Software Requirements Specification

**(SRS) for** **Hospital Appointment System**

**Version 1.0 Approved**

**Prepared by**

Sampanna Ghimire Priyanka Pokharel

Orchid International College

**Table of Contents**

1. **Introduction**
   1. [Purpose 1](#_TOC_250035)
   2. [Document Conventions 1](#_TOC_250034)
   3. [Intended Audience and Reading Suggestions 1](#_TOC_250033)
   4. [Product Scope 2](#_TOC_250032)
   5. [References 2](#_TOC_250031)
2. [Overall Description](#_TOC_250030)
   1. [Product Perspective 3](#_TOC_250029)
   2. [Product Functions 3](#_TOC_250028)
   3. [User Classes and Characteristics 3](#_TOC_250027)
   4. [Operating Environment 4](#_TOC_250026)
   5. [Design and Implementation Constraints 4](#_TOC_250025)
   6. [User Documentation 4](#_TOC_250024)
   7. [Assumptions and Dependencies 4](#_TOC_250023)
3. [External Interface Requirements](#_TOC_250022)
   1. [User Interfaces 5](#_TOC_250021)
   2. [Hardware Interfaces 5](#_TOC_250020)
   3. [Software Interfaces 5](#_TOC_250019)
   4. [Communications Interfaces 5](#_TOC_250018)
4. [System Features](#_TOC_250017)
   1. [Patient Registration and Login 6](#_TOC_250016)
   2. [Doctor Profile Management 6](#_TOC_250015)
   3. [Appointment Scheduling 6](#_TOC_250014)
   4. [Appointment Reminder System 7](#_TOC_250013)
   5. [Medical Records Management 7](#_TOC_250012)
   6. [Billing and Payment Integration 7](#_TOC_250011)
5. [Other Nonfunctional Requirements](#_TOC_250010)
   1. [Performance Requirements 8](#_TOC_250009)
   2. [Safety Requirements 8](#_TOC_250008)
   3. [Security Requirements 8](#_TOC_250007)
   4. [Software Quality Attributes 8](#_TOC_250006)
   5. [Business Rules 8](#_TOC_250005)
6. [Other Requirements](#_TOC_250004)
   1. [Future Enhancements 9](#_TOC_250003)

[Appendix A: Glossary 9](#_TOC_250002)

[Appendix B: Analysis Models 10-15](#_TOC_250001)

[Appendix C: To Be Determined List 16](#_TOC_250000)

1. Introduction

1.1 Purpose

This document describes the requirements for the Online Polling System, which allows users to create polls, vote on them, and view results in real time. The system will act as a platform for gathering feedback and opinions through easy-to-use polls. It will cover both basic and advanced features, including secure user registration and a reliable polling process.

1.2 Document Conventions

This document uses the following formats:

• Bold for system features and key terms.

• Italic for non-functional requirements.

• Numbered lists for steps and requirements.

1.3 Intended Audience and Reading Suggestions

This SRS is written for:

• Developers to understand the features and constraints of the system.

• Project Managers to track the project’s progress and ensure all requirements are met.

• Testers to create and run test cases based on the requirements.

• Stakeholders to verify the system meets their needs.

It's best to start with Section 2 (Overall Description) for a general overview, and then move on to Section 4 (System Features) for details about each feature.

1.4 Product Scope

The Online Polling System allows users to create and participate in polls across categories like sports, politics, and entertainment. Users can vote anonymously, see real-time results, and view trends through charts and graphs. Admins will have control over managing users, polls, and content. This system helps gather public opinion and insights for better decision-making in businesses, organizations, and for individuals.

2. Overall Description

2.1 Product Perspective

The Online Polling System is a standalone product designed to facilitate the creation, management, and participation in online polls. This system is not part of any existing family of products but is built as a self-contained application with both basic and advanced features. It allows users to create polls, vote anonymously, and view results in real-time. For advanced features, it includes graphical data visualization and user management by admins. This system is meant to streamline opinion collection, whether for small teams or larger organizations. It interfaces with external databases for storing polls, votes, and user data, and can integrate third-party tools for data visualization (e.g., Chart.js).

2.2 Product Functions

The Online Polling System must perform the following core functions:

User Registration and Login: Allow users to sign up, log in, and securely access the platform.

Poll Creation: Let users create customizable polls with various question types and options.

Voting Mechanism: Allow users to vote on polls, with support for anonymous voting.

View Results: Display poll results in real-time.

Advanced Features: Include poll categorization, graphical reports (using Chart.js), and administrative control for managing polls and users.

2.3 User Classes and Characteristics

There are three main types of users for this system:

Admin Users: Admins can manage the overall system, including creating polls, managing users, and viewing detailed analytics. They require advanced access to features such as data visualization and system monitoring.

Registered Users: These users can create, participate in, and manage their own polls. They must register for the platform and can view results of the polls they participate in.

Anonymous Users: These users can vote on polls without logging in or revealing their identity. They can only view results once they have voted but cannot create or manage polls.

2.4 Operating Environment

The Online Polling System will be accessible through modern web browsers on any device (desktops, tablets, smartphones). It will be deployed on cloud-based servers, making it available across various operating systems like Windows, macOS, Linux, and iOS/Android for mobile devices. The system will require:

Browser compatibility: Chrome, Firefox, Safari, and Edge (latest versions).

Database: SQL or NoSQL databases for data storage.

Backend: The server will run on any standard environment supporting Node.js or PHP.

2.5 Design and Implementation Constraints

The following constraints apply:

Security: All user data and votes must be encrypted, and secure login must be implemented using industry standards such as OAuth2 or JWT.

Database Performance: The system must handle concurrent votes and polls, with a focus on low-latency response.

Technology: The system will use Chart.js for graphical representations, and any external data visualization libraries should be compatible.

Mobile-Friendly: The user interface must be responsive to different screen sizes for an optimal experience on mobile devices.

2.6 User Documentation

The following documentation will be provided for users:

User Manuals: Detailed instructions on how to register, create polls, vote, and view results.

Online Help: A help section within the platform that includes FAQs and step-by-step guides.

Tutorial Videos: Short videos demonstrating common tasks such as poll creation and viewing analytics.

2.7 Assumptions and Dependencies

The Online Polling System assumes that users have access to a stable internet connection and modern web browsers. Key dependencies include:

Third-party libraries: Dependencies like Chart.js for graphical displays, which may impact the system if they are updated or deprecated.

Database Availability: The system depends on a robust backend database to store poll data, votes, and user information.

Cloud Infrastructure: The system relies on cloud hosting, and any downtime or maintenance of the hosting provider could affect system availability.

3. External Interface Requirements

3.1 User Interfaces

The Online Polling System will have a clean, user-friendly interface accessible via web browsers. The user interface will follow standard web design practices, ensuring consistency and ease of navigation. Key interface elements include:

* Home Page: Contains links to view and create polls.
* Poll Creation Page: A form where users can input poll details like the question, options, and settings (e.g., categories, start/end times).
* Voting Page: Users can select their choice and submit their vote.
* Results Page: Displays real-time poll results in both numerical and graphical (charts) formats.
* Admin Dashboard: Allows admins to manage polls, users, and view detailed analytics.

Standard buttons and features include:

* Submit/Cancel Buttons: For actions like submitting a vote or creating a poll.
* Help Button: Provides access to FAQs and user guides.
* Error Messages: Display in red when a user attempts an invalid action, such as submitting a poll with incomplete details.

3.2 Hardware Interfaces

The system will interact with the hardware mainly through user devices (e.g., desktop computers, tablets, smartphones). No specialized hardware is required, and it will support:

* Desktop and Laptop Computers: The system will be compatible with general input devices like keyboard and mouse.
* Mobile Devices: The system will support touch input on mobile devices like smartphones and tablets.

Communication between hardware and the software will occur over standard web protocols, making the system platform-independent.

3.3 Software Interfaces

The Online Polling System will interact with the following software components:

* Database Management System (DBMS): The system will connect to either a SQL or NoSQL database for storing poll data, user information, and votes.
* Operating System: The system will be compatible with major operating systems such as Windows, macOS, Linux, iOS, and Android.
* Web Browser: The system will be accessible through any modern browser (e.g., Chrome, Firefox, Safari, Edge).
* Libraries: Chart.js will be used for displaying real-time poll results in graphical formats. The system may also integrate other third-party libraries for additional features.

Communication with the database will be done through RESTful APIs to ensure smooth data transfer between the frontend and backend.

3.4 Communications Interfaces

The system will use standard web-based communication protocols to function, including:

* HTTP/HTTPS: The primary protocol for data communication between users and the server, ensuring secure data transfer with SSL/TLS encryption for HTTPS.
* Email Notifications: The system will send email notifications for user registration, password resets, or when a poll is closed or results are published. These communications will use SMTP protocols.
* WebSockets: For real-time updates on poll results, ensuring that the data is dynamically updated without the need for page refreshes.
* JSON: The data format used for sending requests and responses between the frontend and backend.

4. System Features

4.1 Poll Creation

4.1.1 Description and Priority

This feature allows users to create polls by entering a question and multiple options for voting. It includes the ability to set the poll's category (e.g., sports, politics) and its duration (open/close dates). This feature is critical to the system's functionality, as it forms the foundation of the polling system.

* Priority: High
* Benefit: 9
* Penalty for not having: 9
* Cost to implement: 5
* Risk if not implemented correctly: 6

4.1.2 Stimulus/Response Sequences

* User Action: The user logs in and clicks on the "Create Poll" button.
* System Response: The system opens a form where the user can enter the poll question, options, and other settings.
* User Action: The user fills in the poll details and submits the form.
* System Response: The system saves the poll in the database and confirms the poll has been created successfully. The poll is now visible to other users.

4.1.3 Functional Requirements

* REQ-1: The system shall provide a form for users to create a poll with a title, description, and multiple voting options.
* REQ-2: The system shall allow users to select the poll's category from a predefined list (e.g., Sports, Politics, etc.).
* REQ-3: The system shall allow users to set the start and end date/time for when the poll will be open for voting.
* REQ-4: The system shall ensure that only registered users can create polls.
* REQ-5: The system shall validate poll inputs to prevent empty fields or duplicate options.
* REQ-6: The system shall provide a success message after poll creation, with a link to view the newly created poll.
* REQ-7: If the poll creation fails due to invalid input or server issues, the system shall display a clear error message explaining the failure.

4.2 Voting Mechanism

4.2.1 Description and Priority

This feature allows users to vote on existing polls by selecting one of the available options. It is essential to the system as it provides the core functionality of collecting user opinions.

* Priority: High
* Benefit: 10
* Penalty for not having: 10
* Cost to implement: 4
* Risk if not implemented correctly: 7

4.2.2 Stimulus/Response Sequences

* User Action: The user views a poll and selects an option.
* System Response: The system records the vote in the database and updates the poll results in real time.

4.2.3 Functional Requirements

* REQ-1: The system shall allow both registered and anonymous users to vote on a poll.
* REQ-2: The system shall ensure that each user can only vote once on a given poll.
* REQ-3: The system shall store votes in the database and update results in real time.
* REQ-4: The system shall display a confirmation message once the vote is submitted.
* REQ-5: If the user has already voted, the system shall prevent duplicate votes and notify the user.
* REQ-6: The system shall allow admins to enable or disable anonymous voting on a poll.

4.3 View Results

4.3.1 Description and Priority

This feature allows users to view the real-time results of a poll. Users can see how many votes each option has received and view the data in a graphical format (e.g., bar chart or pie chart).

* Priority: Medium
* Benefit: 7
* Penalty for not having: 6
* Cost to implement: 4
* Risk if not implemented correctly: 3

4.3.2 Stimulus/Response Sequences

* User Action: The user clicks on the "View Results" button for a poll.
* System Response: The system fetches the current results from the database and displays them as numbers and charts.

4.3.3 Functional Requirements

* REQ-1: The system shall allow users to view poll results in real time after voting.
* REQ-2: The system shall display results using charts (e.g., bar charts or pie charts) created with Chart.js.
* REQ-3: The system shall allow users to toggle between viewing results as raw numbers or graphical charts.
* REQ-4: The system shall update the displayed results dynamically without requiring the user to refresh the page.
* REQ-5: The system shall ensure only registered users can view detailed poll results (e.g., demographics or trends).

4.4 Admin Control

4.4.1 Description and Priority

This feature provides admin users with tools to manage the polls and users, including deleting inappropriate polls or users, viewing detailed analytics, and generating reports.

* Priority: High
* Benefit: 9
* Penalty for not having: 8
* Cost to implement: 6
* Risk if not implemented correctly: 4

4.4.2 Stimulus/Response Sequences

* User Action: The admin logs in and accesses the admin dashboard.
* System Response: The system provides options for managing polls, viewing reports, and handling user accounts.

4.4.3 Functional Requirements

* REQ-1: The system shall provide a dashboard for admin users to manage polls, including deleting or editing existing polls.
* REQ-2: The system shall allow admin users to view detailed poll analytics, including user participation and vote trends.
* REQ-3: The system shall enable admin users to delete inappropriate polls or block users who violate the platform's rules.
* REQ-4: The system shall generate reports summarizing poll activity for admin users.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The Online Polling System must perform efficiently even during peak usage. This includes:

The system should be able to handle up to 10,000 concurrent users voting without delays.

Results should be updated in real-time within 2 seconds after a vote is cast.

The system should respond to user actions, such as submitting a vote or creating a poll, within 1 second.

5.2 Safety Requirements

The system should ensure that no harm or loss can occur from its usage. Important safety aspects include:

Users must not lose their data during unexpected crashes or server failures. Automatic backups will be performed daily.

Admins should have the ability to recover accidentally deleted polls or votes from backup storage.

5.3 Security Requirements

To protect user data and maintain privacy:

All user data must be encrypted, especially personal information like passwords.

User authentication will be done via secure methods such as OAuth2 or JWT.

Only registered users should be able to create polls, and only admin users can delete polls.

The system should prevent multiple votes from the same user on a single poll to ensure the accuracy of results.

5.4 Software Quality Attributes

The system must meet the following quality standards:

Usability: The interface should be simple and easy for all users, even for those with limited technical skills.

Reliability: The system should have 99.9% uptime, ensuring it's always available for users to create or vote on polls.

Maintainability: The codebase should be modular, making it easy for developers to fix bugs or add new features.

Portability: The system should work seamlessly across devices, including desktop browsers, tablets, and smartphones.

5.5 Business Rules

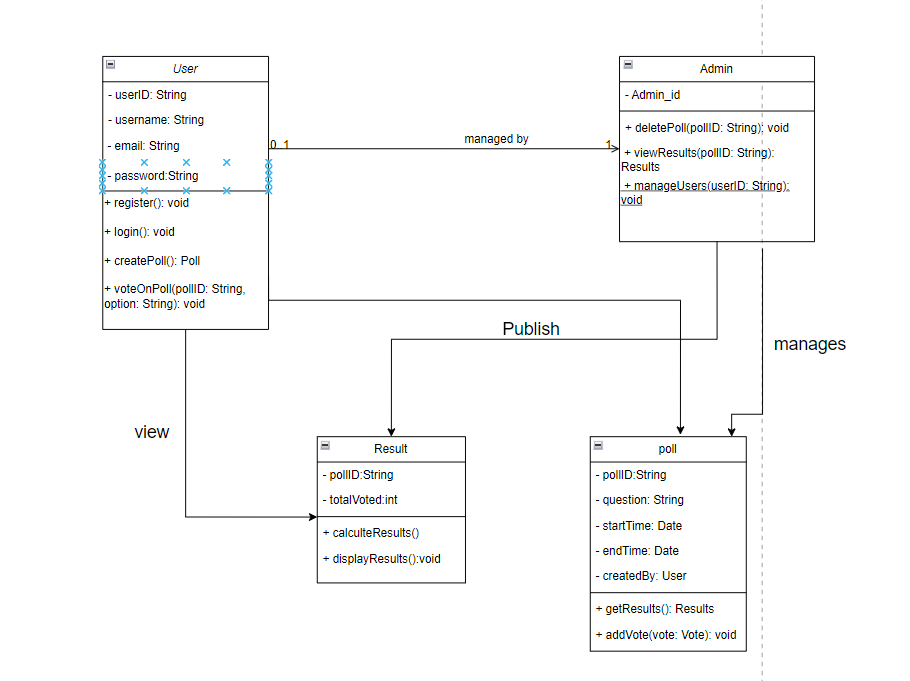
The Online Polling System will follow these rules:

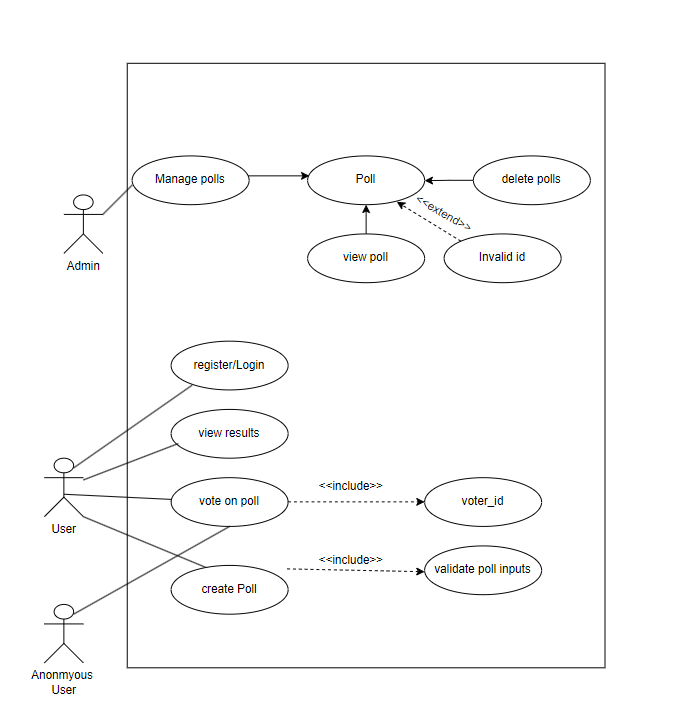
Admins can create, edit, or delete any poll.

Regular users can only create and manage their own polls.

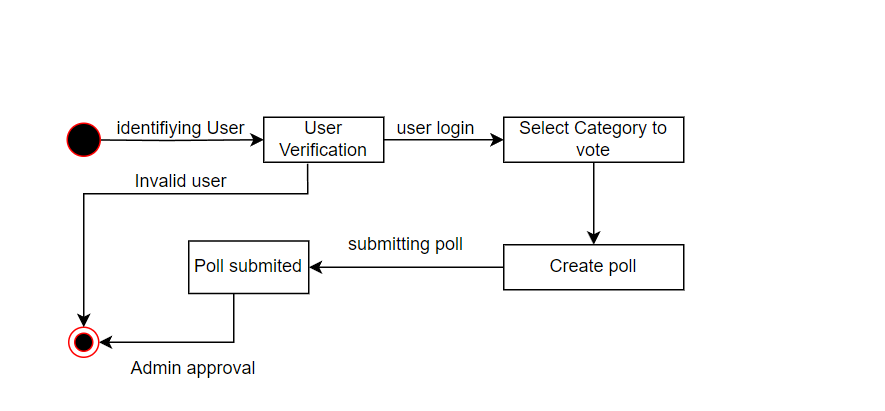
Only one vote per user is allowed per poll to ensure fair voting.

Appendix B:

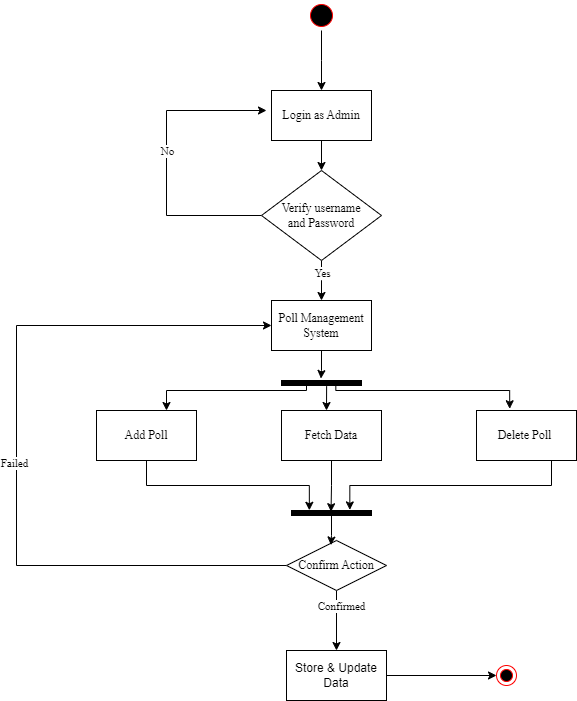


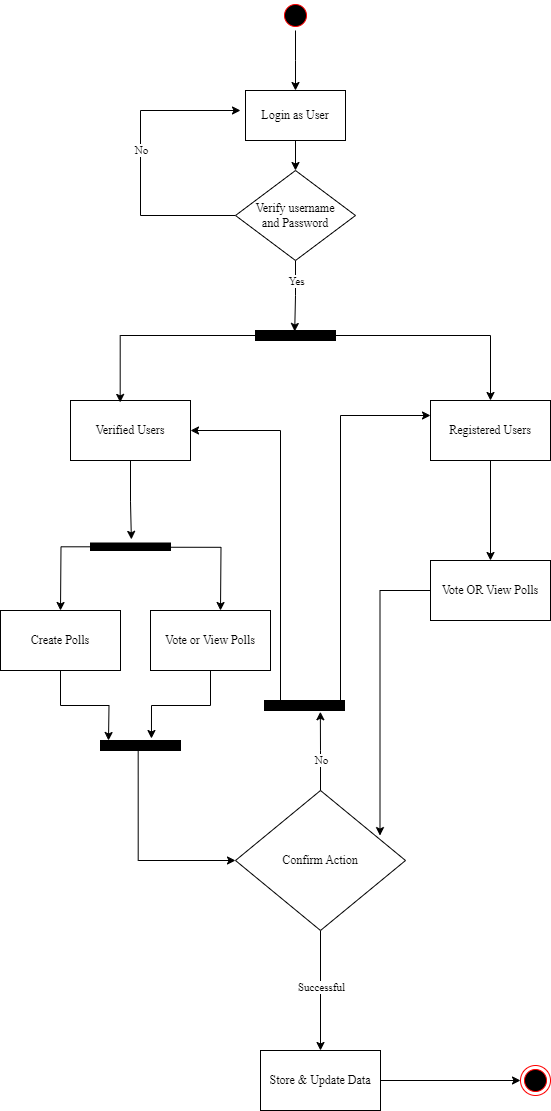
Use Case Diagram

State Diagram

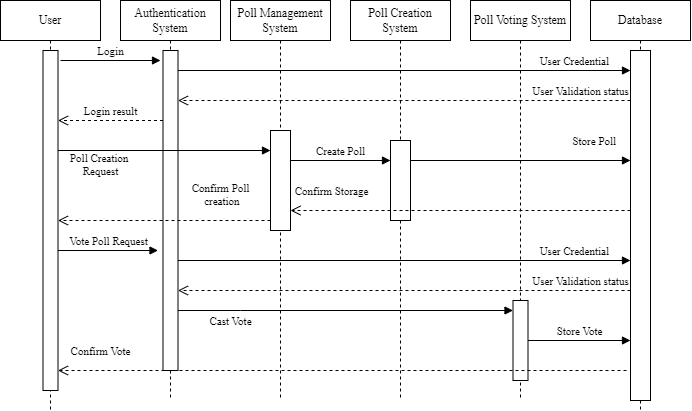


Admin Activity Diagram:

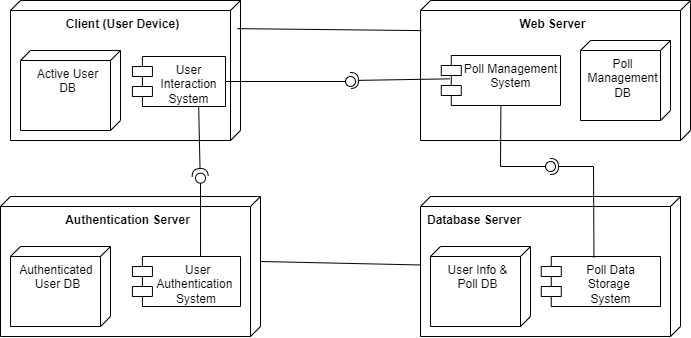




Sequence Diagram:



Deployment Diagram:



Component Diagram:

