Software Requirements Specifications (SRS)

The Binary Bros – "ChatFAU"
Jordan Small
Nico Arcelin
Alirio Da Rocha

September 22, 2024

CEN4010 – Principles of Software Engineering
Fall 2024
Professor Safak Kayikci
Florida Atlantic University

Document Revisions

Version	Description
1.00	Initial commit. The document has been filled with project outline and formatted with MS Word headings for Table of Contents.
1.10	The document has been filled with all necessary information, proofread, and carefully formatted. Terms deemed arbitrarily important were added to the Glossary and references used were cited. Table of Contents updated.

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to provide software requirement specifications (SRS)—both functional and non-functional—regarding ChatFAU, a web-based chat application used to facilitate arbitrary academic discussions. ChatFAU will be developed by considering the best interests of both students and educators, aiming to bridge the gap in their course-relevant communication. This SRS will detail and provide information about the project, including its relevant resources, constraints, requirements, and applicable interfaces. The target audience for this document includes project stakeholders and developers, as it will serve as a comprehensive guide to the project's vision.

1.2 Scope

ChatFAU will launch as a web application with full backend management and availability. Users will be able to create accounts and join relevant study groups or academic categorizations to connect with their peers and discuss their educational goals. User profiles will feature a variety of selectable markup display attributes and opportunities to showcase personal information, merits, and/or affiliations. All chats will be live and updated synchronously with the server, moderated by users with special permissions to ensure a productive and social experience. The primary purpose of ChatFAU will remain in providing an open platform to both students and educators to discuss intricacies of their learning and teaching endeavors.

1.3 Definitions, Acronyms, and Abbreviations

- **Administrator –** someone who can make changes on a system that will affect other users
- **Backend –** the part of a computer system or application that is not directly accessed by the user, typically responsible for storing and manipulating data
- **Caching Server –** a dedicated network server or service acting as a storage
- **ChatFAU –** Chat Florida Atlantic University: our platform can be generalized, but will be launched with a special designation for FAU
- **Database -** a structured set of data held in a computer, especially one that is accessible in various ways.
- **Emojis –** a small digital image or icon used to express an idea or emotion
- **Frontend –** relating to or denoting the part of a computer system or application with which the user interacts directly
- **Git –** distributed version control system that tracks versions of files
- **Ghost Loading –** a term used in web development and user experience design to describe a technique where placeholders or "ghost" elements are displayed on a webpage while the actual content is loading
- **Haptic Feedback –** refers to the use of touch sensations (such as vibrations) to provide information or responses to users through a device
- **IT –** Information Technology: a set of related fields that encompass computer systems, software, programming languages, and data and information processing, and storage
- **Latency -** the delay before a transfer of data begins following an instruction for its transfer
- **Preloading –** refers to the practice of loading resources or data in advance of their actual use to improve performance and user experience

- **RLS –** Row Level Security: lets you filter data and enables access to specific rows in a table based on qualifying user conditions
- **SSO –** Single Sign-on: an authentication scheme that allows a user to log in with a single ID to any of several related, yet independent, software systems
- **SRS –** Software Requirements Specifications: a document that is a standardized framework for writing the requirements of a software system
- **Standard Markup Language -** a standardized system used to define the structure and presentation of text in documents. It uses tags or annotations to specify elements like headings, paragraphs, lists, links, and other structural components of a document
- **TLS -** Transport Layer Security: a cryptographic protocol designed to provide communications security over a computer network
- **UI –** User Interface: how the user and a computer system interact, in particular the use of input devices and software
- **WebSockets -** a computer communications protocol, providing a simultaneous two-way communication channel over a single Transmission Control Protocol connection

1.4 References

Jwmsft. (2024). *Screen Sizes and break points for Responsive Design - Windows Apps*. Screen sizes and break points for responsive design - Windows apps | Microsoft Learn. https://learn.microsoft.com/en-us/windows/apps/design/layout/screen-sizes-and-breakpoints-for-responsive-design

KianNH. (2024). *How Cloudflare works | Cloudflare Learning Paths*. Cloudflare Docs. https://developers.cloudflare.com/learning-paths/get-started/concepts/how-cloudflare-works/

Supabase Docs. (2024, September 17). Supabase Docs. https://supabase.com/docs

1.5 Overview

The remainder of this document will outline any relevant project requirements and features, including functional, non-functional, and external interface requirements. External interface requirements will cover aspects such as the user interface (UI), hardware, software, and communication protocols. Non-functional requirements will address important qualities that the system must possess, such as performance, security, usability, readability, and maintainability. Finally, the functional requirements in this document will provide detailed descriptions of the system’s features, organized into use cases or functional areas.

2. Overall Description

2.1 Product Perspective

ChatFAU will be initially designed as an independent system, offering an organized communication platform exclusively for faculty and students at Florida Atlantic University. As the system evolves and continues to develop, there will be significant opportunities for large system expansion, particularly regarding the implementation of multi-school access. For the purposes of this document, all requirements discussed will pertain solely to this initial design, with the understanding that they will be subject to change with future expansions.

2.2 Product Features

The planned features in this product all have one common aim: to facilitate a social and productive environment focused on academic topics for students and educators. Users of ChatFAU will be able to join academic organizations or categories, such as course identifiers, graduating year, major, department, class sections, and general academic topics. Once registered and affiliated, users will have access to multiple chat rooms for various topics, the ability to send private messages or create relevant groups, and will be encouraged to create a profile showcasing their attributes on the site. Basic chat features will include standard markup language options including bold, italics, underline, strikethrough, and font customization—as well as support for custom emojis. Users that are arbitrarily designated as moderators will have access to moderation tools, including commands such as /mute, /ban, /timeout, /prune, and others.

2.3 User Characteristics

As aforementioned, ChatFAU will be developed keeping the best interest of its users in mind. All users will access the product through the same web-based application, featuring a unified login portal and interface. End users with a verifiable faculty or teaching role will be granted special permission to create organizational structures for their individual classes and sections, which users with a student role can then join. Once granted, these permissions will not require any level of skill from these users, as the infrastructure for these processes is designed to be very user-friendly. Users with a student role will be able to affiliate with these groups but will not have the ability to create ones of their own. Otherwise, they share all the same functionality, features, and access as those with the educator role. Users designated as moderators will receive additional functionalities related specifically to chat moderation.

2.4 Operating Environment

To use ChatFAU, users will need a stable internet connection and access to any device with a web browser installed. Since the product is designed to run on the web, users will be able to enjoy all product features as intended, regardless of their hardware specifications. Regular project updates will ensure that the application remains compatible with evolving web technologies and security standards. All project data and information will be stored on the server, allowing users to log in from multiple locations and start or continue any conversations.

2.5 Design and Implementation Constraints

ChatFAU was designed by a group of students at the university and thus does not have any official endorsement from the institution, which may be perceived as a marketing constraint. As the implementation expands across the school following the product launch, users may become more comfortable trusting the platform with sensitive educational data. Additionally, existing communication platforms—both educational and social—may already have user loyalty, leading to hesitancy in adopting a new platform. Another potential constraint involves server resources. Since it is difficult to predict the initial site volume, resources will be allocated based on estimations and may need to be adjusted as the userbase grows.

2.6 Assumptions and Dependencies

For ChatFAU to be widely implemented and functionally efficient, its success depends on adoption by educational institutions and their members. Without an active and engaged user base, other the product’s potential benefits may not be realized. The system has been designed with the assumption that users will uphold academic integrity when using the platform, intentionally ensuring data privacy and protection. Additionally, the platform’s effectives relies on server stability, as a key project goal is to keep data readily accessible. Another constraint involves user training and support; effective use of the platform may require comprehensive onboarding and ongoing support to help users fully utilize its features and address any issues that arise.

3. Specific Requirements

3.1 Functional Requirements

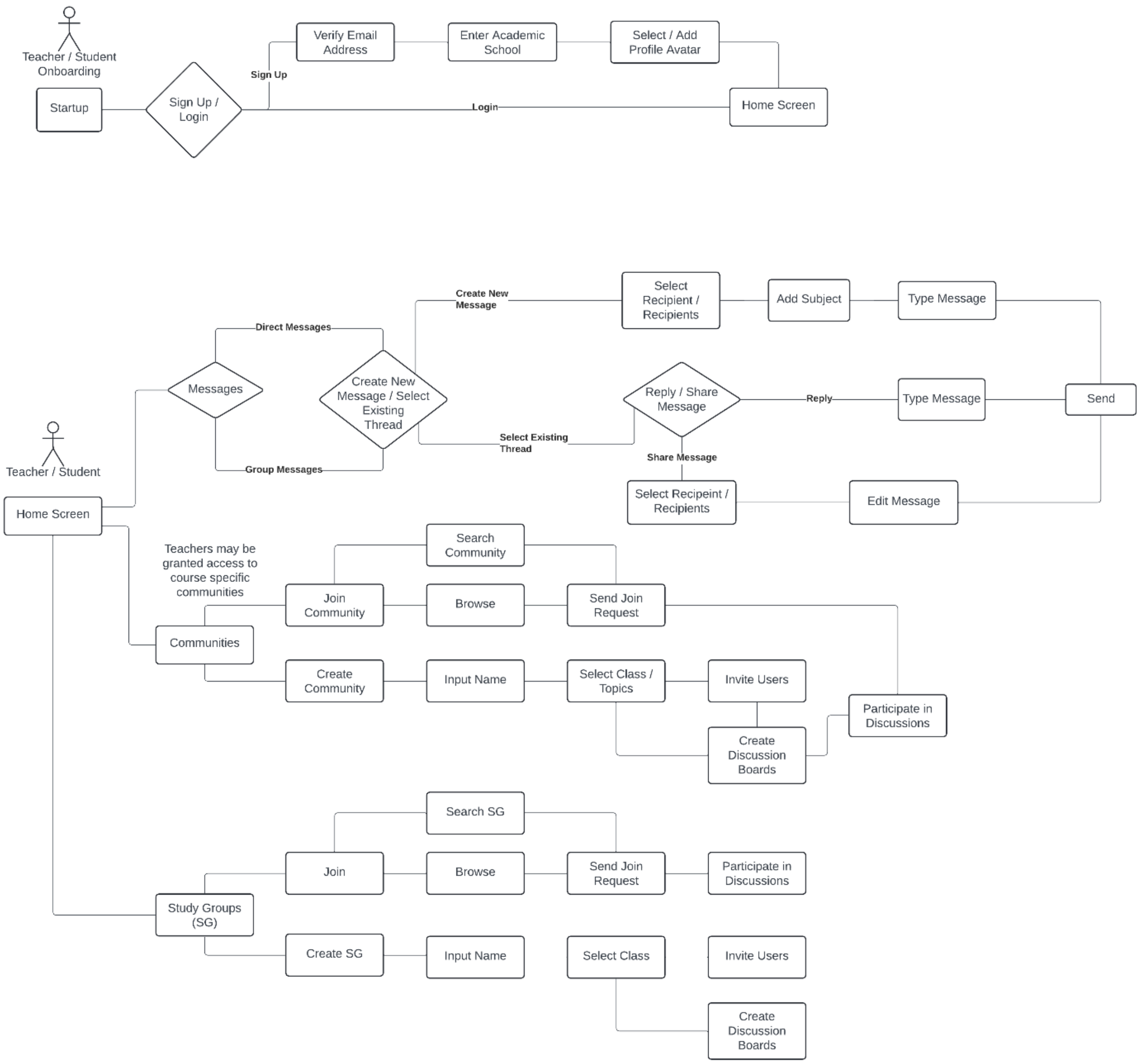

Figure 1: Functional Requirements Flow Chart

3.1.1 Users Permissions

- Administrators will have high level access to ChatFAU, including the ability to see and edit all communities, study groups, and messages. These rights should be reserved for the school's IT department as well.
- Teachers will have access to create and edit course specified communities and have a higher level of access than students.
- Moderators will have more access than students and can be granted this role by a teacher.
- Students will only have visibility and access to the content they create and are added to.

3.1.2 User Onboarding

- The application must allow users to create an account using school emails.
    - Personal emails may be added as a backup option.
    - Teacher specific accounts should be verified.
    - Accounts should have a name, username, UID, school, avatar, and academic level.
- Registered users need to be able to log in and out on request.

3.1.3 User Messaging Functionality

- Users will have the ability to create direct or group messages.
- Messages will contain recipients, subject matter, and the message itself.
- Attachments may be added to messages.
- Messages will include standard markup language options as well as support for custom emojis.

3.1.4 Community Functionality

- Users will have the ability to either join or create communities.
- Study groups may be linked inside of communities but not vice versa.
- Roles may be assigned to grant a level of access within the community.

3.1.5 Study Group Functionality

- Users will have the ability to either join or create study groups.
- Study groups shall only be available during the duration of the class.

3.1.6 Notifications

- Users shall have multiple options on how they would like to be notified.
- Messages, communities, and study groups shall have grouped and individual notification settings.

3.1.7 Role Specific Tools

- Moderators / teachers will have tools to moderate inappropriate behavior.
- Community and study group leaders, as well as moderators, shall have tools like the ability to silence users.

3.1.8 Polls and Quizzes

- Users shall be allowed to post polls and quizzes in study groups or community discussions according to permissions granted by moderators.

3.1.9 Profile Customization

- Users shall be granted permission to change their profile avatar, profile theme, and profile background.
- Users will have the option to showcase badges they have procured.

3.1.10 Users Groups

- Users must send applications for certain groups.
- Badges can be granted for users to showcase.

3.1.11 Badges

- Teachers may grant badges to students.
- Students may receive badges through groups, communities, or study groups.
- Badges may be awarded through quizzes or overall quiz scores.

3.2 Non-Functional Requirements

3.2.1 Reliability

- Backup servers shall contain snapshots of database.
- An emergency backup server shall be developed to run without the need for Supabase.

3.2.2 Availability

- The user interface should be simplistic and intuitive, mimicking typical discussion boards and other similar applications that require minimal training for new users.
- Customization will only be allowed in background themes and colors. This will force the structure to be consistent across different users.
- Any device with access to the internet shall be able to use this application.
- Applications shall have breakpoints to accommodate different standardized screen sizes. According to the authors of the Microsoft article (2023), common standard breakpoints are at 640px, 641px – 1007px, and anything larger than 1007px.

3.2.3 Security

- Accounts shall hold different levels of access, student being the lowest and admins the highest.
- Users' information will be stored in Supabase which is a cloud service database with security tools like RLS.
- RLS (Row Level Security) will be implemented as an extra layer of security when manipulating stored data.
- Minimal data will be locally stored. Any sensitive data shall be hashed and stored in the database.
    - Users' passwords
    - Private message data
    - Users shall be directed to FAU’s SSO for an authentication token.

3.2.4 Maintainability

- Documentation shall be created explaining the code.
- Git / GitHub will be used to store and track progress of source code.
- Unused and nonessential data within the database shall be archived after a period of time.

3.2.5 Portability

- ChatFAU shall be accessible on any device with access to the internet.
- Internet service is required to access the application.

2.3.6 Performance

- The application needs to perform at typical responsive speeds.

- Ghost loading, preloading, and prefetching shall all be used in order to create a fast and fluid application.
- The system shall have haptic feedback on most user inputs.
- User shall be notified if there are any errors encountered.

4. External Interface Requirements

4.1 User Interfaces

4.1.1 Design

The user interface, or UI, will be easy to navigate and highly intuitive, adhering to modern standards commonly used by chat platforms to enhance usability.

4.1.2 Accessibility

ChatFAU will be quick and responsive, adapting seamlessly to various screen sizes and devices, including desktops, mobile phones, and tablets, whether on Android or iOS.

4.1.2 Inputs

Users will have multiple opportunities to interact within the application, including chatting on different servers and through private messaging. Text will support standard formatting and the addition of customized emojis.

4.1.3 Customization

Users will be able to customize their profiles by adding status updates, different usernames for servers, and profile pictures. They can also include avatars while adhering to established customization standards.

4.2 Hardware Interfaces

4.2.1 Server Hardware

ChatFAU will utilize a cloud-based hosting system to support server-side operations. This includes primary servers for user authentication, real-time chat services, and database interactions. Additionally, database servers will manage user data, message histories, and various information from servers and the community, effectively handling large volumes of user requests. Caching servers, such as Cloudflare, will further enhance performance.

4.2.2 Client Devices

Users can access ChatFAU from any device with internet access and an updated web browser, including macOS, iOS, Windows, and Android, covering both desktops and mobile devices.

4.2.3 Backup Hardware

Backup servers will ensure data integrity in the event of system failures, power outages, or hardware malfunctions, serving as part of our disaster recovery systems.

4.3 Software Interfaces

4.3.1 Version Control

- Git and GitHub will be used for source code management.

4.3.2 SSO

- Single sign-on (SSO) will be integrated to ensure the system remains secure for all users on the platform, including staff, students, and faculty.

4.3.3 Web Compatibility

- The application will be compatible with major browsers such as Chrome, Firefox, Safari, Edge, and Brave.

4.2.4 Database Interface

- ChatFAU will use Supabase as its database management service, built on PostgreSQL. This robust and secure system will handle user profiles, messages, chat logs, communities, study groups, permissions, and achievements, including badges.

4.4 Communication Interfaces

4.4.1 WebSockets

- Real-time messaging will be enabled to ensure synchronous communication in the communities and chat rooms.

4.4.2 Encryption

- Data will be encrypted using Transport Layer Security (TLS), and sensitive data will be hashed and stored securely.

4.4.3 Notifications Protocol

- Push notifications and emails will be sent appropriately using the correct protocols.

4.4.4 HTTP/HTTPS

- Connections to the application and communication between users and servers will occur over this protocol to ensure the security and privacy of data transmission.

5. System Features

5.1 User Roles and Permissions

Users on ChatFAU will be assigned roles (admin, teacher, moderator, student) that define their level of access. Admins will have the highest level of control, including access to all communities, study groups, and user data. Teachers will be able to create and manage course-specific communities, while moderators will help enforce rules in chat groups. Students will have limited permissions, primarily focused on participation within groups.

- **Dependencies**:
    - Supabase will manage role-specific data using row-level security (RLS) to filter information based on user roles.

    - FAU’s SSO will ensure that users are authenticated and assigned the correct role upon login.
- **Constraints**:
    - User role management will rely heavily on the integrity of the Supabase database and the proper integration of RLS.

5.2 Communities and Study Groups

Users will be able to create and join communities and study groups that align with their academic needs. Moderation will be implemented and assigned to specific individuals, either those who created the group or trusted users, granting them the authority to silence, ban, or revoke access to maintain a safe-for-work (SFW) environment. Study groups will focus on specific courses, providing a space for students to collaborate and thrive in their academic pursuits

- **Dependencies**:
    - Supabase will manage the study groups and community data, including memberships and time-bound restrictions on groups.
    - Real-time communication via WebSockets will ensure that groups are updated instantly.
- **Constraints**:
    - Server performance will need to scale with the increasing number of users and communities.

5.3 Notifications

Users can customize the push notifications they receive, choosing not to receive notifications for every message. This includes options for mobile, email, or in-app notifications. Additionally, users can set preferences based on specific conversations or topics, allowing them to stay informed without being overwhelmed. This flexibility helps ensure that users only receive the notifications that matter most to them.

- **Dependencies**:
    - Backend notification services will be integrated to send push notifications as needed.
    - Notification settings will allow users to customize their preferences, which will be stored in their profiles in Supabase.
- **Constraints**:
    - Timely delivery of notifications, especially for time-sensitive academic messages, must be ensured.
    - Managing the frequency and types of notifications could put stress on the system, particularly with a large user base.

5.4 Polls and Quizzes

Users can create polls within the communities and study groups, while teachers or TAs can create quizzes in the servers, awarding users achievements and badges based on their scores. This interactive feature encourages engagement and participation among users. Additionally, it fosters a sense of accomplishment, motivating students to improve their knowledge and skills.

- **Dependencies**:
  - Supabase will store quiz results, polls, and user responses.
  - The frontend UI for quizzes and polls needs to be user-friendly and accessible to all users.
- **Constraints**:
  - The volume of responses could strain the database, so pagination or caching techniques may be required.

5.5 Messaging Features

The application supports emojis and various text formats, such as italic and bold, along with custom emoji integration. Additionally, users can attach files, including PDFs and images, in their chats. Direct messages can be sent between users through private messaging, enhancing communication within the platform.

- **Dependencies**:
  - WebSockets will facilitate real-time messaging between users.
  - Supabase will store data and logs for retrieval, including message history.
- **Constraints**:
  - Ensuring data consistency and reliability during high volumes of messages may pose challenges.
  - The system must be designed to handle potential latency issues that could arise during peak usage times.

5.6 Moderation

Moderators will have tools to manage community behavior, including commands such as /mute, /ban, /timeout, and /prune. This ensures that communities remain productive and free from disruptive behavior. These tools empower moderators to maintain a positive environment for all users. By swiftly addressing issues, they can foster a sense of safety and respect within the community.

- **Dependencies**:
  - Supabase will handle role-based access to moderation tools.
  - Real-time commands must be executed instantly for effective moderation.
- **Constraints**:
  - Incorrectly implemented moderation tools could lead to overreach or abuse by moderators, so proper review of their actions is required.

6. Other Requirements

6.1 Legal, Regulatory, and Compliance Needs

- **Terms of Use & Privacy Policy:** Clear terms of use and privacy policy documentation will be provided, outlining how user data is collected, stored, and managed.
- ChatFAU will follow data privacy best practices by hashing sensitive data and minimizing the collection of personal data. It will also comply with the Family Educational Rights and Privacy Act (FERPA) to ensure the privacy and security of student records.

- ChatFAU will follow data privacy best practices by hashing sensitive data and minimizing the collection of personal data. It will also comply with the Family Educational Rights and Privacy Act (FERPA) to ensure the privacy and security of student records.