

Software Requirement Specification Document for Classmate Project

September 11, 2023

Table 1: Document version history

| Version | Date | Reason for Change |
|---------|-------------|---|
| 1.0 | 11-Sep-2023 | SRS First version's specifications are defined. |

Project: Classmate

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 3 |
| 1.1 | Purpose of this Document | 3 |
| 1.2 | Scope of this Document | 3 |
| 2 | Similar Systems | 4 |
| 2.1 | Academic Systems | 4 |
| 2.2 | Business Systems | 5 |
| 3 | System Description | 5 |
| 3.1 | Problem Statement | 5 |
| 3.2 | System Overview | 5 |
| 3.3 | Key Features | 6 |
| 3.3.1 | Student Key Features: | 6 |
| 3.3.2 | Center Key Features: | 6 |
| 3.4 | System Scope | 6 |
| 3.5 | System Context | 6 |
| 3.5.1 | Users | 7 |
| 3.5.2 | External Systems | 7 |
| 3.5.3 | Interfaces | 7 |
| 3.6 | System Objectives | 7 |
| 3.7 | For Students: | 8 |
| 3.8 | For Course Centers: | 8 |
| 4 | Functional Requirements | 9 |
| 4.1 | System Functions | 9 |
| 5 | Design Constraints for Mobile App with Flutter and Firebase | 10 |
| 5.1 | Standards Compliance for Flutter and Firebase | 11 |
| 5.2 | Hardware Limitations | 12 |
| 6 | Non-Functional Requirements | 12 |
| 7 | Operational Scenarios | 13 |

Abstract

In today's fast-paced educational landscape, effective communication and access to course schedules are critical for students' academic success. To address these needs, we present "Classmate," an integrated application designed to bridge the communication gap between students and course centers. Classmate offers students a user-friendly platform to access static course schedules, search for courses, and engage in direct communication with course centers. This project represents the first phase of a broader vision, with future developments set to revolutionize the educational experience by incorporating dynamic scheduling and interactive features. Classmate strives to enhance efficiency, organization, and transparency in the realm of student-center interactions, fostering a more informed and connected learning environment.

1 Introduction

The Software Requirements Specification (SRS) serves as a comprehensive document outlining the specifications and functional requirements for the development of "Classmate," a cutting-edge course management mobile application. "Classmate" is designed to bridge the communication gap between students and course centers, providing an efficient and user-friendly platform for accessing course schedules, instructor information, and course-related communications.

This document is intended to provide a clear and structured blueprint for the development team, stakeholders, and all parties involved in the "Classmate" project. It outlines the system's functionality, performance, and design requirements, ensuring a shared understanding of the project's scope, objectives, and constraints.

"Classmate" is envisioned as a dynamic, mobile-first application that empowers both students and course centers to streamline their interactions, enhance learning experiences, and optimize course management. The application will leverage the Flutter framework for the front end to provide a cross-platform, user-centric interface, while Firebase will serve as the back end to facilitate real-time data synchronization and secure user authentication.

1.1 Purpose of this Document

The primary purpose of this Software Requirements Specification (SRS) document is to provide a comprehensive and well-structured guide that defines the functional and non-functional requirements for the development of the "Classmate" course management mobile application. This document serves as a critical communication tool among project stakeholders, including developers, designers, testers, project managers, and clients.

1.2 Scope of this Document

This SRS document encompasses the following aspects related to our project:

- **Daily Functions Enhancement:** The document aims to outline how our project will improve daily operations, such as course registration and scheduling, within educational centers. With the implementation of the "Classmate" course management mobile application, these functions can be performed more accurately and efficiently. The user-friendly modules of the application ensure easy access to essential features.

- **Non-Functional Requirements:** This document defines various non-functional requirements that are critical for the successful operation of the "Classmate" application. These requirements include aspects such as confidentiality, data integrity, system efficiency, user privacy, system reliability, safety, security, traceability, and usability. Ensuring compliance with these non-functional requirements is vital for achieving a high-quality user experience.
- **Project Deliverables:** The SRS document also outlines the final deliverables that will result from the project's completion. These deliverables include a comprehensive testing document, a detailed software design document, the Software Requirements Specification (SRS) document itself, and the fully developed and functional "Classmate" course management mobile application.

This SRS document serves as a comprehensive guide and reference for the development, testing, and deployment of the "Classmate" course management mobile application. It ensures that the project aligns with its objectives and meets the expectations of the identified stakeholders.

2 Similar Systems

2.1 Academic Systems

In the academic domain, there are existing systems and platforms that share similarities with "Classmate." These systems serve as points of reference and provide insights into the functionality and features that have been implemented successfully in similar contexts. Notable systems in this category include:

1. **Course Management Systems (CMS):** Many educational institutions utilize CMS platforms like Moodle, Blackboard, and Canvas to manage course content, schedules, and communication. These systems offer features for instructors to upload course materials, students to access resources, and facilitate online discussions. "Classmate" draws inspiration from these systems in terms of course management but aims to provide a more user-friendly and mobile-centric experience.
2. **Communication Apps for Education:** Applications like Microsoft Teams and Slack have gained popularity in the education sector for their communication and collaboration features. They allow students and instructors to communicate in real-time, share files, and organize discussions. "Classmate" incorporates similar communication functionalities while tailoring them to the specific needs of course centers and students.
3. **Student Information Systems (SIS):** Student Information Systems like PowerSchool and Infinite Campus focus on student data management, including enrollment, grades, and attendance. While "Classmate" primarily emphasizes course management and communication, it can complement SIS platforms by enhancing the student experience and engagement.
4. **Educational Mobile Apps:** Several mobile applications, such as edX, Coursera, and Duolingo, offer course enrollment, progress tracking, and content delivery on mobile devices. "Classmate" shares similarities with these apps in terms of mobile accessibility and user engagement, but its primary focus is on facilitating communication between students and course centers.

2.2 Business Systems

In the business domain, there are also systems that exhibit resemblances to "Classmate" in terms of communication and information sharing. These systems provide insights into effective communication and collaboration tools. Examples include:

1. **Communication and Collaboration Tools:** Business communication and collaboration platforms like Slack, Microsoft Teams, and Zoom are widely used for real-time messaging, video conferencing, and document sharing. "Classmate" incorporates similar features to foster collaboration among students and course centers but tailors them to the educational context.

2. **Customer Relationship Management (CRM) Systems:** CRM systems like Salesforce and HubSpot help businesses manage customer interactions and communications. "Classmate" leverages communication functionalities inspired by CRM systems to enhance interactions between course centers and students.

3. **Project Management Tools:** Project management tools like Asana and Trello facilitate task tracking and team collaboration. "Classmate" integrates features inspired by these tools to enable efficient course scheduling and management.

It is important to note that while "Classmate" draws inspiration from these similar systems in both academic and business domains, it aims to offer a unique combination of features tailored specifically to the course management and communication needs of students and course centers. By leveraging the best practices and functionalities of these systems, "Classmate" strives to provide a user-centric, efficient, and innovative solution within the educational ecosystem.

3 System Description

3.1 Problem Statement

- **Limited Data Accessibility** Users face difficulties in accessing necessary data. Information retrieval is cumbersome and time-consuming.
- **Data Disorganization** related to courses and schedules lacks organization. Users find it challenging to locate specific information efficiently.
- **Instructor Information Accessibility** Students struggle to find essential instructor details, such as contact information, office hours, and teaching history. The absence of accessible instructor information hinders students' academic engagement.
- **Communication Gap** There is no reliable, centralized communication channel between students and course centers. This results in missed opportunities for students to seek guidance, clarify doubts, and stay informed about course-related matters. Effective communication between stakeholders is lacking.

3.2 System Overview

The Classmate application is designed to address the pressing challenges faced by students and course centers in managing course-related information and communication. Classmate offers a comprehensive solution by providing easy access to course schedules, organizing data effectively,

ensuring instructor information accessibility, and bridging the communication gap between students and course centers.

3.3 Key Features

3.3.1 Student Key Features:

- **Course Data Access:** Students can access comprehensive information about courses offered by each center, including schedules, descriptions, and registration details.
- **Instructor Tracking:** Students have the capability to follow and keep track of their preferred instructors, including access to contact details, office hours, and teaching history.
- **Course Date Notifications:** The system notifies students promptly when there are updates or changes in course dates, ensuring they stay informed.
- **Session Requests:** Students can request additional sessions or specific course-related sessions as needed to enhance their learning experience.

3.3.2 Center Key Features:

- **Course and Instructor Management:** Course centers have the authority to modify and update course and instructor data, ensuring information accuracy and relevance.
- **Statistical Insights:** Centers can access statistics on the most viewed courses and instructors, enabling data-driven decisions and improvements.
- **Demand-Driven Course Addition:** Centers can add new courses based on demand and market trends, ensuring a flexible and responsive course offering.

These key features cater to the distinct needs of both students and course centers, fostering a more engaging and efficient educational ecosystem.

3.4 System Scope

With the installed clinic software, daily tasks such as patient registration and doctor program management can be done more accurately and easily. The modules of the clinic management software are user-friendly and easy to use.[?]

3.5 System Context

The "Classmate" course management mobile application operates within a dynamic system context that encompasses various elements, including users, external systems, and interfaces. This section provides an overview of the broader context in which "Classmate" functions:

3.5.1 Users

- **Students:** Students are the primary users of the "Classmate" application. They use the app to access course schedules, track instructors, request additional sessions, and engage in communication with course centers.
- **Course Centers:** Course centers are educational institutions or organizations that offer courses. They use "Classmate" to manage course data, instructors, and communicate with students.
- **Instructors:** Instructors are educators who teach courses at course centers. They utilize "Classmate" to update course information, share schedules, and interact with students.
- **Administrators:** Administrators have administrative privileges to manage and oversee the application. They can monitor user activity, resolve disputes, and ensure smooth operation.

3.5.2 External Systems

- **Firebase Backend:** "Classmate" interfaces with Firebase, a cloud-based backend service, to store and retrieve data, manage user authentication, and facilitate real-time communication between users.
- **Third-Party APIs:** The application may integrate with third-party APIs for services like location services, notification delivery, or analytics.

3.5.3 Interfaces

- **User Interface (UI):** The UI of "Classmate" is designed to be intuitive and user-friendly, providing students, course centers, and instructors with easy navigation and access to essential features.
- **Firebase Integration:** "Classmate" communicates with Firebase through APIs and SDKs to perform tasks such as data retrieval, authentication, and real-time updates.
- **External Services:** Interfaces with third-party services may include APIs for geolocation, push notifications, or analytics services to enhance the functionality of the application.

The system context of "Classmate" reflects a network of interactions among users, external systems, and interfaces. Understanding this context is essential for developing and maintaining an application that effectively meets the needs of students, course centers, and instructors while ensuring seamless integration with external systems and services.

3.6 System Objectives

The objectives of the system, named "Classmate," are designed to address the core needs and goals of both students and course centers, fostering effective communication, streamlined access to information, and a more organized educational experience. The system aims to achieve the following key objectives:

3.7 For Students:

1. **Enhanced Course Access:** Provide students with seamless access to comprehensive information about courses, including schedules, descriptions, and registration details, ensuring they can make well-informed decisions about their academic journey.
2. **Improved Instructor Interaction:** Enable students to follow and keep track of their preferred instructors, access contact details, office hours, and teaching history, fostering better communication and engagement with educators.
3. **Real-Time Notifications:** Notify students promptly of any updates or changes in course dates, ensuring they stay informed and can adapt their schedules accordingly.
4. **Flexible Learning:** Allow students to request additional sessions or specific course-related sessions, empowering them to tailor their learning experience to their individual needs and preferences.

3.8 For Course Centers:

1. **Efficient Data Management:** Provide course centers with the tools to modify and update course and instructor data, ensuring that information is accurate, up-to-date, and relevant to the evolving educational landscape.
2. **Data-Driven Decisions:** Enable course centers to access statistics on the most viewed courses and instructors, allowing them to make data-driven decisions, refine course offerings, and optimize their educational programs.
3. **Responsive Course Offerings:** Empower course centers to add new courses based on demand and market trends, ensuring that their course catalog remains flexible and responsive to the changing needs of students and the industry.

By fulfilling these objectives, Classmate aims to create a user-centric educational ecosystem that promotes efficient communication, accessibility to course-related data, and a more tailored learning experience for students while offering course centers the tools they need to adapt and thrive in the dynamic educational landscape.

4 Functional Requirements

4.1 System Functions

In this section, we outline the functional requirements for the development of our course management system, tentatively named "Classmate." These requirements define the specific functionalities and features that the system must possess to meet the needs of both students and course centers effectively.

The functional requirements serve as the foundation for the system's design, development, and testing phases. They provide a clear road map for the development team, ensuring that the system aligns with the objectives and expectations outlined in the Software Requirements Specification (SRS).

The requirements are divided into two main categories: those aimed at addressing the needs of students and those aimed at fulfilling the requirements of course centers. These functional requirements encompass a range of capabilities, from data access and communication to analytics and administrative tasks.

By adhering to these functional requirements, we aim to create a robust and user-centric course management system that simplifies the educational experience for students, empowers course centers, and fosters efficient communication within the educational ecosystem.

- Student:
 - **View All Centers:** Students can access a list of all available centers.
 - **View Courses by Center:** Students can view the courses available at each center.
 - **View Instructors by Course:** Students can see the instructors available for each course.
 - **Follow Instructors:** Students have the option to follow specific instructors and receive notifications when they add a new session.
 - **Follow Courses:** Students can follow particular courses and receive notifications when new sessions are added.
 - **Request Course Addition or Revision:** Students can send requests to any center to propose the addition of specific courses or revisions to existing ones.
- Center
 - **Add New Session:** Centers have the capability to create and add new sessions for courses they offer.
 - **Add New Instructor:** Centers can include new instructors to their roster, expanding the teaching staff.
 - **Edit Session Details:** Centers can modify session data, including time and location, to keep it up-to-date.
 - **Receive Course/Revision Requests:** Centers can track the number of applications received requesting the addition of specific courses or revisions to existing ones.

- **Analytic Dashboard:** Centers have access to an analytics dashboard to gain insights into various aspects of their courses and operations.
- **Schedule Polling:** Centers can conduct polls to gather feedback and preferences from students regarding the timing or date of a course.

5 Design Constraints for Mobile App with Flutter and Firebase

Developing our course management mobile app, "Classmate," with Flutter for the frontend and Firebase for the backend introduces specific design constraints that guide the architectural decisions and operational considerations for mobile platforms. These constraints ensure the stability, security, and scalability of the mobile application while leveraging the capabilities of this technology stack. Below, we detail the key design constraints associated with this mobile app:

- **Technology Stack Compatibility:** Classmate must be designed as a mobile application that utilizes Flutter for the frontend and Firebase for the backend. The mobile app should deliver a seamless and responsive user experience on various mobile devices.
- **Firebase Security:** Given Firebase's role as the backend, security is paramount for the mobile app. The design must encompass Firebase security rules, authentication mechanisms, and data encryption to protect user information and maintain data integrity.
- **Scalability with Firebase:** Firebase's scalability features should be leveraged to ensure that the mobile app can handle increased user loads and data volumes as the user base expands. Firestore's real-time database capabilities are vital for dynamic content updates.
- **Firebase Integration:** Smooth integration with Firebase services, such as Firestore for real-time data storage and Firebase Authentication for user management, should be a focal point in the design to ensure seamless and secure backend operations.
- **Mobile Platform Consistency:** The mobile app's design must prioritize a consistent user interface, features, and functionality across various mobile platforms (e.g., iOS and Android) to provide a unified user experience.
- **Cost Management:** Effective cost management within Firebase should be considered. Design choices must optimize Firebase usage to align with the project's budget constraints and resource allocation for mobile app development.
- **Development Timeline:** The design must accommodate a well-defined development timeline for the mobile app, accounting for the unique considerations of mobile app development using Flutter and Firebase.
- **User Training and Onboarding:** Considerations for user training and onboarding should be incorporated into the mobile app's design to ensure that users can effectively navigate and utilize the Classmate mobile app.

Adhering to these design constraints specific to the mobile app with the Flutter-Firebase architecture is pivotal in creating a robust and user-friendly course management mobile application that meets the educational ecosystem's demands while efficiently managing costs and resources.

Adhering to these design constraints specific to the Flutter-Firebase architecture is pivotal in creating a robust and user-friendly course management solution that meets the educational ecosystem's demands while efficiently managing costs and resources.

5.1 Standards Compliance for Flutter and Firebase

Ensuring standards compliance is essential when utilizing Flutter for the frontend and Firebase for the backend in the development of our course management system, "Classmate." Adhering to relevant industry and educational standards is crucial to maintain data integrity, security, and interoperability within this specific technology stack. Below, we outline the key standards that apply to our Flutter-Firebase architecture:

- **Data Protection Standards:** Classmate must comply with data protection regulations and standards, even more so when utilizing Firebase as the backend. This includes adherence to GDPR (General Data Protection Regulation) and other regional data protection laws. Firebase security rules and data encryption should be configured accordingly.
- **Accessibility Standards:** When designing the Flutter frontend, it is vital to follow accessibility guidelines such as WCAG (Web Content Accessibility Guidelines) to ensure that the mobile and web app is usable by individuals with disabilities. This includes providing alternative text for images, keyboard navigation, and screen reader compatibility.
- **Educational Standards:** Classmate should align with educational standards and best practices, leveraging Firebase's capabilities for real-time data synchronization and user authentication. Firebase Firestore and Firebase Authentication, for instance, should be implemented to support effective teaching and learning.
- **Security Standards:** Security protocols and practices should follow industry standards when using Firebase for the backend. Compliance with Firebase security best practices, including authentication rules and database security, is essential. Additionally, adherence to OWASP (Open Web Application Security Project) guidelines remains critical for frontend security.
- **Interoperability Standards:** Flutter's ability to create cross-platform apps and Firebase's flexible integration options should align with standards for data exchange. Ensure that Firebase's support for RESTful APIs and common data formats (e.g., JSON) facilitates interoperability with other systems or services as needed.
- **User Interface Standards:** The Flutter-based frontend should adhere to established design and usability standards to provide an intuitive and consistent user experience across both mobile and web platforms.

Incorporating these standards into our Flutter-Firebase architecture not only enhances the reliability and security of Classmate but also facilitates seamless data management, collaboration, and user satisfaction within the educational ecosystem.

5.2 Hardware Limitations

Our website should be able to run smoothly on a standard computer with sufficient resources. However, as the website grows and attracts more users, additional hardware resources may be required to maintain optimal performance and functionality. Since the website handles sensitive information like patient medical records. It may require additional hardware resources to ensure that the data is secure and protected from unauthorized access or breaches.

6 Non-Functional Requirements

In addition to the functional requirements, non-functional requirements play a vital role in shaping the design and performance of our course management mobile application, "Classmate." These requirements define the attributes and qualities of the system that contribute to its overall usability, reliability, and scalability. Below, we outline the key non-functional requirements that guide the development of Classmate:

- **Performance:** The mobile application must deliver a responsive and smooth user experience. It should load quickly, provide fast access to information, and handle concurrent user interactions efficiently.
- **Scalability:** Classmate should be designed to accommodate future growth in terms of users, courses, and data. It must scale gracefully to handle increased loads without compromising performance.
- **Usability:** The mobile app should have an intuitive user interface that requires minimal training for users to navigate and utilize its features. Usability testing and user feedback should inform design improvements.
- **Reliability:** Classmate must be highly reliable, with minimal downtime or service interruptions. It should have built-in error handling and recovery mechanisms to ensure uninterrupted service.
- **Security:** Ensuring the security of user data is paramount. Classmate must implement robust security measures, including encryption, authentication, and authorization, to protect user information and maintain privacy.
- **Accessibility:** The mobile app should be accessible to users with disabilities. It should follow accessibility guidelines and provide features like screen reader compatibility and keyboard navigation.
- **Compatibility:** Classmate should be compatible with a range of mobile devices, operating systems (iOS and Android), and screen sizes to reach a broad user base.
- **Adaptive and Responsive Design:** The user interface of Classmate should incorporate adaptive and responsive design principles to ensure optimal viewing and interaction across various devices and screen sizes. This includes fluid layouts, flexible content presentation, and touch-friendly navigation.

- **Data Backup and Recovery:** Regular data backups and recovery mechanisms should be in place to safeguard against data loss and support data restoration in case of unexpected incidents.
- **Response Time:** The system must provide timely responses to user actions, such as retrieving course information, sending notifications, and processing user requests, to maintain a seamless user experience.
- **Cost-Efficiency:** The project should be managed within budget constraints. Development and operational costs, including Firebase usage costs, should be optimized for efficiency.
- **Compliance:** Classmate must comply with all relevant legal and regulatory requirements, including data protection laws, educational standards, and industry-specific regulations.

These non-functional requirements collectively define the performance, reliability, and user experience characteristics of Classmate. By adhering to these requirements, we aim to deliver a mobile application that excels in terms of usability, security, scalability, and adaptive and responsive design while meeting the diverse needs of students and course centers within the educational ecosystem.

7 Operational Scenarios

In the context of our course management mobile application, "Classmate," the following operational scenarios describe how users interact with the app for various purposes:

- **Initial Assumption:** The user opens the Classmate app to explore its services and information before proceeding to book a course.
Description: The user, who is new to the app, begins by browsing through the available courses, centers, and information provided.
Outcome: The user gains a basic understanding of the courses offered, centers, and app functionality, preparing for the course booking process.
- **Normal Scenario:** The user selects their desired course type from the dropdown list (e.g., regular course, special session) and then chooses an appropriate date and time for course registration.
Description: The user, having explored the available courses and centers, decides to proceed with course registration. They select the course type (e.g., regular course, special session) from the provided options and then pick a date and time that fits their schedule.
Outcome: The user successfully registers for the course and receives a confirmation. The course registration details are automatically updated in the system.
- **What Can Go Wrong:** The selected course registration date and time may overlap with existing registrations, leading to potential conflicts.
Description: In some cases, the user's chosen registration time may coincide with existing course registrations of other students.

Resolution: To prevent conflicts and misunderstandings, the system will issue an alert message, informing the user of the scheduling conflict. Additionally, alternative available dates and times will be recommended if feasible.

Outcome: The user is informed about the scheduling conflict and provided with alternative course registration options.

- **System State on Completion:** After successfully registering for a course, the user gains complete access to all app services and is redirected to various sections and menus within the app.

Description: Following the completion of the course registration process, the user is redirected to different sections and menus within the app, allowing them to explore additional services and information.

Outcome: The user has full access to all app services and information, including the ability to review their registered courses and confirmations, ensuring that their course registrations are seamlessly integrated into the system's schedule.

References