics	6
Requirement Identifying Technique	6
Functional Requirements	7
Functional Requirement 1.....	7
Functional Requirement 2.....	7
Functional Requirement 3.....	8
Functional Requirement 4.....	8
Functional Requirement 5.....	9
Functional Requirement 6.....	9
Functional Requirement 7.....	10
Functional Requirement 8.....	10
Functional Requirement 9.....	11
Functional Requirement 10.....	11
Functional Requirement 11.....	11
Non-Functional Requirements	12
Reliability.....	12
Usability	12
Performance	13
Security	13
External Interface Requirements	13
User Interfaces Requirements.....	13
Software interfaces	14
Hardware interfaces	14
Communications interfaces.....	14
<i>Use case Analysis</i>	15
Community User Use Cases	15
Use Case #1 – Register User	15
Use Case #2 – Login User	15
Use Case #3 – Forgot Password	16
Use Case #4 – View Prayer Times	16
Use Case #5 – View Events & Register.....	17
Use Case #6 – View Donation Records	17
Use Case #7 – View Announcements	18

Use Case #8 – Make Donation	19
Use Case #9 – Book Nikah Services	19
Use Case #10 – View Booking Status.....	20
Admin Use Cases.....	20
Use Case #11 – Admin Login	20
Use Case #12 – Manage Donations & Expenses	21
Use Case #13 – Manage Prayer Times	21
Use Case #14 – Manage Events.....	22
Use Case #15 – Manage Announcements.....	22
Use Case #16 – Manage Religious Scholar Account	23
Religious Scholar Use Case	24
Use Case #17 – Check Nikah Requests.....	24
Use Case Diagram	25
<i>Storyboards.....</i>	27
Storyboard 1 – Online Donation System	27
Storyboard 2 – Event Management	27
Storyboard 3 – Nikah Booking	29
Storyboard 4 – Announcement System	29
Summary	30
<i>References</i>	31

List of Tables

Table 1 Functional Requirement 1.....	7
Table 2 Functional Requirement 2.....	7
Table 3 Functional Requirement 3.....	8
Table 4 Functional Requirement 4.....	8
Table 5 Functional Requirement 5.....	9
Table 6 Functional Requirement 6.....	9
Table 7 Functional Requirement 7.....	10
Table 8 Functional Requirement 8.....	10
Table 9 Functional Requirement 9.....	11
Table 10 Functional Requirement 10.....	11
Table 11 Functional Requirement 11.....	11
Table 12 Use Case 1	15
Table 13 Use Case 2.....	15
Table 14 Use Case 3.....	16
Table 15 Use Case 4.....	16

Table 16 Use Case 5	17
Table 17 Use Case 6	17
Table 18 Use Case 7	18
Table 19 Use Case 8	19
Table 20 Use Case 9	19
Table 21 Use Case 10	20
Table 22 Use Case 11	20
Table 23 Use Case 12	21
Table 24 Use Case 13	21
Table 25 Use Case 14	22
Table 26 Use Case 15	22
Table 27 Use Case 16	23
Table 28 Use Case 17	24

List of Figure

Figure 1 Use Case diagram of Community members	25
Figure 2 Use Case diagram of Mosque Admin	26
Figure 3 Use Case diagram of Religious Scholar	26

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 the following use cases for our system:

- UC1: Register User
- UC2: Login User
- UC3: Forgot Password
- UC4: View Prayer Times
- UC5: View Events & Register
- UC6: View Donation Records
- UC7: View Announcements
- UC8: Make Donation
- UC9: Book Nikah Services
- UC10: View Booking Status
- UC11: Admin Login
- UC12: Manage Donations & Expenses
- UC13: Manage Prayer Times
- UC14: Manage Events
- UC15: Manage Announcements

- UC16: Manage Religious Scholar Account
- UC17: Check Nikah Requests

These use cases cover all main features of the system and help us understand how users will perform different tasks.

Functional Requirements

We identified 11 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	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 2

Table 2 Functional Requirement 2

Identifier	FR-2
Title	Show Donation Reports
Requirement	The system will show donation records, expense details, and financial reports so people can see both income and spending.
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-3, FR-10
Priority	High

Functional Requirement 3

Table 3 Functional Requirement 3

Identifier	FR-3
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	FR-1
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	FR-1

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	FR-1
Priority	Medium

Functional Requirement 6

Table 6 Functional Requirement 6

Identifier	FR-6
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	FR-1
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. They will enter the amount and personal details, process payments securely using Stripe, and receive a payment confirmation receipt.
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-1
Priority	Medium

Functional Requirement 8

Table 8 Functional Requirement 8

Identifier	FR-8
Title	Password Reset via Email
Requirement	If users forget their password, they can reset it using their email address.
Source	Community feedback
Rationale	People often forget passwords and need an easy way to get back into their account
Business Rule	The reset link should work only for 24 hours and can be used one time
Dependencies	FR-1
Priority	High

Functional Requirement 9

Table 9 Functional Requirement 9

Identifier	FR-9
Title	Check Nikah Booking Status
Requirement	Users can see if their Nikah service request is pending, accepted, or rejected
Source	Community feedback
Rationale	People want to know what's happening with their booking request
Business Rule	The status should update automatically when scholar changes it
Dependencies	FR-5
Priority	Medium

Functional Requirement 10

Table 10 Functional Requirement 10

Identifier	FR-10
Title	Record Mosque Expenses
Requirement	Admin can add where mosque money is spent like for repairs, electricity etc.
Source	Community trust issues
Rationale	Community wants to see how their donations are being used
Business Rule	Every expense must have description, amount, and date
Dependencies	FR-1
Priority	High

Functional Requirement 11

Table 11 Functional Requirement 11

Identifier	FR-11
Title	Create Scholar Accounts

Requirement	Admin can create special accounts for religious scholars who perform Nikah
Source	Use Case Analysis
Rationale	Scholars need their own accounts to manage Nikah requests
Business Rule	Only admin can create and manage these special accounts
Dependencies	FR-1
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 must be reliable in the day to day running of the mosque. It should not crash frequently and should recover quickly if any problems occur.

- The system should be available 95% of the time during peak hours
- If system goes down, it should recover within 30 minutes
- Donation data should not be lost even if system has problems
- Important data should be automatically backed up after every week

Usability

The system must be easy to use and less complex to the mosque admins and ordinary users. All buttons and forms will be clear and labeled properly.

- Prayer times should be available to new users in 2 clicks
- Donation recording process should take less than 3 minutes for admin
- Nikah booking form should be completable within 5 minutes
- Large fonts and clear buttons should be used by the elderly users in interface
- Every key feature must be available at home page

Performance

When there are several people using the system, it should be fast and easily functioning.

- 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
- The registration of the event must take a maximum of 5 seconds

Security

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

- Database should store user passwords in an encrypted format
- The system must not allow unauthorized access to administrator functions
- Session should timeout after 1 hour of inactivity
- All payment transactions must be secured using SSL encryption

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

We will have a clean and simple interface on our system which will be effective with the mosque users who are aged, administrators and community members.

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:

- Stripe payment gateway will be used for real online donations
- Email service for password reset functionality.

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:

- Web access via Standard HTTP/HTTPS
- Internet connection needed to operate 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

Community User Use Cases

Use Case #1 – Register User

Table 12 Use Case 1

UC Identifier	UC1
Use Case Name	Register User
Requirements Traceability	FR-1
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
Extends	None
Main Success Scenario	<ol style="list-style-type: none">1. User opens registration page.2. Enters name, email, password.3. System validates input.4. Account created successfully.5. User can now login to the system
Alternate Flows	User already exists then system shows “Email already registered.”
Exceptions	Invalid input fields or server error.
Includes	None

Use Case #2 – Login User

Table 13 Use Case 2

UC Identifier	UC2
Use Case Name	Login User
Requirements Traceability	FR-1
Purpose	To let registered users log in using email and password.
Priority	High
Preconditions	User must be registered.
Post conditions	User successfully logged in
Actors	Community Member
Extends	None

Main Success Scenario	1. User enters email and password. 2. System checks credentials. 3. If correct, login successful. 4. If correct, user is logged in and taken to homepage.
Alternate Flows	If wrong password then system show "Invalid email or password."
Exceptions	Server not responding
Includes	User Authentication

Use Case #3 – Forgot Password

Table 14 Use Case 3

UC Identifier	UC3
Use Case Name	Forgot Password
Requirements Traceability	FR-8
Purpose	To help users who forgot their password
Priority	High
Preconditions	User must be registered with valid email
Post conditions	User can set new password and login
Actors	Community Member
Extends	Login User
Main Success Scenario	1. User clicks "Forgot Password" on login page 2. Enters email address 3. System sends password reset link to email 4. User clicks link in email 5. Enters new password 6. System updates password 7. User can now login with new password
Alternate Flows	If email not found, system still shows "If email exists, reset link sent" for security
Exceptions	If email service is not working
Includes	User authentication

Use Case #4 – View Prayer Times

Table 15 Use Case 4

UC Identifier	UC4
Use Case Name	View Prayer Times
Requirements Traceability	FR-6

Purpose	To let users see daily prayer timings
Priority	High
Preconditions	None
Post conditions	User can see all prayer times
Actors	Community Member
Extends	None
Main Success Scenario	<ol style="list-style-type: none"> 1. User visits website homepage 2. Prayer times are displayed on top of page 3. User can see Fajr, Zuhr, Asr, Maghrib, Isha times 4. Special timings for Jummah are also shown
Alternate Flows	User can view weekly prayer schedule
Exceptions	If admin hasn't updated times, show default times
Includes	None

Use Case #5 – View Events & Register

Table 16 Use Case 5

UC Identifier	UC5
Use Case Name	View Events & Register
Requirements Traceability	FR-4
Purpose	To see mosque events and register for them
Priority	Medium
Preconditions	User must be logged in to register
Post conditions	User registered for event
Actors	Community Member
Extends	None
Main Success Scenario	<ol style="list-style-type: none"> 1. User goes to Events page 2. Views list of upcoming events 3. Clicks on event to see details 4. Clicks "Register for Event" 5. System confirms registration 6. User gets confirmation message
Alternate Flows	User can view events without login, but needs login to register
Exceptions	If system cannot register user due to server issue
Includes	Login User

Use Case #6 – View Donation Records

Table 17 Use Case 6

UC Identifier	UC6
Use Case Name	View Donation Records
Requirements Traceability	FR-2
Purpose	To see donation reports and where money is spent
Priority	High
Preconditions	User must be logged in to view reports
Post conditions	User can see financial reports
Actors	Community Member
Extends	None
Main Success Scenario	<ol style="list-style-type: none"> 1. User goes to donation report page 2. Views donation records and amounts 3. Sees expense details like "5000 for new fans" 4. Builds trust in mosque management
Alternate Flows	User can filter reports by date filter
Exceptions	If no data available, show "No records yet"
Includes	Login User

Use Case #7 – View Announcements

Table 18 Use Case 7

UC Identifier	UC7
Use Case Name	View Announcements
Requirements Traceability	FR-4
Purpose	To read important mosque announcements
Priority	Medium
Preconditions	None
Post conditions	User reads announcements
Actors	Community Member
Extends	None
Main Success Scenario	<ol style="list-style-type: none"> 1. User visits website homepage 2. Announcements are shown in main section 3. User reads important updates 4. Urgent announcements are highlighted in red
Alternate Flows	User can see old announcements
Exceptions	If no announcements, show "No current announcements"
Includes	None

Use Case #8 – Make Donation

Table 19 Use Case 8

UC Identifier	UC8
Use Case Name	Make Donation
Requirements Traceability	FR-7
Purpose	To donate money to mosque online using real payments
Priority	Medium
Preconditions	User must be logged in
Post conditions	Donation processed and recorded
Actors	Community Member, Donor
Extends	None
Main Success Scenario	<ol style="list-style-type: none"> 1. User clicks "Donate Online" 2. Selects donation type 3. Enters amount in rupees 4. Enters card details for Stripe payment 5. Clicks "Donate Now" 6. Stripe processes payment 7. System shows "Donation Successful" with receipt
Alternate Flows	If user is not logged in, system redirects user to Login page before allowing donation.
Exceptions	If payment fails then show "Payment failed try again"
Includes	Login User

Use Case #9 – Book Nikah Services

Table 20 Use Case 9

UC Identifier	UC9
Use Case Name	Book Nikah Services
Requirements Traceability	FR-5
Purpose	To book religious scholar for marriage ceremony
Priority	Medium
Preconditions	User must be logged in
Post conditions	Nikah booking request submitted
Actors	Community Member
Extends	None
Main Success Scenario	<ol style="list-style-type: none"> 1. User goes to Nikah Services page 2. Selects preferred date and time

	3. Fills contact details and ceremony information 4. Submits the request 5. System sends confirmation "Request Submitted"
Alternate Flows	User can view their previous bookings
Exceptions	If date not available, show "Please select another date"
Includes	Login User

Use Case #10 – View Booking Status

Table 21 Use Case 10

UC Identifier	UC10
Use Case Name	View Booking Status
Requirements Traceability	FR-9
Purpose	To check if Nikah booking is accepted or pending
Priority	Medium
Preconditions	User must be logged in and have booking request
Post conditions	User sees current status of their request
Actors	Community Member
Extends	Book Nikah Services
Main Success Scenario	1. User goes to My Bookings page 2. Views list of their Nikah requests 3. See status like Pending, Accepted, or Rejected 4. If accepted, sees confirmed date and time 5. If rejected, sees reason if provided
Alternate Flows	User can cancel pending requests
Exceptions	If no bookings, show "No booking requests yet"
Includes	Login User

Admin Use Cases

Use Case #11 – Admin Login

Table 22 Use Case 11

UC Identifier	UC11
Use Case Name	Admin Login
Requirements Traceability	FR-1
Purpose	To let mosque admin access the admin dashboard
Priority	High
Preconditions	Admin must have admin account credentials

Post conditions	Admin gains access to admin dashboard
Actors	Mosque Admin
Extends	None
Main Success Scenario	<ol style="list-style-type: none"> 1. Admin goes to special /admin login URL 2. Enters admin username and password 3. Clicks "Admin Login" 4. System verifies admin details 5. Admin dashboard opens with all management options
Alternate Flows	If wrong details, show "Invalid admin login"
Exceptions	If admin account not set up yet
Includes	User Authentication

Use Case #12 – Manage Donations & Expenses

Table 23 Use Case 12

UC Identifier	UC12
Use Case Name	Manage Donations & Expenses
Requirements Traceability	FR-3, FR-10
Purpose	To handle all money records like donations and expenses
Priority	High
Preconditions	Admin must be logged in
Post conditions	Financial records updated
Actors	Mosque Admin
Extends	None
Main Success Scenario	<ol style="list-style-type: none"> 1. Admin goes to Manage Donations & Expenses section 2. Can add new cash donations with donor details 3. Can record expenses like "5000 for mosque repairs" 4. Can edit or delete existing records 5. System updates financial reports automatically
Alternate Flows	Admin can generate monthly financial reports
Exceptions	If invalid data entered, show appropriate error messages
Includes	Admin Login

Use Case #13 – Manage Prayer Times

Table 24 Use Case 13

UC Identifier	UC13
Use Case Name	Manage Prayer Times
Requirements Traceability	FR-6

Purpose	To set and update daily prayer timings for community
Priority	High
Preconditions	Admin must be logged in
Post conditions	New prayer times displayed on website
Actors	Mosque Admin
Extends	None
Main Success Scenario	<ol style="list-style-type: none"> 1. Admin goes to Prayer Times section 2. Updates all five daily prayer times 3. Sets special Jummah prayer timings 4. Clicks "Save Times" 5. New times immediately show on website for everyone
Alternate Flows	Admin can set weekly schedule instead of daily updates
Exceptions	If invalid time format, show "Please use correct time format"
Includes	Admin Login

Use Case #14 – Manage Events

Table 25 Use Case 14

UC Identifier	UC14
Use Case Name	Manage Events
Requirements Traceability	FR-4
Purpose	To create, update, and manage mosque events
Priority	Medium
Preconditions	Admin must be logged in
Post conditions	Events published and visible to community
Actors	Mosque Admin
Extends	None
Main Success Scenario	<ol style="list-style-type: none"> 1. Admin goes to Events Management section 2. Creates new events with details and registration options 3. Updates existing event information 4. Deletes or cancels events when needed 5. Events appear on website for community registration
Alternate Flows	Admin can save events as drafts before publishing
Exceptions	If past date entered, show "Event date cannot be in past"
Includes	Admin Login

Use Case #15 – Manage Announcements

Table 26 Use Case 15

UC Identifier	UC15
----------------------	------

Use Case Name	Manage Announcements
Requirements Traceability	FR-4
Purpose	To post and manage important mosque announcements
Priority	Medium
Preconditions	Admin must be logged in
Post conditions	Announcements visible on website
Actors	Mosque Admin
Extends	None
Main Success Scenario	<ol style="list-style-type: none"> 1. Admin goes to Announcements section 2. Creates new announcements with titles and details 3. Updates existing announcements if information changes 4. Deletes old or incorrect announcements 5. Marks announcements as urgent if important 6. Announcements show on website immediately
Alternate Flows	Admin can schedule announcements for future dates
Exceptions	If announcement too long, show "Please shorten announcement"
Includes	Admin Login

Use Case #16 – Manage Religious Scholar Account

Table 27 Use Case 16

UC Identifier	UC16
Use Case Name	Manage Religious Scholar Account
Requirements Traceability	FR-11
Purpose	To create and manage account for Nikah service scholars
Priority	Medium
Preconditions	Admin must be logged in
Post conditions	Scholar accounts created and activated for Nikah management
Actors	Mosque Admin
Extends	None
Main Success Scenario	<ol style="list-style-type: none"> 1. Admin goes to Scholar Management section 2. Creates new scholar accounts with login credentials 3. Updates scholar account details if needed 4. Deletes or deactivates scholar accounts 5. Scholars receive login details and can manage Nikah requests
Alternate Flows	Admin can reset scholar passwords if forgotten
Exceptions	If email already used, show "Email already registered"
Includes	Admin Login

Religious Scholar Use Case

Use Case #17 – Check Nikah Requests

Table 28 Use Case 17

UC Identifier	UC17
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 request status is updated in the admin dashboard.
Actors	Religious Scholar, Mosque Admin
Extends	None
Main Success Scenario	<ol style="list-style-type: none"> 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	<p>If no requests available, system shows "No pending requests"</p> <p>If scholar tries to accept conflicting time, system shows "Time not available"</p>
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 of Community members

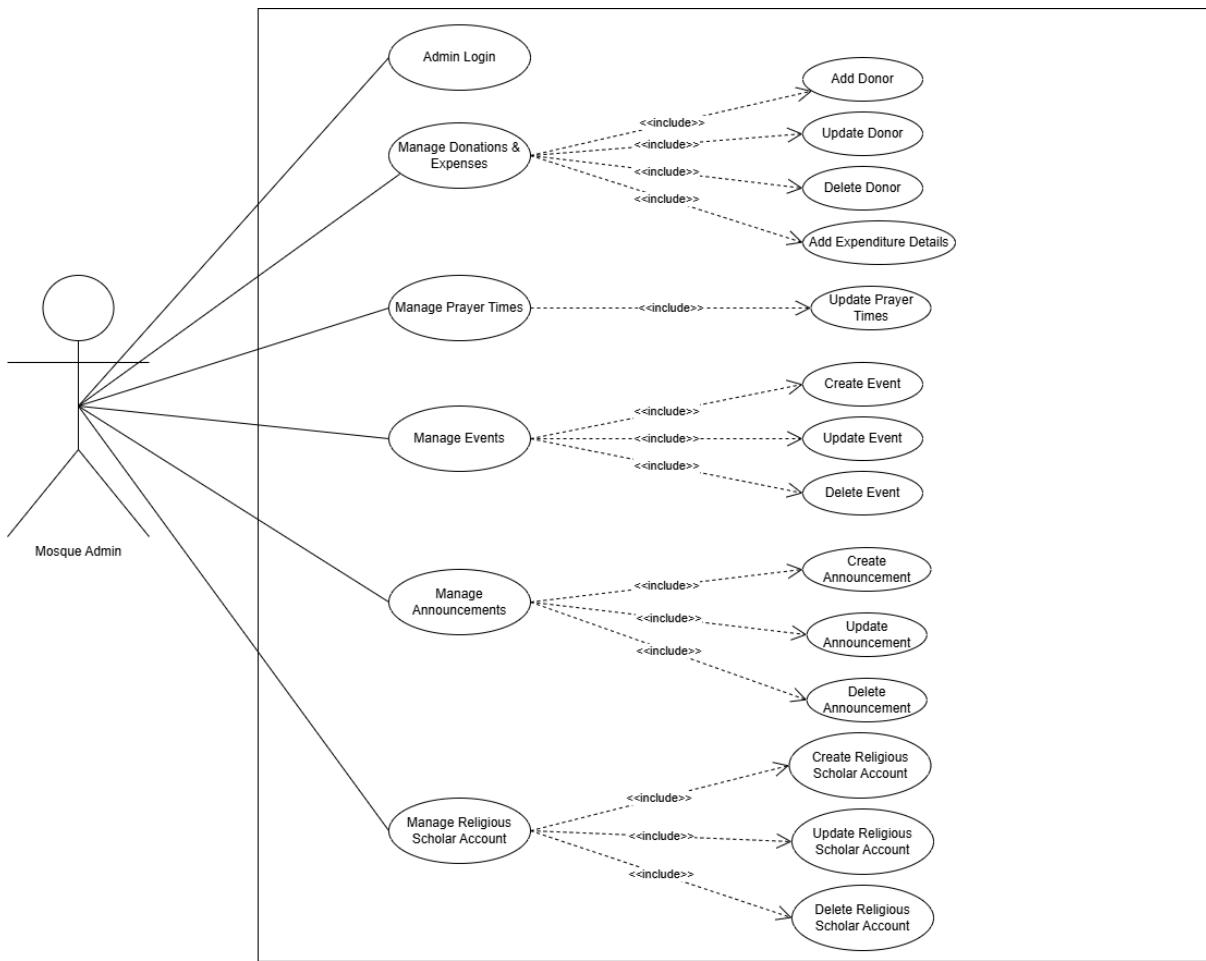


Figure 2 Use Case diagram of Mosque Admin

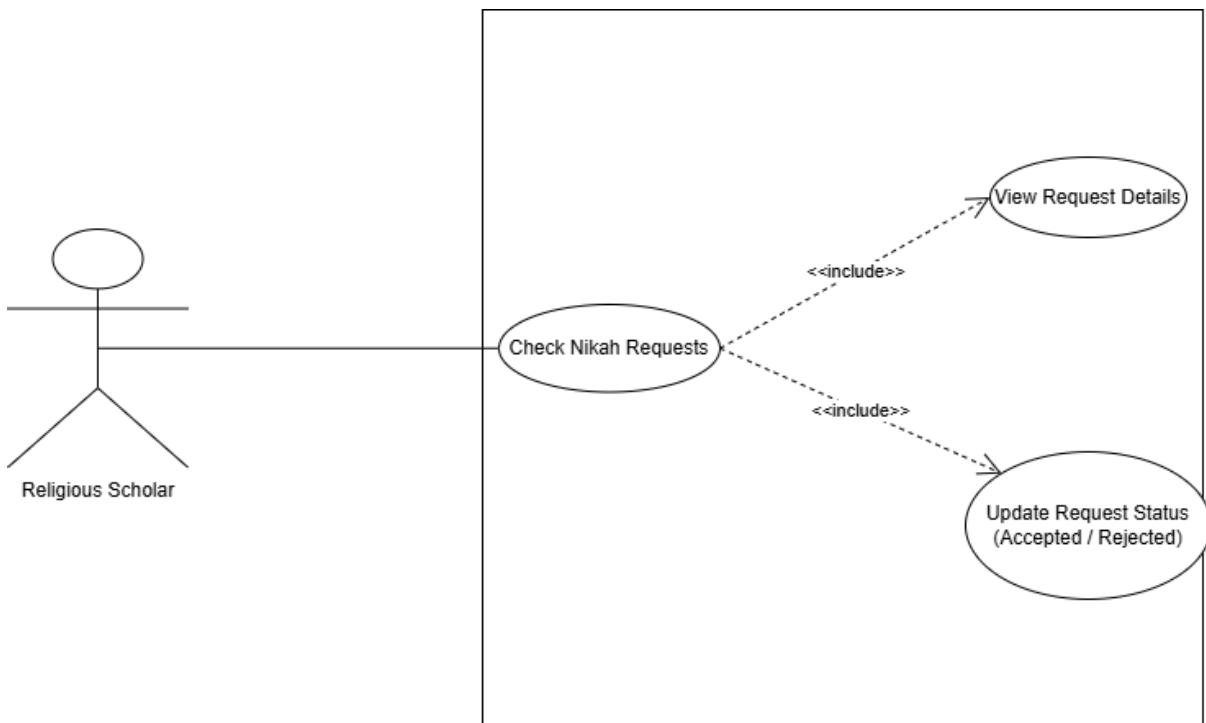


Figure 3 Use Case diagram of Religious Scholar

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 like Zakat, Sadaqah, Mosque Fund.
3. Enters donation amount in rupees.
4. Enters card details like card number, expiry and CVC
5. Stripe processes the payment.
6. On successful payment, 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. Religious scholar checks available date 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 <https://jwt.io/introduction>
10. Stripe Documentation: <https://stripe.com/docs>

Additional Resources:

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