Milestone Report

for

HelpDesk Chat Application

Version 1.0 approved

Prepared by Derry Liu, Harsh Patel, Jihal Patel, and Lukas Rose

York University

February 25, 2022

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
First draft	2022-02-25	Initial draft	1.0

1. Introduction

1.1 Purpose

The main goal of this document is to provide an accurate and complete list of requirements for the online help-desk software system to fulfill its goal. The help-desk software system is tasked with providing a means of communication between an anonymous user (client) and help-desk users (provider). It provides user authentication functionality for the help-desk users along with automatic assignment of clients to providers. Help-desk users can also communicate with other authenticated online help-desk users. The software system will provide the ability to the providers to be able to transfer clients to other online providers. This milestone report document specifies the full system requirements along with the server and client design.

1.2 Intended Audience and Reading Suggestions

The primary audience of this document includes, but is not limited to, the EECS 4481 professor, developers, and end-users. It is unlikely that the software will be deployed commercially as it is part of a school project, however, it may be of use to developers creating a help-desk software system as some of the concepts/code can be utilized to meet their requirements. Any end-users that utilize the system may obtain the full list of specifications through this document.

1.3 Product Scope

The product is a basic help-desk software system. The main purpose is to provide a secure method of communication, being an anonymous client accessing the website with a help-desk user. The key objective is to fulfill the requirements specified by the course that the website is intended for. The software is not intended to be used for business or commercial purposes, rather educational purposes. A key benefit of the system is that it will be a basic chat software system which can be extended to be used by different software systems as the foundation of a secure chat system will be present.

2. Server and Client Software Design

2.1 Application Architecture

Our application architecture can be categorized as Monolithic. As seen in the diagrams in sections 7.3 and 7.6 most of the functionality of the Help Desk Application is tightly integrated into a single stack that contains all of the functionality of our application. Updating any functionality will have an impact on the entire application. While almost all our services are tightly integrated, our authentication system relies on a third party service called MongoDB for database capabilities.

2.2 Development Methodology

We chose to develop the program using Feature-Driven Development (FDD), focusing primarily on implementing features that would satisfy the Software requirements given to us. This methodology was the most useful for us, as it allowed us to focus on completing deliverables, while still maintaining some iterability.

2.2.1 Notable Design Decisions

For the development of the project, we chose to use the MERN (MongoDB, Express, React.js, Node.js) stack due to the majority of our group members' familiarity.

The main difference between the client user and provider user is simply that the provider has to log into the app from a separate interface. This allows the handling of both types of users in very similar ways. For this version of the system, both users will use the same chat interface to simplify development.

The choice to use a MongoDB Atlas cloud Database was deliberate, as it simplified the process of connecting to the database, and allowed multiple development members simultaneous access to the same data for ease of testing and debugging.

Since both types of users are treated the same, both the client and provider users will see the same chat interface. To prevent the client users from seeing or having access to chatroom manipulation features available to the provider users (e.g. moving the other user to a different chatroom), the choice was made to allow commands to be entered into the chatbox, which would be parsed, and source-validated by the server, before execution.

2.3 Programming Platform

For the development of our web software system, the code editor used was Visual Studio Code. The choice of programming language for the web application was Javascript. The key reason for this was the group's familiarity with developing web applications using Javascript. Node.js and express are also used. The application is hosted on the localhost on port 3000. Lastly, to store the credentials of help-desk users, the database system chosen was MongoDB, a NoSQL database program.

2.4 GitHub Repository Address

The GitHub repository for the help-desk chat software can be found at the following link: https://github.com/Harsh-B-Patel/HelpDeskApp/tree/main. Please make sure that you utilize the **main branch** in the repository.

3. Overall Description

3.1 Product Perspective

The help-desk software system being specified in this SRS is a basic secure communication system similar to existing software systems with secure communication capabilities. The system can be used with a larger software system with which the developed help-desk system can act as a subcomponent. The two major components of this system are the client functionality and the provider functionality. The functions corresponding to each are described in section 3.2.

3.2 Product Functions

- Help-desk user (provider) functionality: This component consists of user authentication to ensure that the person accessing the system is a help-desk user. Upon authentication, they may choose any available room to either converse with another active and free help-desk user or to enter an empty room to converse with anonymous users that may join.
- Anonymous user (client) functionality: This component consists of the option to join any available rooms without the need of user authentication. The precondition to join a room is that there should be an available room with no other clients. If the precondition is not met, the user will be put into a waiting room.

3.3 User Classes and Characteristics

There are two key user classes in the system, help-desk users and anonymous users. Both users are differentiated based on their privilege level. For help-desk users, they are required to authenticate and are able to enter any available and free (rooms with less than 2 people) rooms. Anonymous users do not require authentication are automatically assigned to a room with a help-desk user if one is available, otherwise they are put into a waiting queue. Both users have the same permissions regarding communication capabilities, however, the help-desk users will have other permissions such as being able to move an anonymous user to another room with an available help-desk agent. Both users are equally important to satisfy as without both users the communication system would lose its key purpose of being able to provide a secure communication channel between two separate users.

3.4 Operating Environment

The operating system that the software will be compatible with is any operating system with browser support. However, the key operating systems focused on and that the system is tested on are Windows and Linux (specifically Kali). The software system makes use of the mongoDB database system (noSQL). It also requires node.js to be installed for the modification of the software system. The preferred node.js version is the latest version at the time of writing but earlier versions of node.js may also work.

3.5 Design and Implementation Constraints

A limitation/constraint may be that the database is noSQL and makes use of a single collection of help-desk user credentials. The collection has the following base pattern:

{
    _id: ObjectId(“id”)
    username: “username”
    password: “password”
}

Though the password is stored in plaintext, the password is intended to be hashed for a more secure method of storage for future releases. The software will be maintained until the end of EECS 4481. The key backend of the software system is located in the index.js file. Another potential constraint for the software system is the lack of registration. The help-desk system will not allow new users to be registered and if new users need to be registered, the database must be modified directly.

3.6 Assumptions and Dependencies

The requirements stated in this document assume that the current security practices are in fact secure and free of vulnerabilities. The testing that takes place to ensure the security of the application will use current practices at the time of writing. If new vulnerabilities in current web standards are discovered past the deployment date, then the website will no longer be considered secure and would require an update to meet the new standards. Similarly, if any vulnerabilities are discovered in the mongoDB database system in the future, then the website nor the credentials of its help-desk users can be considered secure.

4. External Interface Requirements

4.1 User Interfaces

The helpdesk system will interact with the user through a graphical user interface (GUI) displayed on a web browser. The user will first select whether they would like to use the helpdesk system as a client or as an agent. Then a client user will register a name, or an agent user will login with their username and password. Finally the client user will then be able to chat in real time with an agent.

This software product will have 4 unique interfaces:

4.1.1 Default interface

This interface will allow the user to select whether they would like to use the helpdesk system as a client or as an agent. The user will then be taken to the respective registration/login interfaces.

4.1.2 Client registration interface

The client registration interface will allow a user who is registering as a help-desk client to register their name, and upon doing so, the client will be placed in the first available chatroom to speak with an agent user.

4.1.3 Agent login interface

This interface will allow an agent user to login to the system by providing their username and password. This interface will also allow the agent user to manually select a chatroom to enter.

4.1.4 Chatroom interface

The chatroom interface will allow both client users and agent users to send messages to another in real time. It will display the room number as well as previous messages, each with the sender name and timestamp. This interface will also allow agent users to send commands allowing the agent to move client users to other rooms.

4.2 Hardware Interfaces

The user must have a keyboard and mouse to access the helpdesk system through the browser, and interact with the graphical user interfaces defined in section 4.1.

4.3 Software Interfaces

4.3.1 MongoDB

The helpdesk server will connect to a cloud-based MongoDB database hosted on MongoDB Atlas. This database will contain login information for agent users, including their usernames, passwords and id numbers. Upon a login attempt the system will send a query to the database for entries with matching usernames and passwords, and the database will return the entries that match.

4.3.2 Express

The express library is used to extend the functionality of the node.js server used for the help desk system. Express provides tools for URL routing, and handling HTTP requests and responses.

4.3.3 Socket

The socket library is used to create and manage TCP websockets to facilitate the sending of messages and commands in real time between users.

4.4 Communications Interfaces

4.4.1 Web Browser

All users will access the system through a web browser. HTTP will be used to send the GUI to the users, and a POST request will be used to send form data from the user to the help desk server.

4.4.2 Network Server

The helpdesk server will communicate with the MongoDB Atlas cloud database via HTTPS. Queries and responses will be sent and received in JSON format.

4.4.3 Communication Protocols

Users will send and receive messages in real-time, via TCP connections. Sent messages will be received and displayed within 5 seconds

5. System Features

5.1 Authentication

5.1.1 Description and Priority

The authentication feature is responsible for the authentication of help-desk users in order to verify their identity. This feature is of high priority as help-desk users have a higher feature set and are able to choose which room they would like to enter.

5.1.2 Stimulus/Response Sequences

The user action required for this feature is to click on the help-desk user login button from the home page and the authentication page should show up. Upon entering appropriate credentials and clicking login the user will be authenticated and redirected to the appropriate page.

5.1.3 Functional Requirements

The requirements listed below must be completed in the order listed for successful execution and display of the next section.

REQ-1: User is a help-desk user
REQ-2: User has appropriate login credentials
REQ-3: User clicks on the help-desk agent login button on the home page
REQ-4: User enters the appropriate credentials
REQ-5: User clicks login button

5.2 Join Chat Room Button

5.2.1 Description and Priority

The join chat room button is a feature that allows anonymous users to access the chat room. This feature is of high priority as it is the main functionality of the application which is to assign anonymous users to a room with a corresponding help-desk agent.

5.2.2 Stimulus/Response Sequences

The user action required for this feature is to click on the join button from the home page and the waiting queue or the chat room should appear if there is an available help-desk agent.

5.2.3 Functional Requirements

The requirements listed below must be completed in the order listed for successful execution and display of the next section.

REQ-1: User is a anonymous user (client)
REQ-2: User clicks on the join button
REQ-3: User waits for an available room/help-desk agent

5.3 Room Selection

5.3.1 Description and Priority

Room selection is a feature available to help-desk users to choose the room they want to enter. This feature has a medium priority as without it, the help-desk user would be able to get an automatically assigned empty room but would not be able to join other help-desk users to discuss with them.

5.3.2 Stimulus/Response Sequences

The user action required for this feature is to click on the help-desk user login button from the home page and the authentication page should show up. Upon entering appropriate credentials and clicking login the user will be authenticated and redirected to the room selection page. Upon selecting the room the user wishes to enter, they can click join and wait for an anonymous user (client) in need of help.

5.3.3 Functional Requirements

The requirements listed below must be completed in the order listed for successful execution and display of the next section.

REQ-1: User is a help-desk user
REQ-2: User has appropriate login credentials
REQ-3: User clicks on the help-desk agent login button on the home page
REQ-4: User enters the appropriate credentials
REQ-5: User clicks login button
REQ-6: User is authenticated

REQ-7: User selects room from list of available rooms
REQ-8: User clicks on join room button

6. Other Nonfunctional Requirements

6.1 Performance Requirements

REQ-1: Navigating between pages should complete within 1 second or less.
REQ-2: A sent message should be received within 1 second or less.
REQ-3: A command sent by a client user should be denied within 1 second or less.
REQ-4: A command sent by an agent user should be executed within 1 second or less.

6.2 Safety Requirements

The software system does not directly affect user's systems. The only possible loss, damage or harm caused by this helpdesk system is the client user incorrectly following advice from agent users. To prevent this, all agent users must be certified help-desk agents, and validate their credentials upon login. The client users must also be aware that the helpdesk system is not responsible for damage, loss or harm caused by incorrectly followed advice, or any other client user's actions.

6.3 Security Requirements

The system must protect the privacy of the client users. Client users must not be able to view or listen in on the conversations of other client users. Only certified agent users can be allowed to use the system as an agent user. Agent users must enter their username and password upon login to validate their credentials.

6.4 Software Quality Attributes

REQ-1: The system must be available whenever an active agent user is logged in.
REQ-2: The system should be easy to use for a client user, each interface should have no more than 5 items to interact with.
REQ-3: The client user should be able to fully learn how to use the system upon their first use.

6.5 Business Rules

6.5.1 Client user

REQ-1: The client user can register their name.
REQ-2: The client user can send and receive messages.
REQ-3: The client user cannot execute commands.

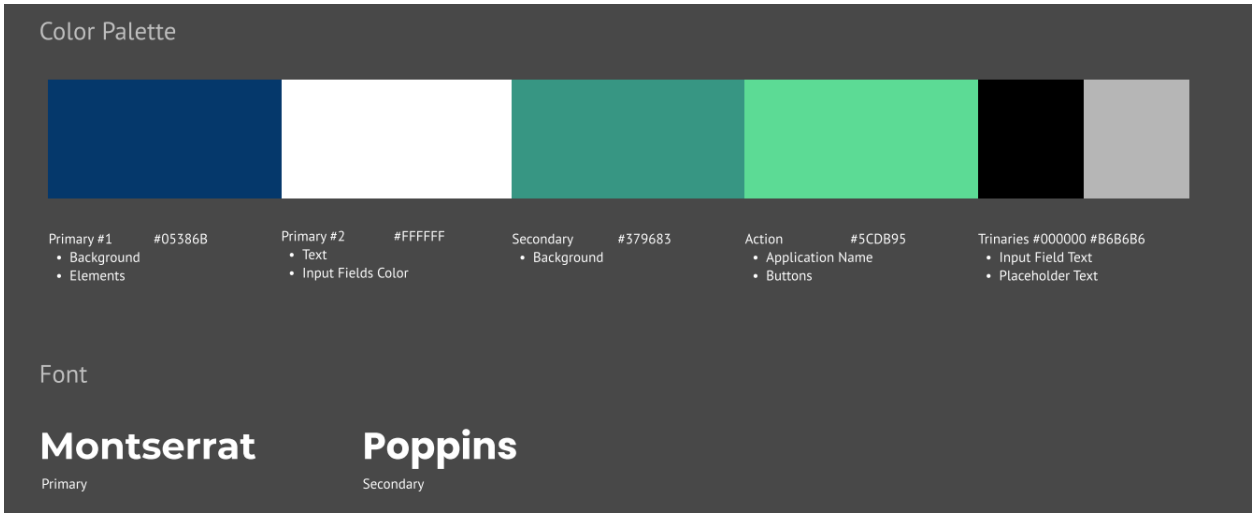
6.5.2 Agent user

REQ-1: The agent user can validate their credentials.
REQ-2: The agent user can send and receive messages.
REQ-3: The agent user can execute commands after logging into a chat room.

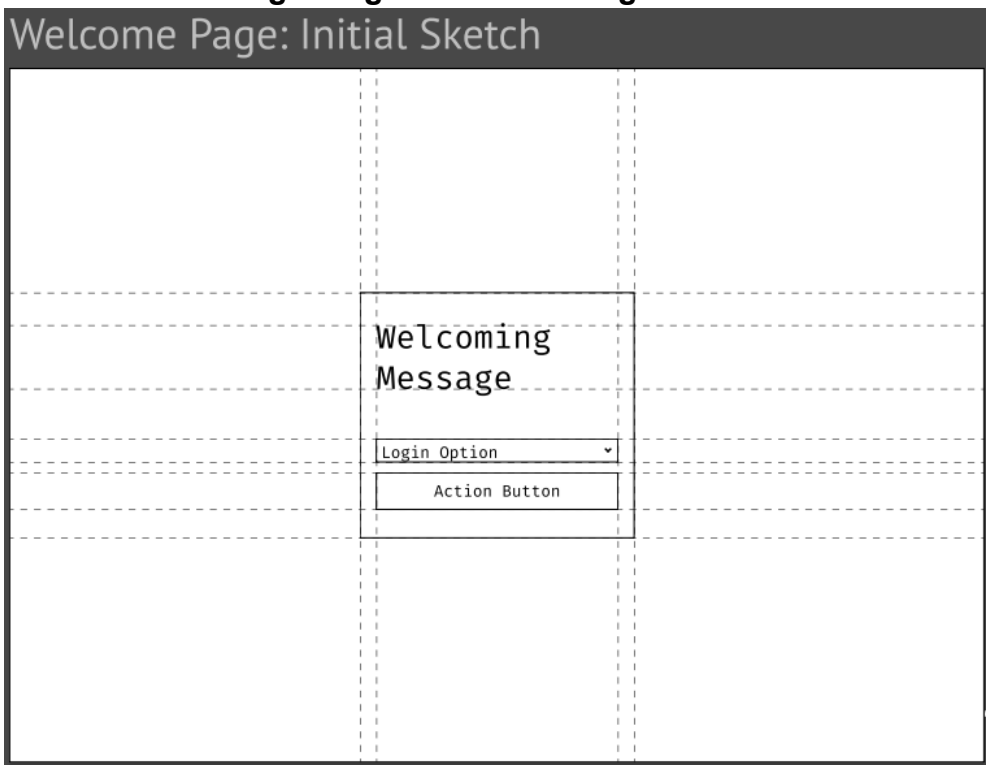
7. Diagrams

7.1 GUI Layout Design

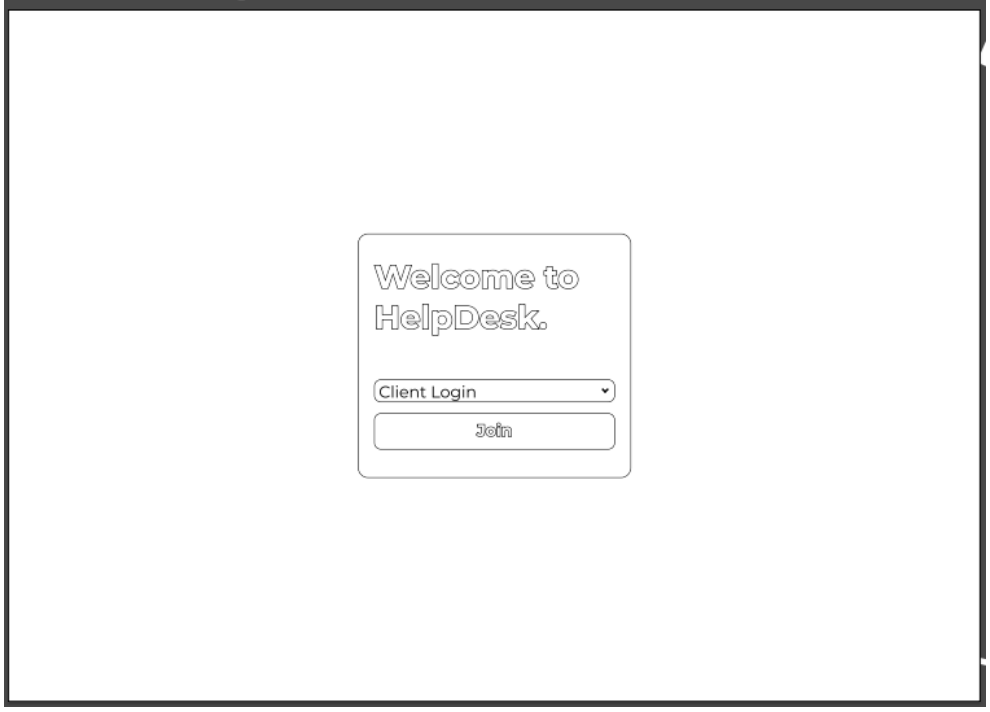
7.1.1 Visuals



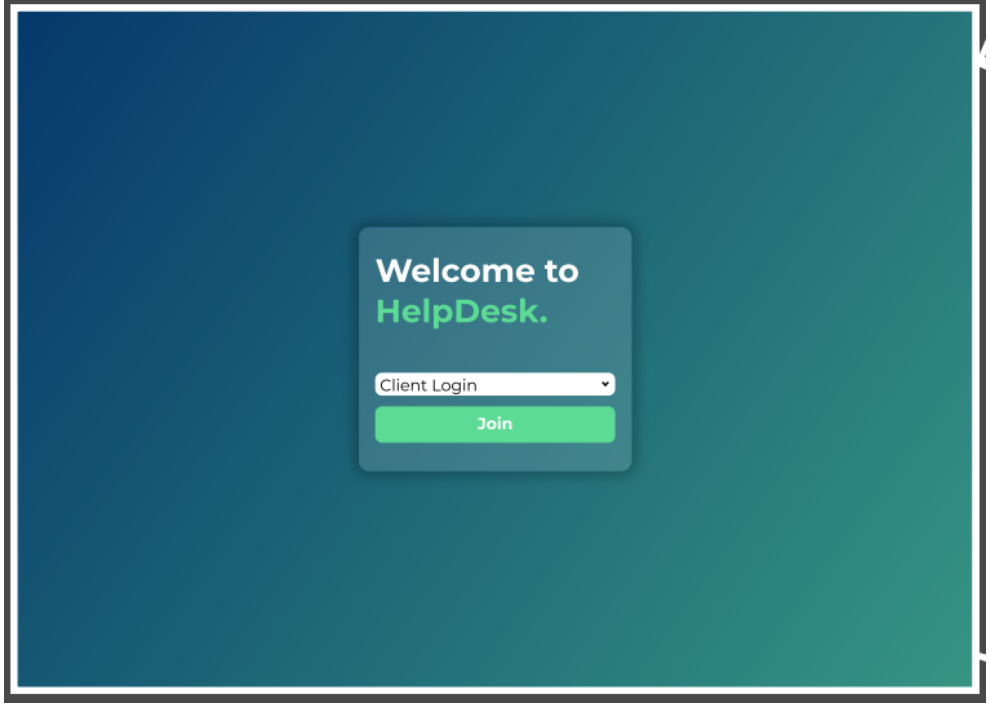
7.1.2 Welcome Page Stages to Final Design



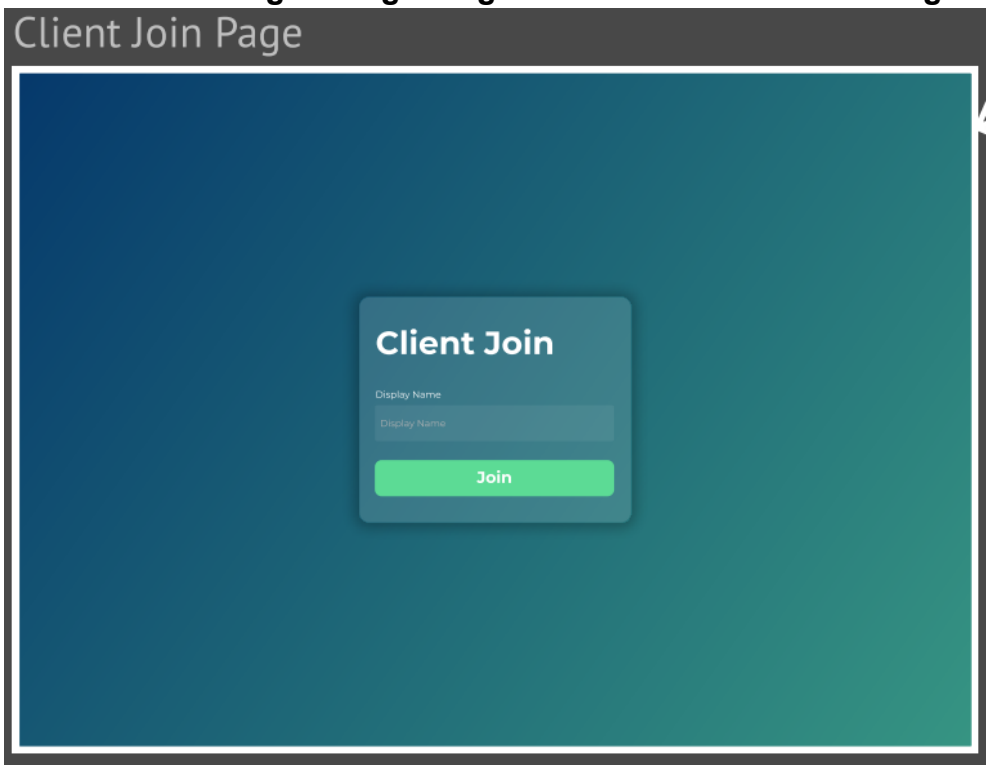
Welcome Page: Final Sketch

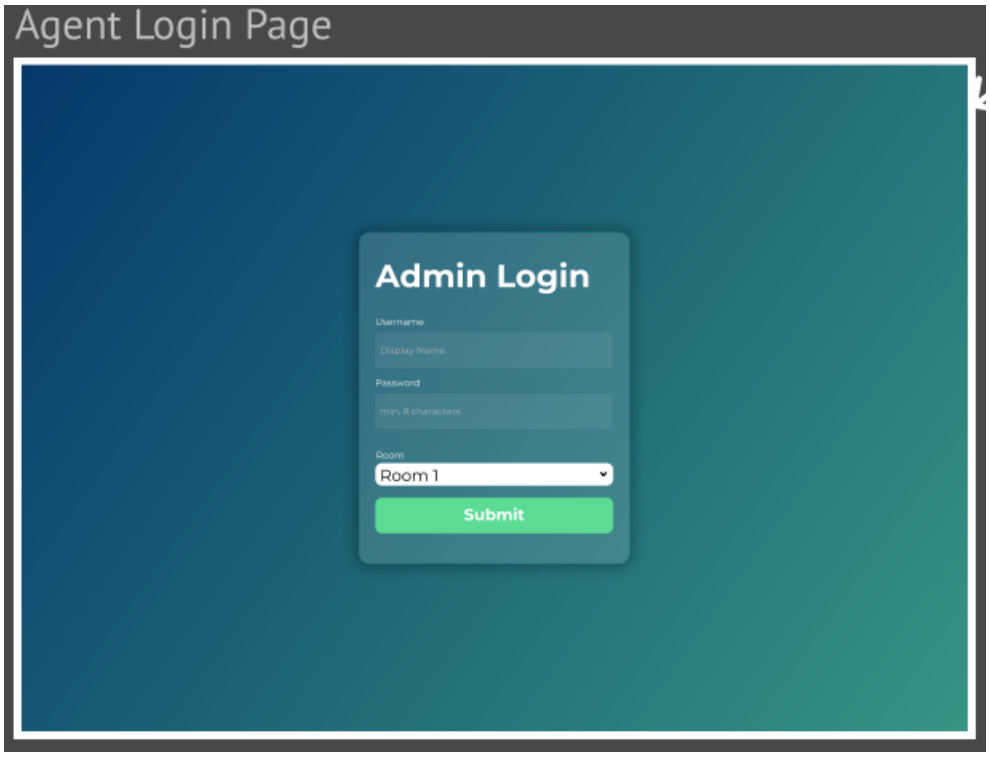


Welcome Page

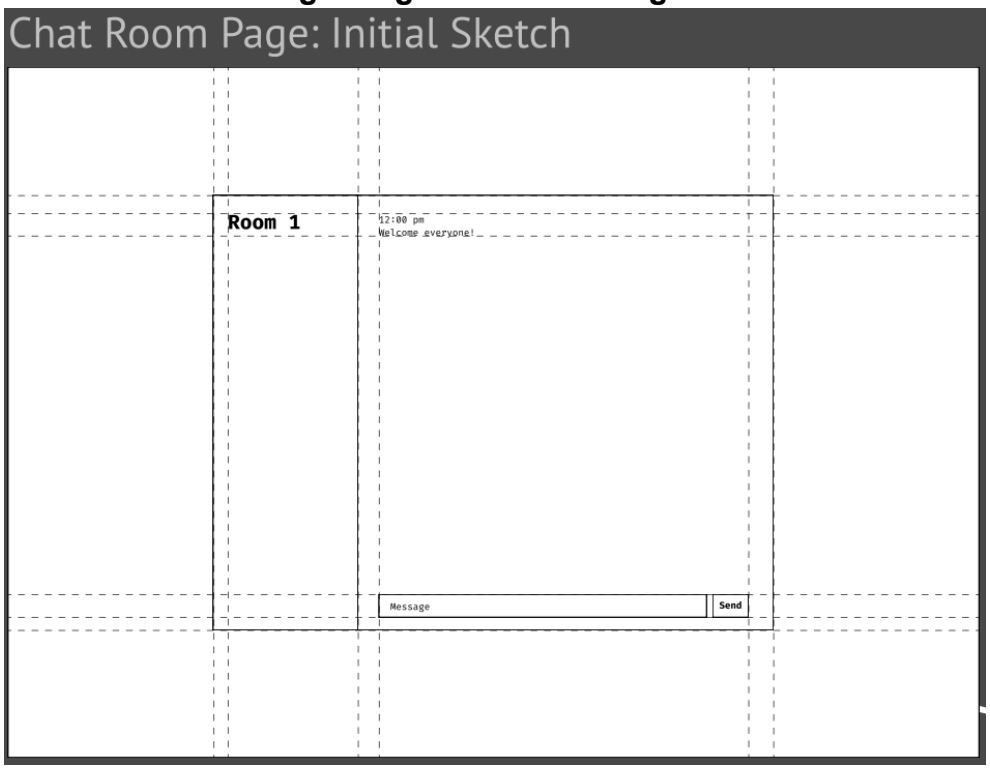


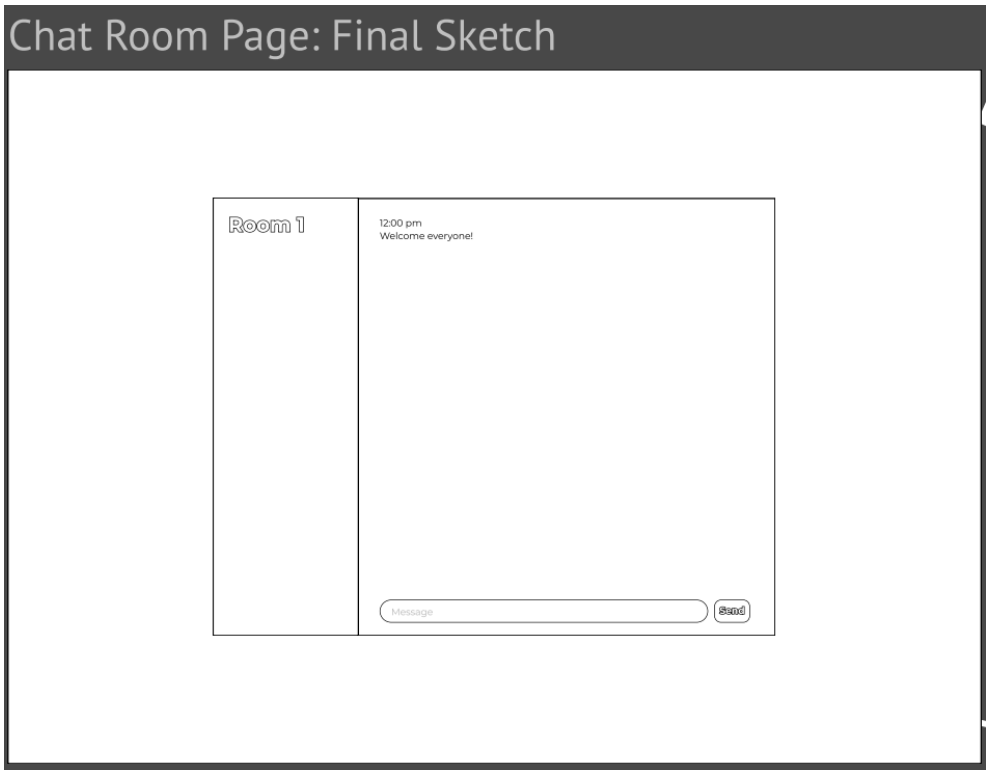
7.1.3 Client and Agent Login Pages Based off of Welcome Page

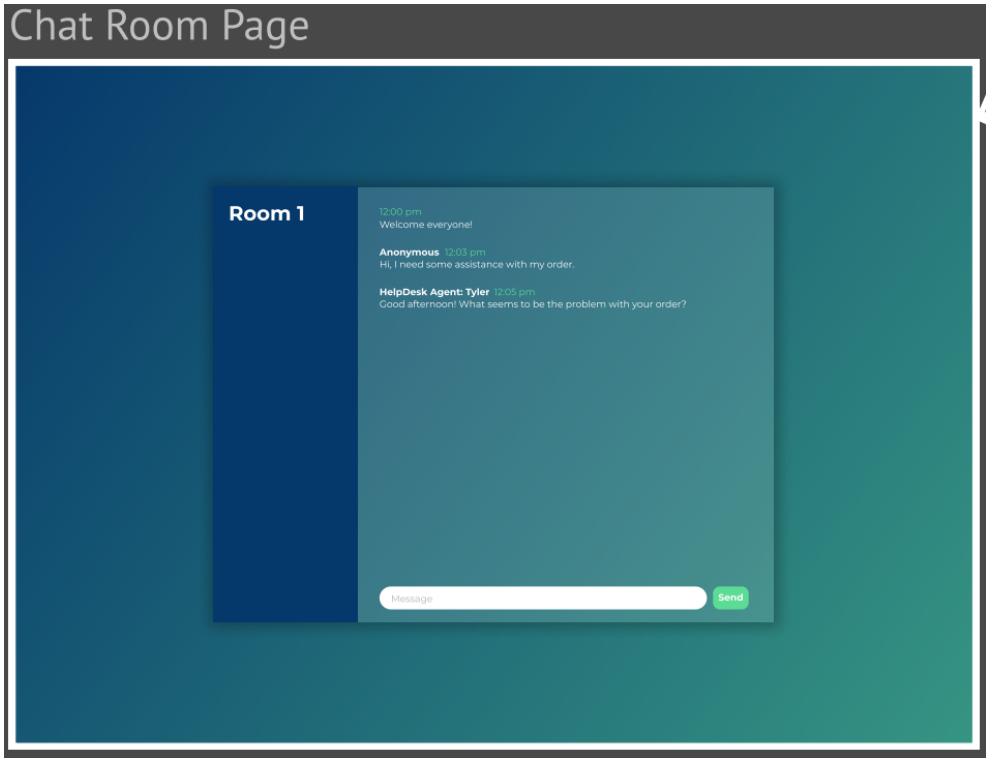




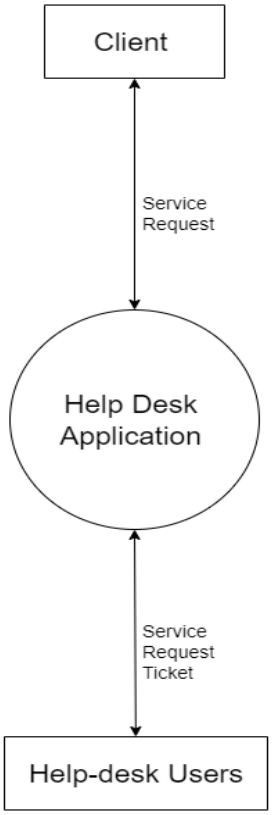
7.1.4 Chat Room Page Stages to Final Design



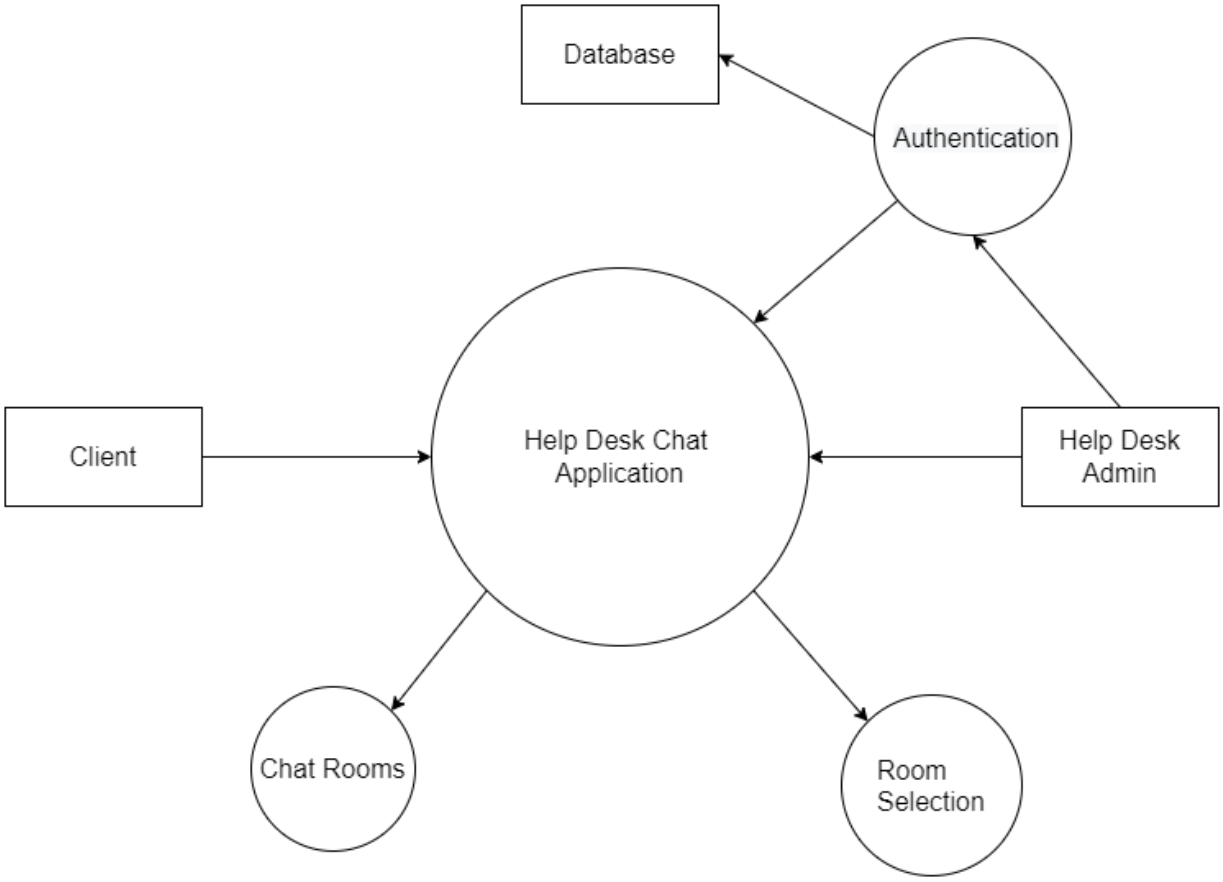




7.2 Context Data Flow Diagram

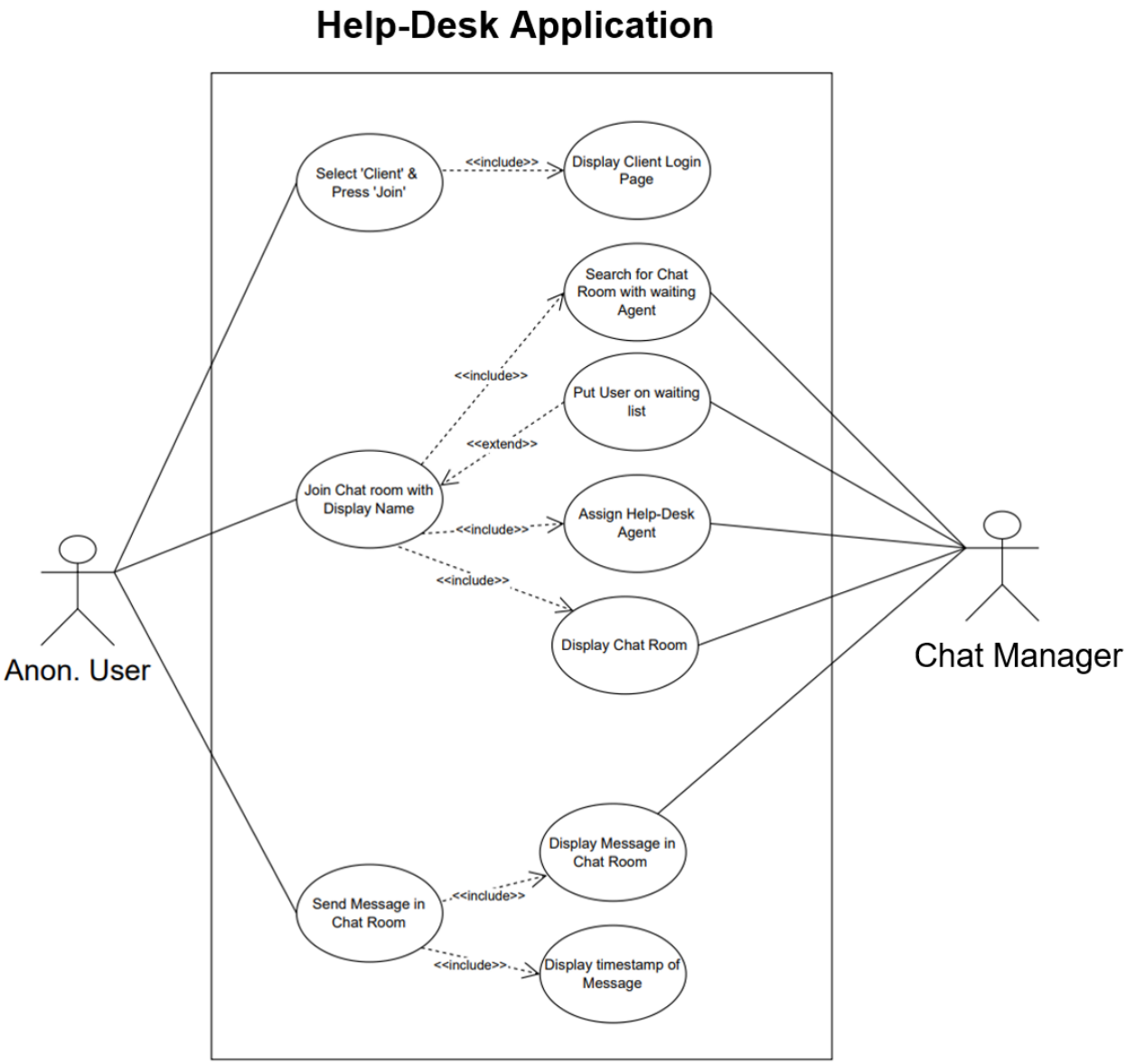


7.3 Level-1 Data Flow Diagram

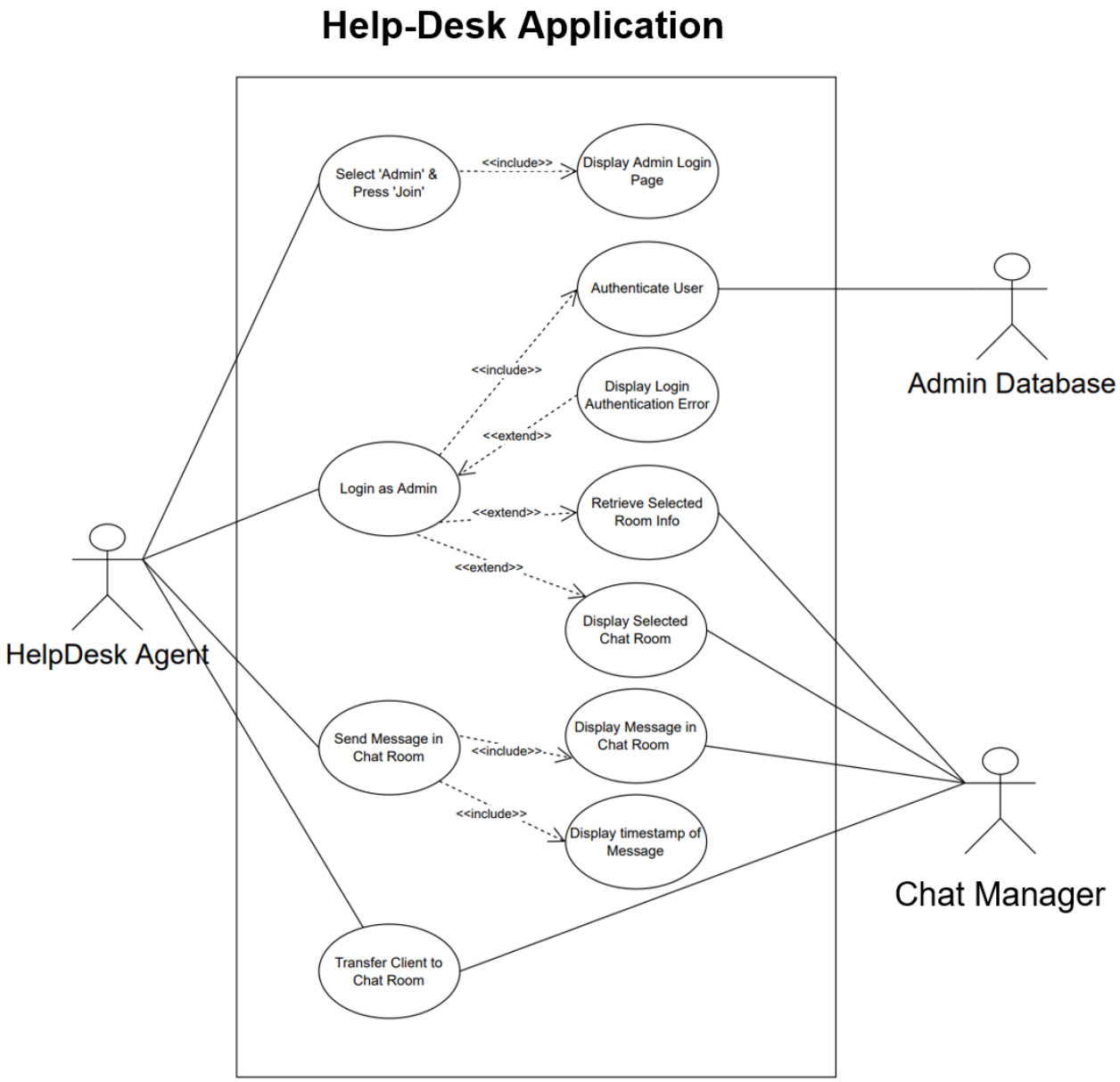


7.4 Use Case Diagrams

7.4.1 Anonymous User Use Case Diagram



7.4.2 Help-desk User Use Case Diagram



7.5 Entity Relationship Diagram

helpdeskdb.credentials	
PK	_id ObjectId NOT NULL
	username String NOT NULL password String NOT NULL

7.6 Component Diagram