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.