SOFTWARE REQUIREMENTS SPECIFICATION

for

Samvidhan

Prepared by : Akshat Jain (24ESKIT016)

Submitted to : Sanju Choudhary

September 7, 2025

Contents

1 Introduction

1.1 Purpose

The purpose of the Samvidhan – Parliament Question Management Portal is to provide a digital platform where citizens can directly submit questions to the Parliament. The system is designed to strengthen democratic participation by giving people a voice to raise their queries, concerns, and suggestions to the concerned departments.
Through this portal, citizens can:

- Submit questions related to governance, policies, and public issues.
- Track the status of their questions and whether they are accepted for discussion.
- View responses or clarifications provided by parliamentary representatives.

1.2 Intended Audience and Reading Suggestions

This project is intended towards:

- Citizens : To understand how they can ask questions and track responses.
- Members of Parliament : To access and respond to public questions.
- System Administrators : To manage users, security, and technical aspects.

1.3 Project Scope

Samvidhan is a citizen-driven Parliament Question Management Portal aimed at bridging the gap between people and lawmakers.

The portal will:

- Provide transparency by allowing the public to view answered questions in a searchable archive.
- Allow citizens to submit questions digitally.
- Provide categorization of questions (policy, governance, local issues, public interest).
- Enable MPs to view, review, and respond to public questions.
- Include a moderation workflow where Admin filter inappropriate or irrelevant questions.

The project's main goal is to enhance citizen participation in democracy, ensure greater accountability from representatives, and reduce dependency on offline/manual communication.

2 Overall Description

2.1 Product Perspective

The Samvidhan portal will be a web-based application that serves as a centralized system for citizen–Parliament interaction. It is an independent platform that integrates with Parliament's existing IT infrastructure for authentication and session management.

It will follow a client-server model with secure APIs to ensure scalability, security, and reliability.

2.2 User Classes and Characteristics

"Samvidhan" has basically 4 types of users.

1. Citizens (General Public): Submit questions; non-technical; require a simple interface.
2. Members of Parliament (MPs): Respond to questions; moderate digital literacy expected.
3. System Administrators: Handle maintenance, manage users, roles, and logs.

2.3 Product Functions

Citizen Features:

- Register/login to the portal.
- Submit questions under specific categories.
- Track status of submitted questions.
- View past responses and public archive.

MP Features:

- Access assigned questions.
- Draft and submit responses.

Admin Features:

- Manage user activity and permissions.

- Asks Questions directly to MP's
- Monitor system performance.

2.4 Operating Environment

The website will be operate in any Operating Environment - Windows, Mac, Linux etc.

3 System Features

"Samvidhan" is a Question Portal. So the main art of this product is to provide answers of the questions put up by the citizens.

3.1 Description and Priority

"Samvidhan" has features that are main and also some are sub. But all the feature is necessary for this software.

1. High Priority: Citizen question submission, Secretariat approval workflow, MP responses.
2. Medium Priority: Notifications, deadlines, tracking.
3. Low Priority: Analytics, public reports, sentiment analysis.

3.2 Functional Requirements

"Samvidhan" has the following features:

- The system shall allow citizens to register and log in.
- The system shall allow citizens to submit questions under specific categories.
- The system shall allow MPs to view and respond to assigned questions.
- The system shall display the status of each question (Submitted, Approved, Assigned, Answered).
- The system shall maintain a public archive of answered questions.
- The system shall allow admins to manage user accounts and permissions.

Following Technology Stack is used to achieve these features:

- Frontend: HTML5, CSS3/Bootstrap, JavaScript.
- Backend: Node.js.
- Database: MongoDBL.
- Authentication: UserId, Password (JVT Authentication).

4 Other Nonfunctional Requirements

4.1 Performance REquirements

- Portal must be available 24/7.
- Submissions should be fast and lag proof,
- Portal shall support enough users.

4.2 Software Quality Requirements

- Reliability: Backup and recovery mechanisms.
- Usability: Easy-to-use citizen interface.
- Maintainability: Modular codebase with documentation.
- Scalability: Must handle high traffic during Parliament sessions.
- Transparency: Public access to answered questions.
- The system should support exporting questions and answers in PDF format.

4.3 Business Requirements

- Citizens must be registered to submit questions.
- Each citizen may submit a limited number of questions per session.
- Admin must approve all questions before MPs see them.
- MPs must respond within a defined timeframe.

5 Other Requirements

Support for exporting questions and answers in PDF format.
It should provide analytics for policymakers (e.g., most common citizen concerns).
Samvidhan should include help/guidance for first-time users.
Mobile app version may be developed in the future.
It will need re-factoring and further the requirements can be changed as the field is changing frequently.

6 UML Diagrams

6.1 Use Case Diagram

Abstract:Citizens can register/login, access the homepage, view dashboard, ask questions, track the status of their queries, and view departments.Admins can manage users, view questions, answer questions, and manage departments.
This gives a high-level overview of the system's functionality and scope.

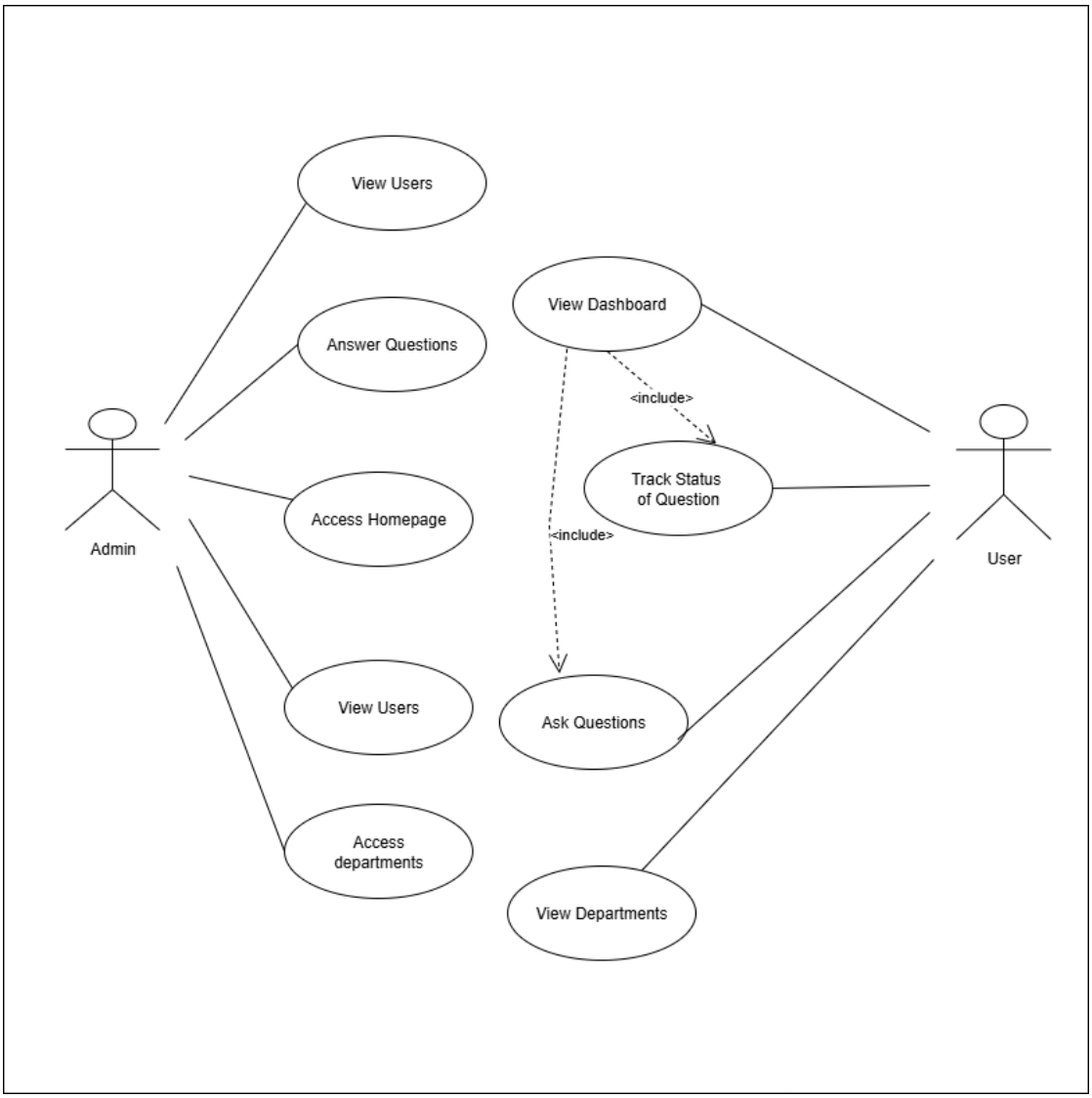

Figure 6.1: UseCase Diagram

6.2 Activity Diagram

Abstract: The Activity Diagram describes the workflow of activities for both Users and Admins.
For Users: The process starts from the homepage, followed by login/registration, accessing the dashboard, asking questions, tracking status, viewing answers, and giving feedback. For Admins: The process includes login, accessing the dashboard, reviewing user questions, sending them to the relevant department, providing answers, and requesting feedback.
This diagram models the dynamic behavior of the system, representing control flows and decision points in different scenarios.

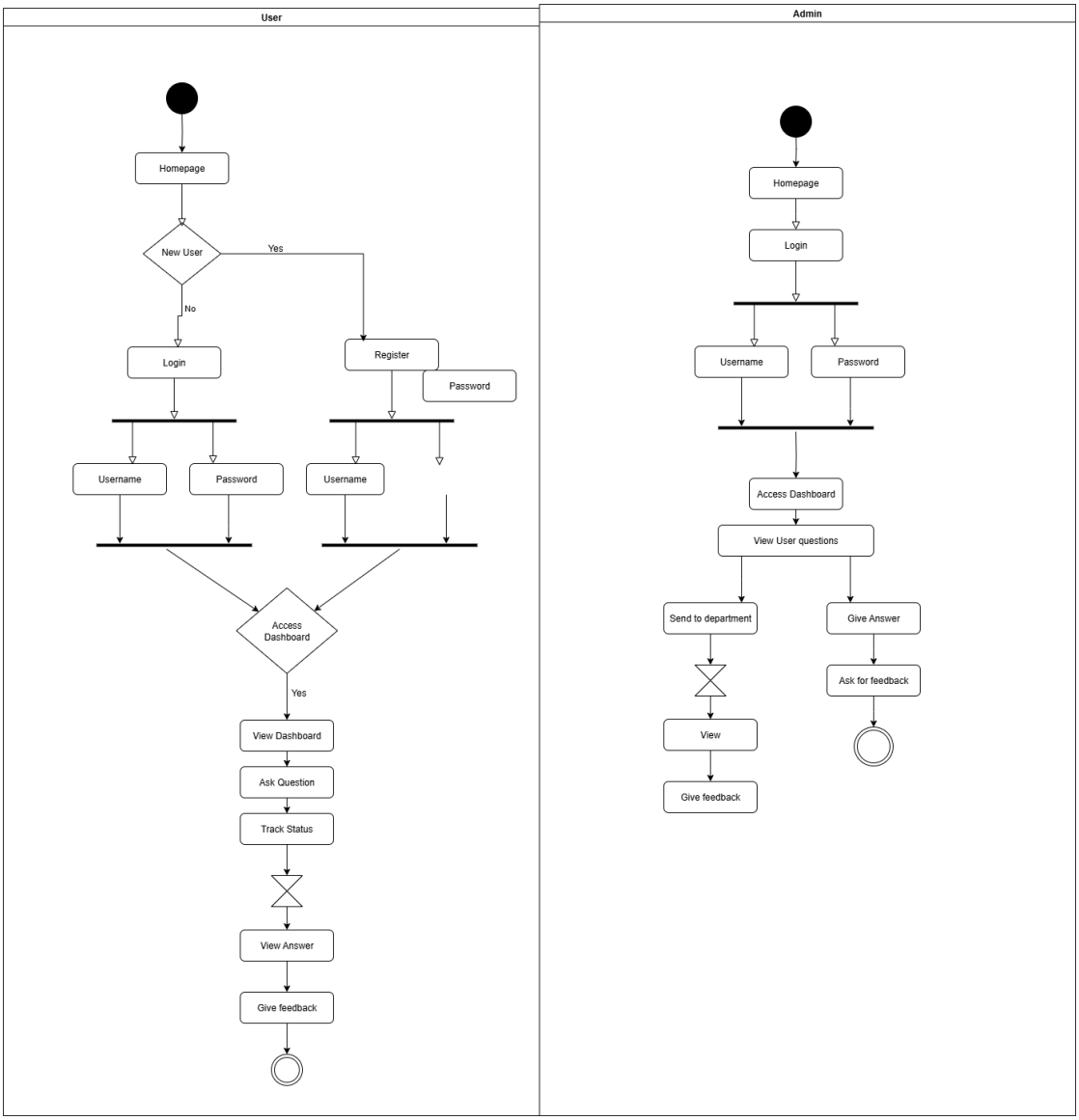

Figure 6.2: ctivity Diagram

6.3 Sequence Diagram

Abstract: The Sequence Diagram illustrates the step-by-step flow of interactions between the User, Admin, Parliament, and the Database.
This diagram highlights the chronological flow of events and message exchanges in the system.

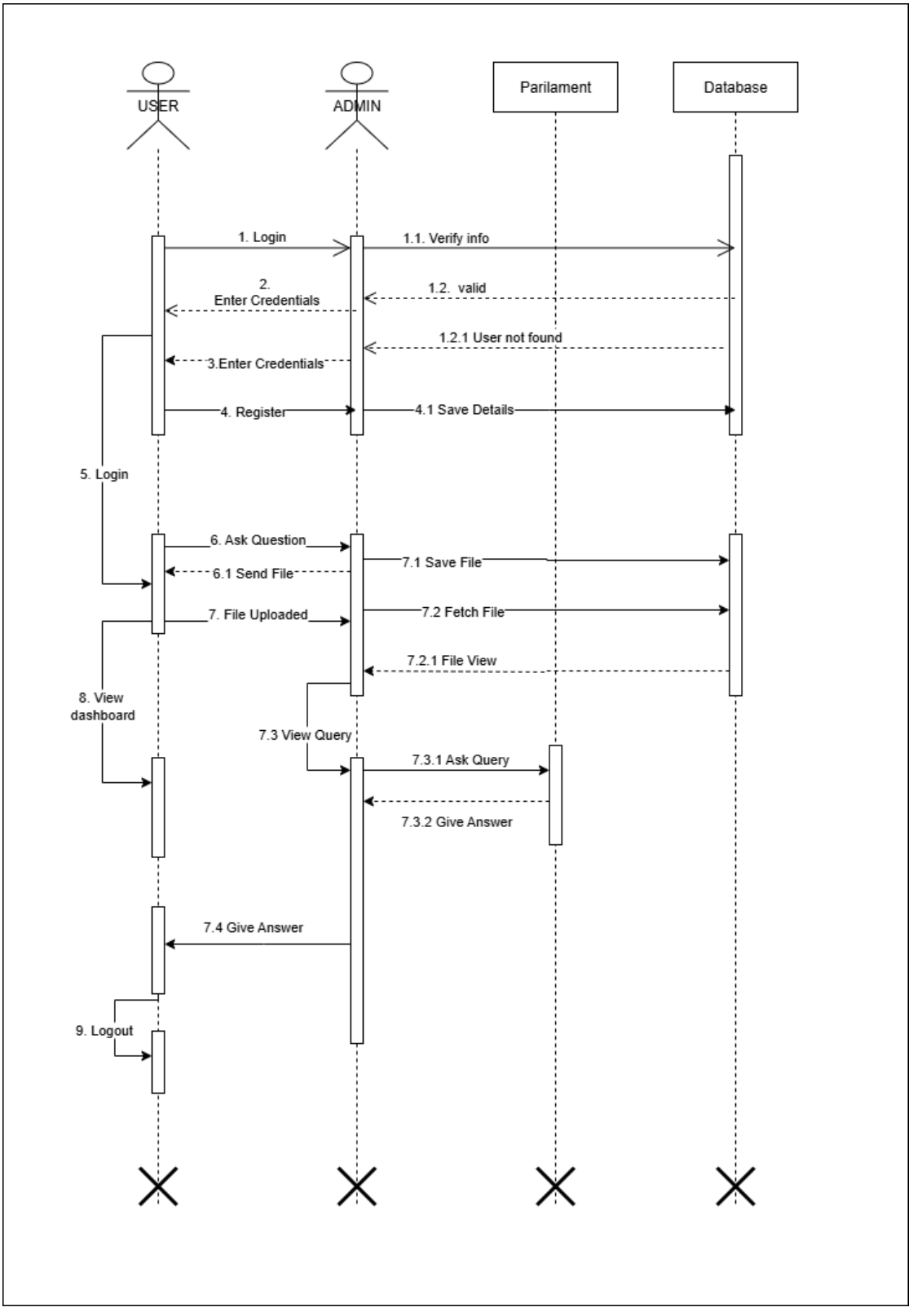

Figure 6.3: Sequence Diagram

6.4 Class Diagram

Abstract: The Class Diagram defines the static structure of the system by showing its main classes, their attributes, methods, and relationships. Each class encapsulates specific data and operations (e.g., Users can login, ask questions; Admin can manage users and upload answers).
This provides the blueprint for the system's object-oriented design.

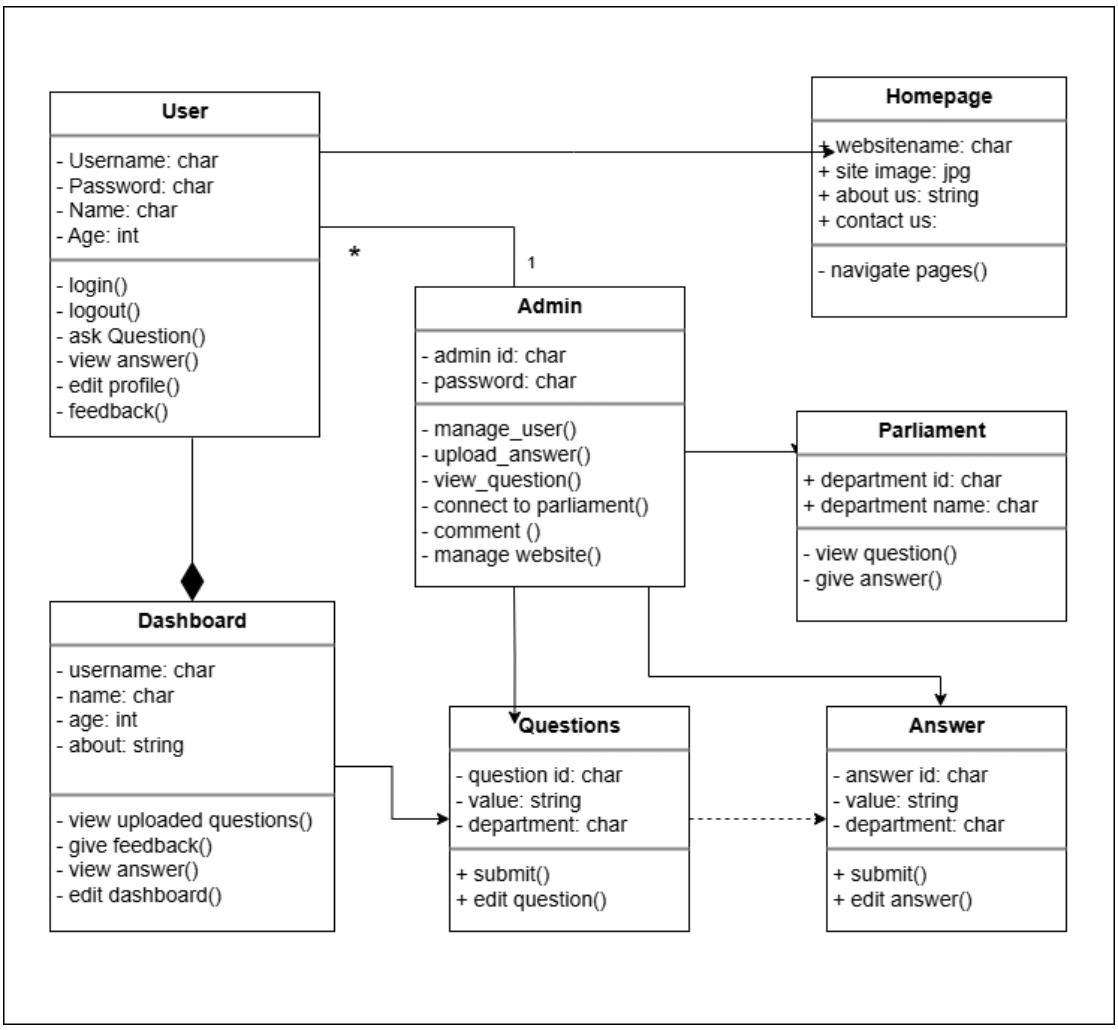

Figure 6.4: Class Diagram