Introduction to Software Engineering

Requirements Analysis

Software Requirements Specification (SRS) document for the assigned course project, following the attached template.

Software Engineering Department
Faculty of Information and Technology
University of Science

Table of Contents

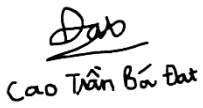
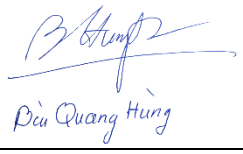
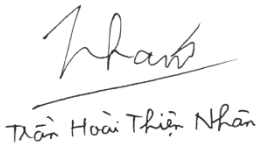
Software Requirements Specification

Objectives

This document focus on the following topics:

- ✓ Complete the Software Requirements Specification (SRS) document with the following contents:
  - Elaborate on the Problem Statement
  - Overview of Requirements (Functional and Non-Functional), Stakeholders
  - Use Case Model
  - Use Case Specifications
  - Create Prototype and Mockup Diagrams of the System Interface

1 Member Contribution Assessment

ID	Name	Contribution (%)	Signature
23127097	Trần Trí Nhân	100%	
23127168	Cao Trần Bá Đạt	100%	
23127193	Bùi Quang Hùng	100%	
23127234	Nguyễn An Nghiệp	100%	
23127238	Trần Hoài Thiện Nhân	100%	

2 Problem Statement

2.1. Business Description of the Software Problem

2.1.1. The Problem and Purpose of the Software

In today's digital environment, both individuals and businesses rely heavily on online platforms to handle customer support and service operations. Maintaining a human support team that can respond consistently and continuously (24/7) is costly, inefficient, and difficult to scale. Organizations typically face large volumes of repetitive inquiries daily, resulting in a significant workload on support staff and reduced overall service quality.

The **ChatBot_HelpDesk** project is designed to address these challenges by providing a **web-based platform that enables users to create, customize, manage, and deploy AI-powered chatbot agents**. These agents help automate customer interaction, provide instant responses, and support internal service operations.

Purpose of the Software

The system enables users to:

- Create custom AI agents for various use cases (customer inquiry handling, complaint resolution, internal assistance, etc.).
- Configure knowledge bases and response behaviors for each chatbot.
- Deploy and integrate chatbots directly into websites or internal applications.
- Share usage access and collaborate within teams.
- Track performance metrics and support workflows through ticket management.

Target Users

The platform focuses on two main user groups:

1. **Individual users** seeking a simple, customizable chatbot solution
   → Subscription: **200,000 VND/month**

2. **Business organizations** requiring collaboration and advanced management features
   → Subscription: **500,000 VND/month** + **100 VND per AI-generated ticket response**

Why the Software Is Needed

Without such a system, organizations face:

- Long response times due to manual support.
- High operational costs for maintaining support staff.
- Difficulty ensuring consistent quality across different support agents.
- Lack of structured ticket management and knowledge base updates.
- No analytics or data insights for improving support processes.

2.1.2. Business Workflow and Operational Process

Current Workflow (Before the Software)

- Customers contact support via email/phone and wait for manual responses.
- Staff must repeatedly answer similar or identical questions.
- Customer data and support records are fragmented.
- Departments lack a shared knowledge base or shared chatbot agent.

Workflow After Introducing ChatBot_HelpDesk

1. **AI Agent Creation**
    a. Users define the agent's purpose and upload domain-specific knowledge.
    b. They configure instructions that shape the AI's behavior.
    c. A built-in testing tool lets users refine agent responses before deployment.

2. **Sharing and Collaboration**
    a. Users may share agents individually via email.
    b. Businesses can create group workspaces (teams, departments) where multiple employees share access to an agent.

3. **Website Integration**
    a. The system generates embed scripts for quick installation into websites.
    b. Customers interact directly with chatbots on the webpage.

4. **Helpdesk and Ticket Management**
    a. If an AI agent cannot answer, unresolved queries become tickets automatically.
    b. Staff members handle these tickets and update the agent's knowledge base accordingly.

5. **Reporting and Analytics**
    a. The system provides metrics such as interaction counts, resolution time, user satisfaction, and ticket volumes.
    b. Managers use these insights to optimize service processes and improve workflows.

Benefits

- Automates 70–90% of repetitive inquiries.
- Reduces support, workload, and operational costs.
- Provides 24/7 instant responses for improved customer satisfaction.
- Enhances internal collaboration through shared agents.
- Provides actionable analytics to optimize customer support performance.

2.2. Operating Environment

2.2.1. Client Environment

The platform runs fully in a web environment with:

- Support for modern browsers: **Chrome, Edge, Brave,...**
- Responsive design for **desktops and** tablets.
- No installation required: only an Internet connection is needed

Client Requirements

- Modern web browser (updated within the last 2–3 years)
- Stable Internet connection
- No additional plug-ins or software installations

2.2.2. Server and Backend Environment

- **Framework:** Next.js (full-stack capabilities)
- **Deployment:** Vercel (optimized for Next.js applications)
- **Database:** Firebase (Firestore + Authentication)
    - Real-time data synchronization
    - Firebase Security Rules for data protection

2.2.3. Integration Capabilities

- Simple website integration via a JavaScript embed script
- No additional services required on the client side
- Future expansion planned for connecting with **Banking and Visa/Mastercard Payment** Gateways, enhanced **Mobile Responsiveness**

2.3. Design & Implementation Constraints

2.3.1. Technological Constraints

- **Programming Language:** TypeScript
- **Web Framework:** Next.js
- **Database:** Firebase (Firestore)
- **Deployment Platform:** Vercel

2.3.2. Documentation & Technical Standards

- Must follow TypeScript and JavaScript coding conventions.
- Must use modular, reusable UI components.
- Project documentation follows standard Software Engineering guidelines:
    - UML diagrams such as **use-case model**, class, and sequence diagrams.
    - Structured report writing aligned with university project expectations (ex: **PA**).

2.3.3. Non-Functional Constraints

- **Performance:** Fast response time using serverless architecture on Vercel.
- **Security:** Firebase Authentication + Firestore security rules.
- **Scalability:** Both Firebase and Vercel offer horizontal scaling.
- **Maintainability:** Modular code structure designed for long-term development.

3 Requirements Overview

3.1 Stakeholders

STT	Requirements	Description
1	Project Sponsor	Senior Management/Executive Leadership (e.g., CTO/COO). The person who approves the budget, defines the strategic objectives, and gives final approval on the project's scope and outcome.
2	Project Manager	The student team member responsible for project management. Ensures the project stays on track, acts as the main point of contact, and manages the scope, timeline, and resources.
3	End Users/Internal Staff	All employees within the company who will use the chatbot for internal customer support, information retrieval, and notifications. They are the ones who evaluate the usability and performance of the chatbot.
4	Content/Knowledge Managers	Specialists responsible for internal documentation and processes (e.g., HR, Administration Dept.). They are accountable for the quality of input data via the "Update/Knowledge upload" stream.
5	IT/Infrastructure Team	The company's IT department. They are responsible for the deployment environment, system security, integration with existing infrastructure, and ensuring stable chatbot operation.
6	Subject Matter Experts (SMEs)	Internal professional experts (e.g., Head of Accounting, Sales). They provide validation and accuracy checks for the AI-generated responses (often working alongside Content Managers).

3.2 Requirements

3.2.1. Functional Requirements Specification

FR.1. Knowledge Management Requirements (Upload/Update Knowledge)

STT	Requirements	Description
1.1	Upload	The system must allow Knowledge Managers to upload various types of internal documents (e.g., PDF, DOCX, TXT, CSV) via the web interface to serve as the AI's knowledge source.
1.2	View	The system must allow viewing, searching, and categorizing the uploaded knowledge sources.

1.3	Delete	The system must allow deletion or deactivation of outdated or inaccurate knowledge sources.
1.4	Index	The system must automatically process and index the uploaded data to prepare it for AI retrieval and answering (RAG - Retrieval-Augmented Generation).

FR.2. Web Interface Requirements (Usability)

STT	Requirements	Description
2.1	Chat	The web interface must provide an intuitive chat area, allowing users to input questions and receive answers from the chatbot.
2.2	Multi-platform responsive	The web interface must be user-friendly and responsive across various screen sizes (desktop, mobile devices).
2.3	History	The system must display the current user's conversation history.
2.4	Admin monitor	The system must have a separate administration interface to perform the functions specified in FR.1 and FR.3.

FR.3. Group Chat Session and Access Management Requirements (Invite member)

STT	Requirements	Description
3.1	Authentication	The system must require login/authentication to access and use any feature.
3.2	Create group chat	The system must allow any logged-in user (customers, sales staff, etc.) to create a new group chat session with the AI agent.
3.3	Invite member to group chat	The group creator/owner must be able to invite members to the group chat session by entering the invitee's email address.
3.4	Send email	The system must automatically send an invitation email to the invitee's email address, along with an access link to the group chat.
3.5	Role	The system must define and manage user roles within the group chat (e.g., Owner/Creator, Member), allowing the Owner to manage membership (invite, remove).
3.6	Concurrent	The system must support multiple users concurrently interacting with and viewing the AI agent's responses within the same group chat session.

FR.4. Chatbot AI Core Requirements

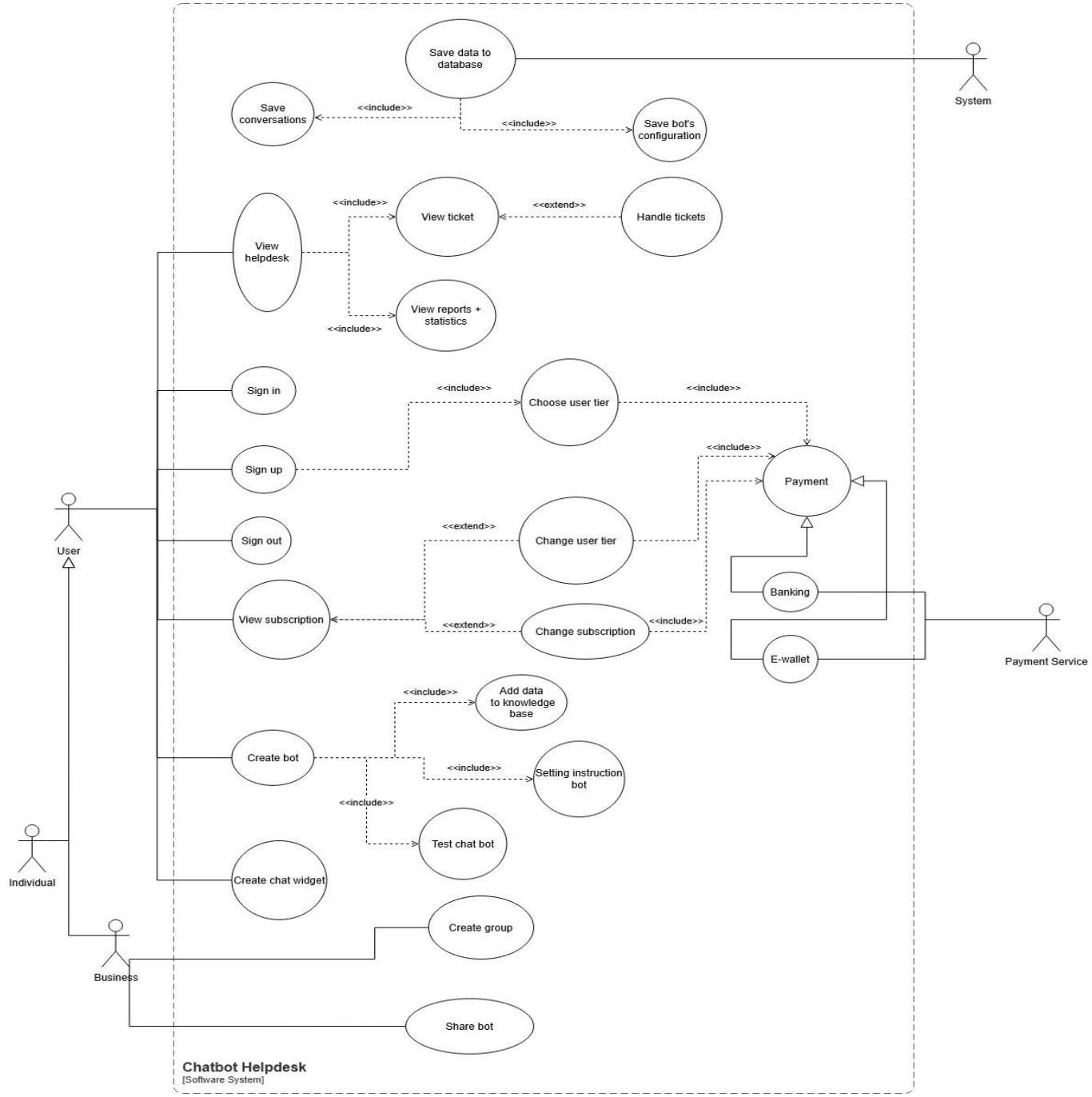
STT	Requirements	Description
4.1	AI Answer	The Chatbot must be capable of extracting information from the internal knowledge base (FR.1) to answer specific employee questions.
4.2	AI Fallback	When no suitable information is found in the knowledge base, the chatbot must provide a friendly fallback message instead of giving incorrect answers (e.g., "I apologize, this information has not yet been updated in the system").
4.3	AI Context	The Chatbot must be able to maintain context within a conversation to answer follow-up questions logically.

3.2.2. Non-Functional Requirements Specification

- Loading pages takes under a second. Transitions between components within a screen have response time under 0.5 second.
- A user's password needs to be hashed before inserting into the database.
- Users can efficiently navigate through the website after an hour of interacting.
- Maintain availability of 90% and response time under 3 seconds during high traffic.
- The website must be responsive.
- Data in the database must be backed up once every 2 days.
- The website can handle 100 users simultaneously.
- The website can run on most of the popular browsers like Chrome, Firefox, Safari, Edge,...
- The website needs to adhere to data protection regulations.
- In the event of failure, the website needs to be recovered within 24 hours.

4 Requirements Analysis

4.1 Use Case model



4.2 Use Case Specification

4.2.1. Use Case 1

Use case ID	**UC01**
Use Case	Sign up
Brief Description	The user creates a new account to use the system. This process involves selecting a user tier and making a payment.
Actor	User (Individual, Business), Payment Service
Pre-Condition	The user is not logged in and is on the homepage.
Result	The account is created, and the service tier is activated.
Main Scenario	1. The user selects the "Sign up" function. 2. The user enters personal information (Email, password, name...). 3. The system requests the user to choose a service tier. 4. The user selects a tier. 5. The system proceeds to the payment step. 6. The user selects a payment method (Banking or E-wallet). 7. The system sends a request to the Payment Service. 8. The Payment Service confirms the payment is successful. 9. The system creates the account and notifies success.
Alternative Scenarios	*At step 8: If the payment fails:* *8a. The system notifies a transaction error.* *8b. The system requests the user to choose another method or try again.*
Non-Functional Constraints	- Passwords must be encrypted before storage. - Payment transactions must comply with security standards (e.g., secure API connection).

4.2.2. *Use Case 2*

Use case ID	**UC02**
Use Case	Sign in
Brief Description	The user accesses the system to use Bot management and Helpdesk functions.
Actor	User (Individual, Business)
Pre-Condition	The user is not logged in (Guest).
Result	The user receives an authentication token and accesses the system.
Main Scenario	1. The user clicks the "Sign in" button. 2. The system displays the login form. 3. The user chooses a login method (Enter Email/Password OR select Google SSO). 4. The system authenticates the user information. 5. If the information is correct, the system redirects the user to the Dashboard page.
Alternative Scenarios	*At step 4: If the username or password is incorrect:* 4a. The system displays an "Incorrect login information" notification. 4b. The system requests the user to re-enter the information.
Non-Functional Constraints	- Authentication tokens (e.g., JWT) must be managed securely. - The client must never connect directly to cloud provider services using raw credentials.

4.2.3. Use Case 3

Use case ID	UC03
Use Case	Sign out
Brief Description	The user ends the current session to secure their account.
Actor	User (Individual, Business)
Pre-Condition	The user is currently logged in.
Result	The session is terminated, and the user returns to the Guest state.
Main Scenario	1. The user clicks on the Avatar or account menu. 2. The user selects "Sign out". 3. The system requests confirmation (displays a confirmation popup). 4. The user confirms. 5. The system deletes the login session information (Session/Token) on the browser. 6. The system redirects to the Landing Page or Login Page.
Alternative Scenarios	At step 4: if the user declines. 4.a: User returns to bot management page.
Non-Functional Constraints	- Session invalidation must happen immediately on the client side. - Navigation to protected pages after logout must be blocked.

4.2.4. Use Case 4

Use case ID	UC04
Use Case	View & Change Subscription
Brief Description	The user upgrades or downgrades their current service package.
Actor	User (Individual, Business), Payment Service
Pre-Condition	The user is logged in and is viewing subscription information (View subscription).

Result	The account is updated with the new service package.
Main Scenario	1. From the subscription view interface, the user selects "Change Subscription". 2. The user chooses to change the user tier (Change user tier). 3. The user selects a new package. 4. The system calculates the cost difference and requests payment. 5. The user performs the payment via Banking or E-wallet. 6. The system updates the new service package for the account.
Alternative Scenarios	*At step 5: If the user cancels the payment:* 5a. The system retains the current package. 5b. Return to the subscription information screen.
Non-Functional Constraints	- Pricing calculations (e.g., 200,000 VND for Individuals, 500,000 VND and 100 VND/token for Business) must be accurate.

4.2.5. Use Case 5

Use case ID	**UC05**
Use Case	Create Bot
Brief Description	The process of creating a new chatbot, including loading knowledge data, setting instructions, and testing before completion.
Actor	User (Individual, Business)
Pre-Condition	The user is logged in.
Result	The Bot is successfully created, data is vectorized, and the bot is ready for operation.
Main Scenario	1. The user selects the "Create bot" function on the Dashboard. 2. The system displays the Bot creation interface. 3. The user enters basic information (Bot Name, Description). 4. **Add Data to Knowledge Base:** The user selects a data source, performs **Upload file** (supports: .pdf,

	.docx, .txt, images .jpg/.png...) or enters text directly. 5. The system processes the file and vectorizes the data. 6. **Setting Instruction Bot:** The user enters instructions (System Prompt) into the editor or selects **Upload file instruction** (if a behavioral rule description file is available). 7. **Test Chat Bot:** Right on the Create Bot page, the system displays a **Preview** (Chat Demo) window. The user sends a test message to the Bot; the Bot responds based on the newly entered data and instructions. 8. The user clicks "Save/Finish" to complete. 9. The system saves the Bot configuration and notifies success.
Alternative Scenarios	At step 4: If the file format is not supported: 4a. The system displays a file format error notification. 4b. The system requests the user to upload a valid file. At step 5: If the data vectorization process fails: **5a. The system reports a data processing error.** **5b. The user checks the content and tries again.**
Non-Functional Constraints	- Strict data privacy: Third-party AI APIs must not use internal data for training. - Supported file formats must include PDF, DOCX. - File processing and embedding creation must handle "Cold Starts" from serverless functions gracefully.

4.2.6. Use Case 6

Use case ID	**UC06**
Use Case	Create Chat Widget
Brief Description	Generate embed code or a chat interface to integrate into a website.
Actor	User (Individual, Business)
Pre-Condition	At least one Bot has been created.
Result	The user receives the script code to embed into their

	website.
Main Scenario	1. The user selects the Bot needing a Widget. 2. The user selects the "Create chat widget" function. 3. The system generates the embed code. 4. The user copies the code for use.
Alternative Scenarios	
Non-Functional Constraints	- The generated script must be optimized to not affect the loading speed of the target website. - The widget must be responsive on both desktop and mobile devices.

4.2.7. Use Case 7

Use case ID	**UC07**
Use Case	View Helpdesk
Brief Description	Access the customer support dashboard to view tickets and reports.
Actor	User
Pre-Condition	Logged in.
Result	The user sees an overview of the customer support status.
Main Scenario	1. The user selects the "Helpdesk" menu. 2. The system displays the list of support tickets (View ticket). 3. The system displays report charts and statistics (View reports + statistics). 4. In statistics view the user can export the chart to file.
Alternative Scenarios	At step 4: If the system failed to export to file. 4.a: An error message pops up to indicate failure to export.
Non-Functional Constraints	- Reports must accurately reflect interaction metrics and ticket resolution times. - Dashboard data loading should be efficient.

4.2.8. Use Case 8

Use case ID	UC08
Use Case	Handle Tickets
Brief Description	Staff/User responds to and resolves requests from customers.
Actor	User
Pre-Condition	The user is viewing the ticket list (View ticket).
Result	The ticket status is updated, and the response is sent.
Main Scenario	1. From the ticket view interface, the user selects a specific ticket to handle. 2. The system displays detailed ticket content. 3. The user enters a response or changes the ticket status (Open -> Resolved). 4. The system saves the update.
Alternative Scenarios	At step 3: if changes can't be made to the ticket status at the moment due to some error: 3.a: an alert shows up. 3.b: user retries to change the ticket status.
Non-Functional Constraints	- Updates to ticket status must be recorded in the NoSQL database immediately. - The system must manage customer and employee data securely,

4.2.9. Use Case 9

Use case ID	UC09
Use Case	Create Group
Brief Description	Create a working group for multiple people to manage bots together.
Actor	Business (Business account only)
Pre-Condition	The account type is Business.
Result	A new group is created with the invited members.

Main Scenario	1. The Business User selects the group management function. 2. The user selects "Create group". 3. The user enters the group name and adds members (enter email). 4. The system sends invitations and creates the group.
Alternative Scenarios	*At step 3: If the email does not exist in the system:* 3a. The system notifies that the email is not registered. 3b. The system sends an email inviting them to join the platform (optional).
Non-Functional Constraints	- Access control must ensure only Business accounts can create groups

4.2.10. Use Case 10

Use case ID	**UC10**
Use Case	Share Bot
Brief Description	Share Bot access rights with a group or another user.
Actor	Business
Pre-Condition	A Bot exists, and the account is Business.
Result	The shared user has access rights to the Bot.
Main Scenario	1. The Business User selects the Bot to share. 2. The user selects the "Share bot" function. 3. The user selects a Group. 4. The user sets permissions (View/Edit). 5. The system grants access rights.
Alternative Scenarios	
Non-Functional Constraints	- Sharing settings must be updated in real-time for collaboration. - Security constraints regarding data access must be maintained for shared users.

4.2.11. Use Case 11

Use case ID	**UC11**
Use Case	Save Data to Database
Brief Description	System-level use case to ensure data persistence.
Actor	System
Pre-Condition	A data change event occurs (create bot, chat, edit config).
Result	Data is securely stored in the database.
Main Scenario	1. This use case is triggered automatically. 2. When there is a new conversation -> Trigger **Save conversations**. 3. When a user creates/edits a bot -> Trigger **Save bot's configuration**. 4. The system performs data writing to the Database. 5. The system returns a success status.
Alternative Scenarios	At step 5: If the system fails to write data to database. 5.a: The system returns failure status. 5.b: The system retry from step 4 until success.
Non-Functional Constraints	- Data is stored in a NoSQL database (Firebase) . - Security by Layer: Sensitive data handling must be restricted to Server Components.

5 Prototype/Mockup

Link to design prototype/mockup on figma: Prototype and wireframe