Closet Cloud

Final Year Project

Session 2021-2024

A project submitted in partial fulfillment of the degree of

BS in Computer Science

Department of Software Engineering

Faculty of Computer Science & Information Technology

The Superior University, Lahore

Fall 2024

Type (Nature of project)	[✓] **D**evelopment [ ] **R**esearch [ ] **R**&**D**
Area of specialization	
FYP ID	

Project Group Members

Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	Bcsm-s20-050	Talha Khawar	bcsm-s20-050@superior.edu.pk	
(ii)	Bcsm-s21-032	Ghamamah Athar	bcsm-s21-032@superior.edu.pk	
(iii)				

The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Plagiarism Free Certificate

This is to certify that, I **Talha Khawar** S/D of **Khawar Javed**, group leader of FYP under registration no **BCSM-S20-050** at Computer Science Department, The Superior College, Lahore. I declare that my FYP report is checked by my supervisor.

Date: **05/12/24** Name of Group Leader: Talha Khawar Signature: _____

Name of Supervisor: Adqas Tanvir Co-Supervisor: Mr. XYZ

Designation: Lecturer Designation: Associate Professor

Signature: _____ Signature: _____

HoD: Dr. Muhammad Azam

Signature: _____

Project Report

Closet Cloud

Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
Talha Khawar, Ghamamah Athar	1.0	05/02/2024	Initiated project by collecting relevant data and resources.	
Talha Khawar, Ghamamah Athar	1.1	15/03/2024	Commenced development of the approved project with initial setup and planning.	
Talha Khawar, Ghamamah Athar	1.2	05/04/2024	Completed the design phase and prepared the project structure.	
Talha Khawar, Ghamamah Athar	1.3	05/05/2024	Conducted preliminary testing on the core features of the project.	
Talha Khawar, Ghamamah Athar	2.0	05/08/2024	Incorporated additional features based on user feedback and initial testing results.	
Talha Khawar, Ghamamah Athar	2.1	25/08/2024	Presented the project prototype to the supervisor for evaluation and feedback.	
Talha Khawar, Ghamamah Athar	2.2	05/09/2024	Made improvements and resolved issues as per feedback from the supervisor and stakeholders.	
Talha Khawar, Ghamamah Athar	2.3	25/11/2024	Finalized the project and prepared the documentation for submission.	

APPROVAL

PROJECT SUPERVISOR

Comments: _____

_____

Name:_____

Date:_____ Signature:_____

PROJECT MANAGER

Comments: _____

_____

Date:_____ Signature:_____

HEAD OF THE DEPARTMENT

Comments: _____

_____

Date:_____ Signature:_____

Dedication

This initiative aims to overcome the everyday wardrobe management challenges faced by both fashion aficionados and busy persons. I hope that the Cloud Closet platform serves as a source of inspiration, organization, and inventiveness, helping users navigate the confusing selection of clothes and confidently convey their own sense of style. Allow this project to function as your online closet buddy, easing the hassles of choosing an outfit and creating a community where fellow fashion enthusiasts can interact, share, and be inspired. Let's look forward to a time when individuals may effortlessly express their own style and proudly go through their clothes.

Acknowledgements

We would like to extend our sincere gratitude to our project supervisor for his or her great guidance, perceptive criticism, and unwavering support during the Cloud Closet project. We also want to express our profound appreciation to everyone who volunteered to take part in user testing; their insightful comments and helpful criticism allowed us to improve and fine-tune our platform. Their commitment and cooperation have been essential to the project's success, guaranteeing that Cloud Closet will develop into a more comprehensive, user-friendly, and practical option for fashion fans everywhere.

Executive Summary

The goal of the Cloud Closet project is to transform personal styling and wardrobe management by introducing a comprehensive platform. With its novel features, which include outfit creation, closet planning, user authentication, and item upload, our system provides customers with effortless management, outfit planning, and style inspiration. Cloud Closet enables customers to effortlessly browse their clothes collections, create unique ensembles, and remain up to speed with the newest fashion trends by utilizing cutting-edge technology and AI-driven algorithms. Cloud Closet is ready to revolutionize the way people interact with their clothes in the digital age with its user-centric approach and careful project planning.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Chapter 1: Introduction

Many people find it difficult to properly manage their clothes in the fast-paced world of today. Making sense of the world of fashion may be difficult, from disorganized wardrobes to the never-ending struggle of what to wear. An example of creativity and pragmatism, the Cloud Closet initiative acknowledges these difficulties and rises to the occasion. By using technology, our platform hopes to transform wardrobe management by giving customers effortless organizing, outfit planning, and style inspiration.

The Cloud Closet platform is thoroughly examined in this project report, which also explores the underlying technology, salient features, and extensive project plan developed for the platform's effective deployment. Cloud Closet enables customers to seamlessly explore their clothes collections, create unique ensembles, and stay up to date with the newest fashion trends by utilizing state-of-the-art developments in AI and user-centric design concepts.

1.1 Background

The idea for the Cloud Closet concept came from the realization that many people struggle to properly manage their outfits. People frequently feel overwhelmed by their messy wardrobes and the difficult chore of choosing what to wear in today's fast-paced world. The goal of this project is to provide a contemporary solution by utilizing technology to address these issues. Our team has created a complete platform that transforms wardrobe management via intensive study and teamwork. Through the integration of sophisticated features like outfit creation, virtual closet management, and AI-powered outfit recommendations, Cloud Closet aims to streamline the process and enable users to seamlessly express their individual style.

1.2 Motivations and Challenges

Motivation:

The motivation behind developing the Cloud Closet web application stems from the desire to simplify wardrobe management for individuals with busy schedules. By providing an all-inclusive platform with user-friendly features, we aim to empower customers to efficiently organize their wardrobes and create customized looks according to their preferences and schedules. Our goal

is to enhance the overall experience of managing clothing and styling, ultimately making it more convenient and enjoyable for users to express their personal style.

Challenges:

Developing a successful Cloud Closet web application comes with its own set of challenges. One major challenge is creating reliable algorithms for organizing virtual closets effectively. This requires intricate problem-solving skills and a deep understanding of user behavior and preferences. Additionally, seamlessly integrating the application with various mobile platforms poses another challenge, as it demands technical expertise and compatibility across different devices and operating systems. Furthermore, ensuring the accuracy of outfit recommendations presents a significant challenge, as it involves analyzing vast amounts of data and continuously refining the recommendation algorithms. Overcoming these obstacles will require teamwork, creative problem-solving, and meticulous attention to detail. Despite the challenges, our commitment to simplifying wardrobe management and enhancing individual style experiences drives us to overcome these hurdles and succeed with the Cloud Closet initiative.

1.3 Goals and Objectives

The Cloud Closet project aims to develop a comprehensive platform that transforms personal styling and wardrobe management. The following are the endeavor's goals:

- Creating strong algorithms for outfit generation, virtual closet organizing, and outfit recommendations.
- Creating an interface that is simple to use and intuitive for smooth interaction and navigation.
- Using AI-driven technology to correctly monitor fashion trends, weather, and consumer preferences.
- Verifying and testing the platform to make sure it is accurate, dependable, and user-friendly.

By achieving these goals, we hope to improve customers' entire wardrobe management experience by enabling them to easily arrange their clothes collections, make customized ensembles, and keep up with the newest fashion trends.

1.4 Literature Review/Existing Solutions

For the most part, internet platforms and smartphone applications providing rudimentary organizing features have been the mainstays of current wardrobe management systems. These programs help a little, but they frequently don't have all the functionality needed to handle the complexity of managing a modern wardrobe. Nor do many of the current systems provide customers any insightful analysis or recommendations; instead, they concentrate only on cataloging clothing items.

Some programs let users organize their virtual closets by classifying their clothes according to season, kind, or color. Nevertheless, sophisticated features like AI-driven outfit recommendations or outfit design are frequently absent from these systems. Furthermore, there are currently no tools for wardrobe management.

The goal of the Cloud Closet project is to overcome these constraints by offering an all-inclusive and user-friendly platform that transforms wardrobe management. Cloud Closet gives customers a comprehensive way to simplify their wardrobe management process by integrating cutting-edge features including, outfit creation, virtual closet organizing, and outfit recommendations. It is clear from a review of previous studies and solutions in the area of wardrobe management that a more advanced and approachable platform is required, and this is what the Cloud Closet project aims to provide.

1.5 Gap Analysis

The market is noticeably lacking in complete wardrobe management platforms with sophisticated features like outfit creation, AI-powered outfit recommendations. Current solutions frequently do not have these features, so users are left without a comprehensive approach to organizing and styling their wardrobes. The Cloud Closet project aims to close this gap by offering a user-friendly platform that meets the various needs of people looking for effective and cutting-edge wardrobe management solutions.

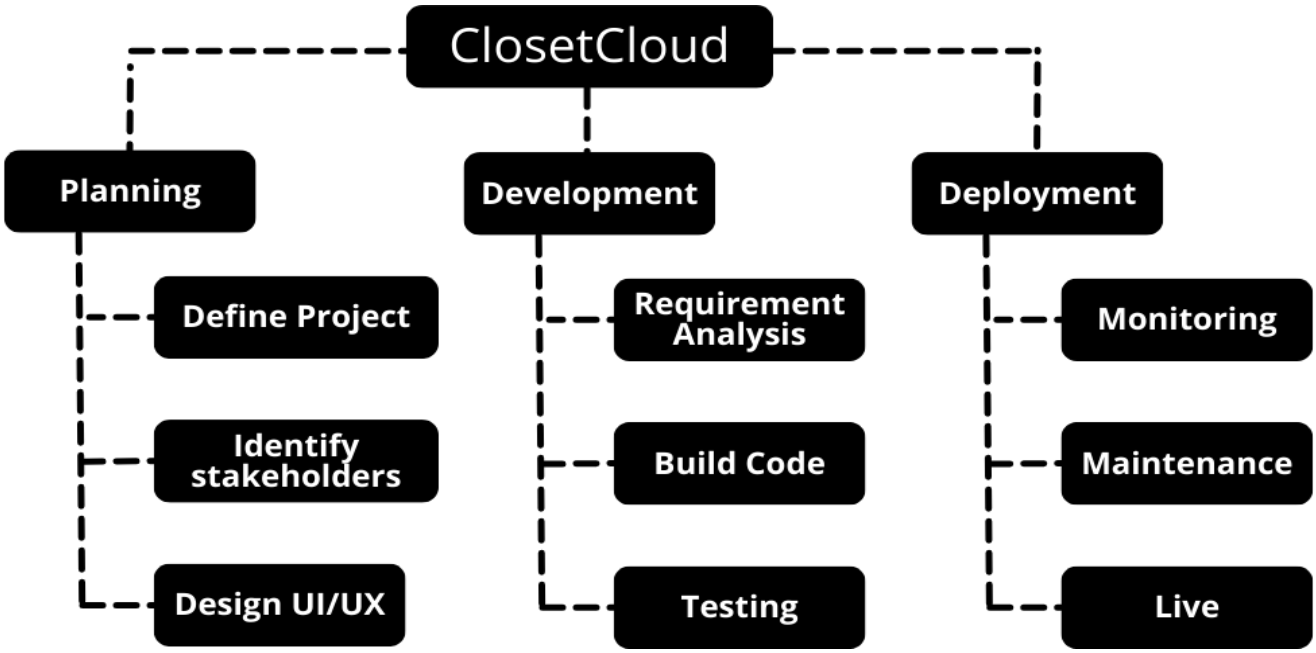
1.6 Proposed Solution

A feature-rich online application intended to improve individual style experiences and expedite wardrobe maintenance is the suggested solution for the Cloud Closet project. Advanced features

like outfit creation, AI-driven outfit recommendations, and virtual closet management will all be included in this program. Users will find it easy to plan their wardrobe for different situations, make customized ensembles, and arrange their clothes collections thanks to the user-friendly design and seamless navigation. With the use of innovative technology and user-centered design concepts, the Cloud Closet platform hopes to completely transform how people interact with their closets in the digital age.

1.7 Project Plan

1.7.1 Work Breakdown Structure



1 WBS Structure

Table 1 WBS

WBS #	WBS Deliverable	Activity #	Activity to Complete the Deliverable	Duration (# of Days)	Responsible Team Member(s) & Role(s)
1	**Planning**	1	Define project objectives	3	Project Manager
		2	Identify stakeholders	2	Project Manager
		3	Design UI/UX	5	UI/UX Designer
2	**Development**	4	Requirement Analysis	10	Project Manager, Developers
		5	Build Code	45	Developers
		6	Testing	10	SQA Engineer
3	**Deployment**	7	Monitoring	10	Project Manager, Deployment Manager
		8	Maintenance	Ongoing	Developers, Support Specialist
		9	Live	Ongoing	Deployment Manager, User Support
4	**Project Documentation**	10	Create User Manuals	7	Technical Writer
		11	Technical Documentation	Ongoing	Technical Writer

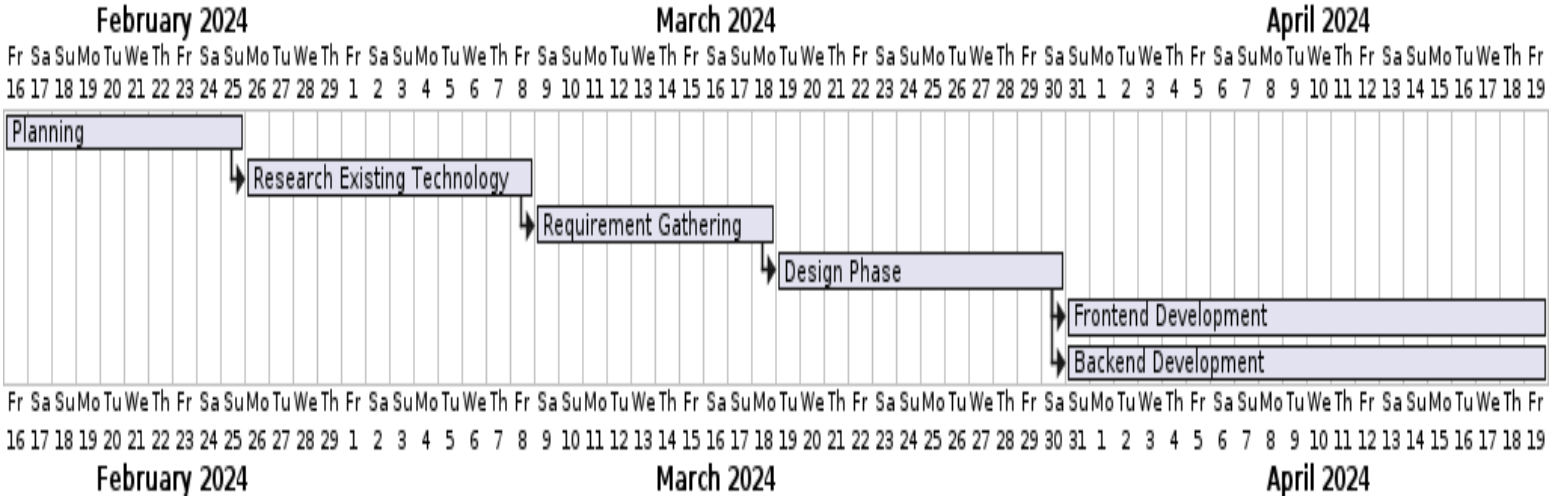
1.7.2 Roles & Responsibility Matrix

Table 2 Roles & Responsibility Matrix

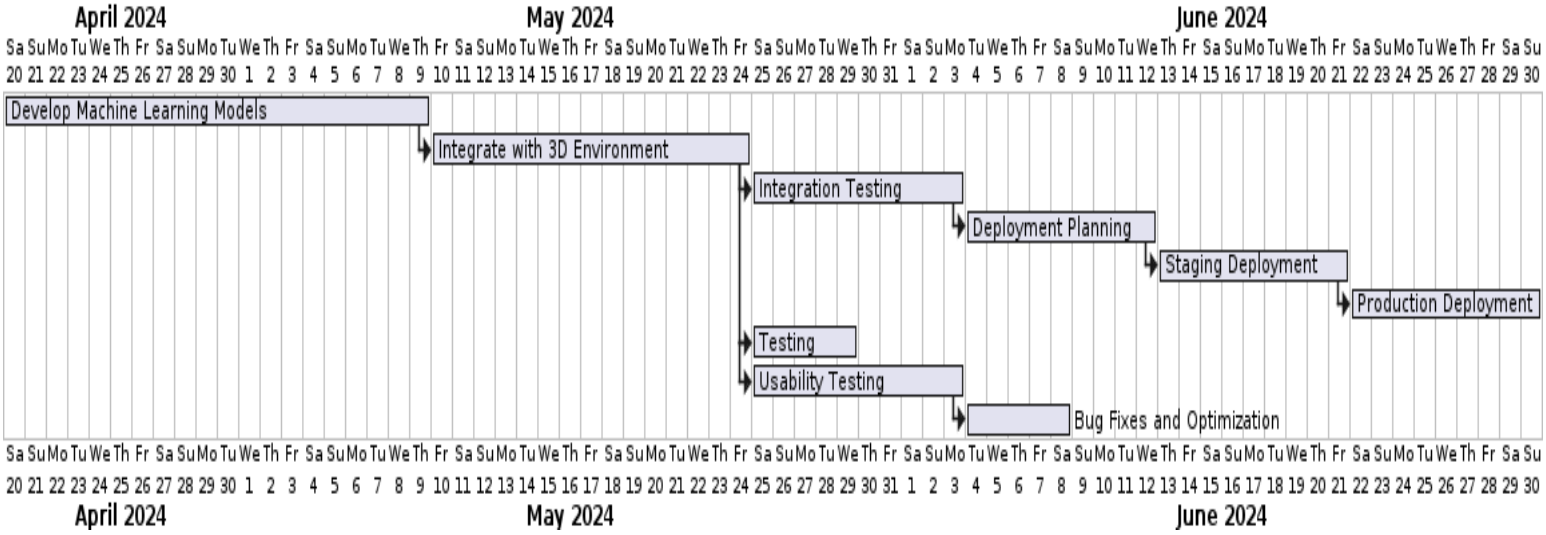
Activity	Activity to be Delivered	Roles	Responsibilities	Project Roles	Duration (days)	Project Team Members
Define project	Project Objectives	Project Manager	Define project scope, goals, objectives	Project Manager	3	Project Manager
Identify stakeholders	Stakeholder analysis	Project Manager	Identify project stakeholders	Project Manager		Project Manager
Design UI/UX	UI/UX designs	UI/UX Designer	Design user interface and experience	UI/UX Designer	15	UI/UX Designer
Requirement Analysis	Requirements document	Project Manager, Developer	Analyze project requirements	Project Manager, Developer	10	Project Manager, Developer
Build Code	Functional code	Developers	Develop code based on requirements	Developers	45	Developers
Testing	Tested application	SQA Team	Conduct tests to ensure functionality	SQA Team	14	SQA Team
Monitoring	Monitoring reports	Deployment Manager, Support Team	Monitor application performance	Deployment Manager, Support Team	Ongoing	Deployment Manager, Support Team
Maintenance	Bug fixes, updates	Maintenace Team	Address reported	Maintenace Team	Ongoing	Maintenace Team

			issues, implement updates			
Live	Live application	Deployment Manager	Ensure application is live and running	Deployment Manager		Deployment Manager
Create User Manuals	User manuals	Technical Writers	Create user manuals for application	Technical Writers	10	Technical Writers
Technical Documentation	Technical documentation	Technical Writers	Document technical aspects of application	Technical Writers	Ongoing	Technical Writers

1.7.3 Gantt Chart



2 Gantt Chart

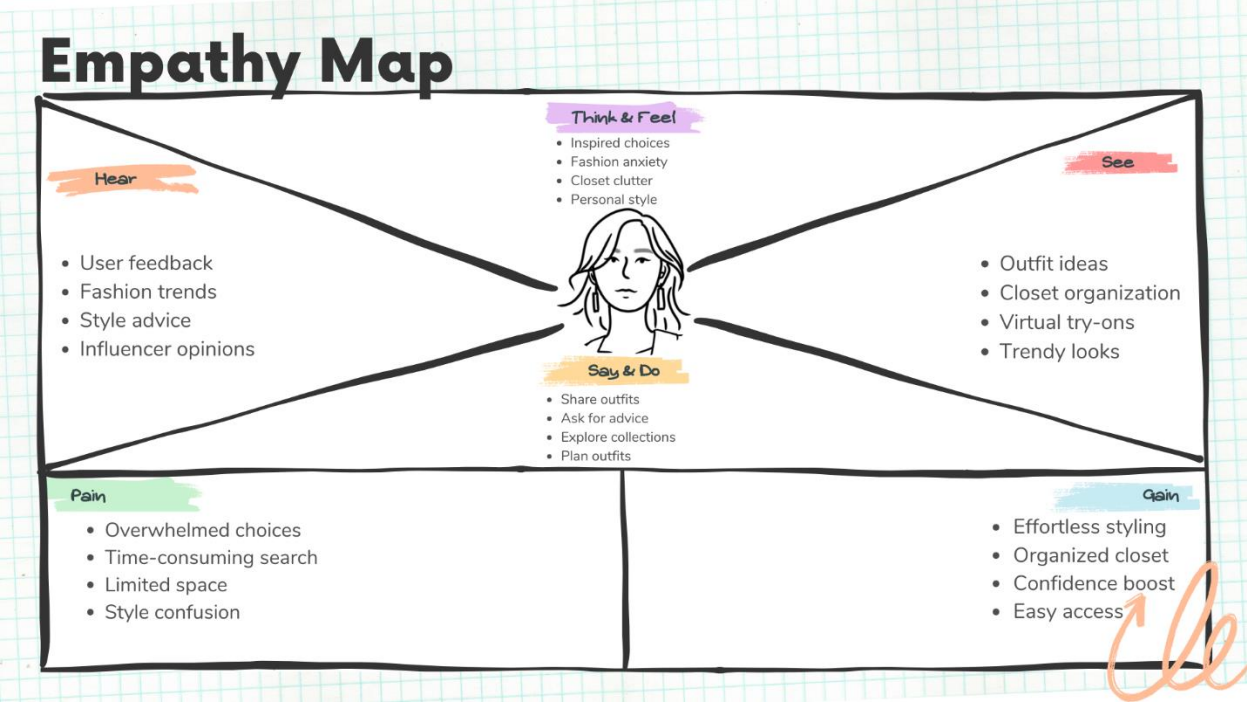


1.8 Report Outline

- **Introduction**
  - Background and Motivation: Discuss the challenges in wardrobe management and the need for a modern solution like ClosetCloud.
  - Methodology: Provide an overview of the approach taken in developing ClosetCloud.
- **Requirements Gathering**
  - Functional and Non-functional Requirements: Detail the features and characteristics expected from ClosetCloud.
  - Stakeholder Analysis: Identify the key stakeholders and their roles in the project.
- **System Design**
  - Architecture Design: Explain the overall structure and components of ClosetCloud.
  - User Interface Design: Describe the design principles and user experience considerations.
  - Database Design: Outline the database schema and data management strategies.
- **Implementation**
  - Technologies and Tools Used: List the programming languages, frameworks, and software used in developing ClosetCloud.
  - Testing and Quality Assurance: Discuss the testing methodologies and quality assurance practices employed.

- **User Acceptance Testing**
  - Results and Analysis: Present the findings from user acceptance testing, including performance metrics and user satisfaction.
  - User Feedback and Suggestions: Discuss the feedback received from users and any improvements suggested.
- **References**
  - List of Sources Used in the Report: Include all references, citations, and resources consulted during the project.

1.9 Empathy Map



3 Empathy Map

Chapter 2

Software Requirement Specifications

Chapter 2: Software Requirement Specifications

2.1 Introduction

2.1.1 Purpose

The Closet Cloud project aims to provide a complete and intuitive software platform that transforms personal styling and wardrobe management. Closet Cloud strives to streamline the process of managing clothes collections, designing customized ensembles, and keeping up with the newest fashion trends by utilizing cutting-edge technology and creative design ideas. The goal of this platform is to enable users to organize their wardrobes with ease, express their stylistic inventiveness, and improve their entire experience with fashion. Closet Cloud's ultimate goal is to be the go-to partner for anybody looking for effective and fun clothing management solutions.

2.1.2 Document Conventions

This document adheres to the IEEE 830-1998 standard for Software Requirements Specifications (SRS). It follows the recommended structure and format outlined in the standard, including sections such as Introduction, Scope, Functional Requirements, Non-Functional Requirements, and more. Additionally, it employs clear and concise language, uses numbered headings for easy reference, and includes tables and diagrams to enhance readability and comprehension. All requirements are specified using standard terminology and are categorized and prioritized as per the IEEE 830-1998 guidelines. Moreover, this document is intended to serve as a comprehensive reference for stakeholders involved in the development of the Closet Cloud project, ensuring clarity, consistency, and traceability of software requirements throughout the project lifecycle.

2.1.3 Intended Audience and Reading Suggestions

The intended audience for this document encompasses project stakeholders, including investors, clients, project managers, developers, testers, and any individuals directly involved in the development, deployment, and maintenance of the Closet Cloud platform. It is essential for all stakeholders to read this SRS document in its entirety to obtain a comprehensive understanding of the project requirements, objectives, and scope. Developers and testers will benefit from detailed insights into functional and non-functional requirements, while project managers and

stakeholders will gain valuable insights into project scope, timelines, and deliverables. Additionally, clients and investors will find this document instrumental in aligning their expectations with project outcomes and ensuring that the final product meets their business objectives and user needs. Therefore, it is strongly recommended that all stakeholders thoroughly review this document to facilitate effective communication, collaboration, and successful project implementation.

2.1.4 Product Scope

The Closet Cloud project involves the development of an advanced web application aimed at transforming wardrobe management and personal styling experiences. The core functionality of the application is centered around efficiently organizing clothing collections, simplifying outfit creation, and delivering personalized styling recommendations. Users will have the ability to upload images of their clothing items directly to the web application, where they will be automatically categorized, organized, and stored for easy access. Additionally, Closet Cloud will enable users to mix and match clothing items to create customized outfits, which can be saved and labeled for future reference. Leveraging AI-driven algorithms, the application will provide tailored outfit suggestions based on user preferences, weather forecasts, and current fashion trends. Accessible through web browsers, the application will offer a seamless and intuitive user experience across various devices and platforms. Overall, Closet Cloud aims to streamline wardrobe management processes, empower users to express their individual style, and elevate their overall fashion experience through an innovative web-based solution.

2.1.5 References

- I. Al-Omar, N. N., Al-Rashed, N. M., Al-Fantoukh, H. I., al-Osaimi, R. M., Al-Dayel, A. H. A., & Mostefai, S. (2013). The design and development of a web-based virtual closet: The smart closet project. Journal of Advanced Management Science, 1(1), 124-128.

This paper discusses the design and implementation of a web-based virtual closet system known as the Smart Closet Project. The system aims to provide users with a convenient way to manage their clothing and outfit choices digitally. The authors describe the development process and features of the virtual closet, highlighting its potential benefits for users in terms of organization and decision-making regarding clothing selection.

- II. Li, Y., & Hu, W. (2014). The Conceptual Design of" Smart Closet" Fashion Consultant Expert System. International Journal of Business and Social Science, 5(10).

This paper presents the conceptual design of a "Smart Closet" Fashion Consultant Expert System. The system is intended to assist users in making fashion-related decisions by providing personalized recommendations based on their preferences, body measurements, and wardrobe contents. The authors outline the key components and functionalities of the expert system, emphasizing its potential utility in enhancing users' fashion choices and overall wardrobe management.

2.2 Overall Description

2.2.1 Product Perspective

The ClosetCloud web application is conceived as a standalone product that operates independently without the need for additional software or hardware dependencies. Users will access the application directly through web browsers, eliminating the requirement for any specific software installations or hardware configurations. The application will be hosted on web servers, ensuring seamless accessibility for users across various devices and platforms. Users can navigate and utilize the features of the ClosetCloud web application without reliance on any external systems or resources. Furthermore, the web application will be designed to offer a consistent and intuitive user experience across different web browsers and operating systems. It will be accessible for download from popular web browsers' app stores, ensuring widespread availability and ease of access for users. Overall, the ClosetCloud web application will serve as a self-contained solution for efficient wardrobe management and personalized styling experiences.

2.2.2 User Classes and Characteristics

- **Regular Users:**
  - **Characteristics:** These users are the primary audience of the ClosetCloud web application. They may include individuals of varying ages, backgrounds, and fashion preferences who use the application frequently for organizing their wardrobe, creating outfits, and seeking style inspiration.

- **Frequency of Use:** Regular users are expected to access the application frequently, potentially on a daily basis, to manage their clothing collections and plan outfits for various occasions.
- **Technical Expertise:** Regular users may have varying levels of technical expertise, but they are generally familiar with basic web navigation and usage of web-based applications.
- **Product Functions Used:** Regular users utilize a wide range of product functions offered by the ClosetCloud application, including clothing item upload, virtual closet organization, outfit creation, outfit recommendations, and more.
- **Importance:** Regular users are the most important user class for this product, as their satisfaction and engagement drive the success and adoption of the ClosetCloud web application.

➢ **Fashion Enthusiasts:**

- **Characteristics:** These users are passionate about fashion and style. They may include fashion bloggers, influencers, or individuals with a keen interest in experimenting with different looks and trends.
- **Frequency of Use:** Fashion enthusiasts may use the application frequently, exploring various styling options, experimenting with outfit combinations, and sharing their fashion creations with their audience.
- **Technical Expertise:** Fashion enthusiasts may possess above-average technical expertise, particularly in using web-based applications.
- **Product Functions Used:** Fashion enthusiasts heavily utilize features such as outfit creation, outfit recommendations sharing to express their creativity and inspire others.
- **Importance:** While not as numerous as regular users, fashion enthusiasts play a crucial role in driving engagement and promoting the ClosetCloud application within fashion communities.

➢ **Occasional Users:**

- **Characteristics:** These users may use the application sporadically for specific purposes, such as planning outfits for special events or updating their wardrobe seasonally.

- **Frequency of Use:** Occasional users may access the application infrequently, typically during periods of wardrobe updates or when preparing for specific occasions.
- **Technical Expertise:** Occasional users may have varying levels of technical expertise, ranging from basic to moderate proficiency in using web-based applications.
- **Product Functions Used:** Occasional users may primarily utilize features such as clothing item upload, virtual closet organization, and outfit planning for specific events or occasions.
- **Importance:** While less frequent than regular users, occasional users still contribute to the overall usage and adoption of the ClosetCloud application. Their needs and preferences should be considered to ensure a positive user experience for all user classes.

2.2.3 Operating Environment

The ClosetCloud web application will operate within a modern web-based environment, leveraging standard hardware and software components to deliver a seamless user experience. The operating environment for the ClosetCloud web application includes:

- **Hardware Platform:**
  - The application will run on standard hardware components commonly found in desktop and laptop computers, as well as mobile devices such as smartphones and tablets.
  - Hardware specifications may vary depending on the user's device, but the application is designed to be compatible with a wide range of hardware configurations.
- **Operating System and Versions:**
  - The ClosetCloud web application is compatible with major operating systems, including:
    - Windows: Versions 7, 8, and 10
    - macOS: Latest versions supported by Apple
    - Android: Latest versions supported by Google
    - iOS: Latest versions supported by Apple
  - The application is designed to operate seamlessly across different operating systems, ensuring a consistent user experience for all users.
- **Web Browsers:**
  - The application is accessible through modern web browsers, including but not limited to:

- Google Chrome
- Mozilla Firefox
- Safari
- Microsoft Edge
- Opera

  - The ClosetCloud web application is optimized for compatibility with the latest versions of these web browsers, ensuring optimal performance and functionality.

- **Other Software Components or Applications:**
  - The ClosetCloud web application does not require any specific software components or applications to operate.
  - It is designed to peacefully coexist with other web-based applications and services, allowing users to seamlessly integrate ClosetCloud into their existing workflows and routines.

2.2.4 Design and Implementation Constraints

These constraints include:

- **Corporate Policies and Regulatory Requirements:**

  Compliance with corporate policies and regulatory standards related to data privacy, security, and accessibility must be adhered to throughout the development process.

- **Hardware Limitations:**

  Consideration of hardware limitations, such as varying device specifications and performance capabilities, may constrain design choices and implementation approaches to ensure optimal performance across different devices.

- **Technology and Tools:**

  Utilization of specific technologies, tools, and frameworks for web development, such as HTML5, CSS3, JavaScript, and popular front-end frameworks like React, may limit options available to developers.

- **Databases:**

  Requirement to use specific database technologies or adhere to database schema standards set forth by the organization may constrain database design and implementation decisions.

- **Security Considerations:**

  Adherence to security best practices and implementation of robust security measures, such as encryption of user data and secure authentication mechanisms, may impose constraints on system architecture and design.

- **Design Conventions and Programming Standards:**

  Compliance with design conventions and programming standards established by the organization or industry best practices may restrict certain design choices and implementation approaches to maintain consistency and code quality.

- **Maintenance Responsibility:**

  Consideration of maintenance responsibilities post-deployment, including whether the customer's organization or the development team will be responsible for maintaining the delivered software, may influence design decisions and implementation strategies.

2.2.5 Assumptions and Dependencies

- **Assumed Factors:**
  - **Availability of Third-Party Components:** The successful integration of third-party or commercial components into the ClosetCloud web application is assumed. Any delays or issues in obtaining necessary components may impact project timelines and deliverables.
  - **Stable Development and Operating Environment:** It is assumed that the development and operating environments will remain stable throughout the project duration. Any unexpected changes or disruptions to these environments could affect development progress and the overall project timeline.
  - **Adequate Resource Allocation:** The availability of sufficient resources, including human resources, time, and budget, is assumed for the successful completion of the project. Any constraints on resources may impact project execution and deliverables.
  - **Compliance with Regulatory Standards:** It is assumed that the ClosetCloud web application will comply with relevant regulatory standards and legal requirements related to data privacy, security, and accessibility. Any changes in regulatory requirements may necessitate adjustments to the application's design and functionality.

- **Dependencies:**
  - **Reusable Software Components:** The project may depend on the reuse of certain software components from other projects to streamline development efforts. Dependencies on these components must be clearly identified and managed to ensure compatibility and functionality.
  - **Integration with Web Browsers:** The application's compatibility and functionality may depend on the behavior of various web browsers and their support for web standards. Dependencies on specific browser features or behaviors must be considered during development and testing to ensure consistent performance across different browsers.

2.3 External Interface Requirements

2.3.1 User Interfaces

- **Login/Register Interface:**
  - This interface will enable users to create a new account or log in to an existing one.
  - Users will enter their credentials (username and password) or utilize alternative authentication methods to access the application.
- **Home Interface:**
  - The main menu will be displayed, providing users with options to access different functionalities of the ClosetCloud web application.
  - Users can navigate to various sections of the application, including virtual closet organization, outfit creation, outfit recommendations.
- **Virtual Closet Organization Interface:**
  - This interface will allow users to view and manage their clothing collections within virtual closets.
  - Users can categorize clothing items based on type, color, season, and occasion, using intuitive tools for efficient organization.
- **Outfit Creation Interface:**
  - Users can mix and match clothing items to create personalized outfits within this interface.

- Options to save and name outfits for future reference will be available, along with features for adjusting outfit compositions and visualizing ensemble combinations.

➢ **Outfit Recommendations Interface:**

- AI-driven algorithms will suggest outfit combinations tailored to user preferences, weather conditions, and current fashion trends.
- Users can explore personalized outfit recommendations and receive styling suggestions based on their virtual closet contents and style preferences.

➢ **Profile Interface:**

- This interface will display user information, including username, profile picture, and previously saved outfits and styling preferences.
- Users can update their profile information and manage their preferences within this interface.

2.3.2 Hardware Interfaces

➢ **Compatibility Across Devices:**

- The application will be designed to seamlessly integrate with the camera and microphone features of various devices, including desktop computers, laptops, smartphones, and tablets.
- Compatibility across different device types and operating systems (e.g., Windows, macOS, Android, iOS) will be ensured to accommodate diverse user preferences and hardware configurations.

➢ **Optimized Performance:**

- The hardware interface will prioritize optimized performance and responsiveness, ensuring smooth operation of camera and microphone functionalities within the ClosetCloud web application.
- Efforts will be made to minimize latency and enhance user experience during tasks such as capturing clothing item images.

➢ **User Permission Handling:**

- The application will request user permission to access the camera and microphone functionalities upon installation or first-time use.

- Clear and transparent communication will be provided to users regarding the purpose and usage of these hardware interfaces, along with options to grant or revoke permissions as desired.

➢ **Security and Privacy Considerations:**

- Strict security measures will be implemented to safeguard user privacy and prevent unauthorized access to camera and microphone data.
- Data encryption protocols and secure communication channels will be employed to protect sensitive information transmitted between the user's device and the ClosetCloud web application servers.

2.3.3 Software Interfaces

➢ **Database Connectivity:**

- The ClosetCloud web application will interact with a MongoDB database to store and retrieve user data, including login credentials, virtual closet contents, saved outfits, and user preferences.
- The application will utilize MongoDB's query language to communicate with the database and perform operations such as insertion, retrieval, update, and deletion of data records.
- Data items shared with the database include user profiles, clothing item details, outfit configurations, and user preferences.

➢ **Operating System Compatibility:**

- The web application will be compatible with various operating systems, including Windows, macOS, and Linux distributions, ensuring broad accessibility across different user devices.
- Compatibility testing will be conducted to verify application functionality and performance on different operating system versions and configurations.

➢ **Web Browser Compatibility:**

- The application will support major web browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari, ensuring consistent performance and usability across different browser platforms.

- Compatibility testing will ensure that the user interface and functionality are optimized for each supported browser, minimizing compatibility issues and providing a seamless user experience.

➢ **JavaScript Frameworks and Libraries:**

- Utilization of JavaScript frameworks and libraries such as React.js, or Vue.js will facilitate the development of dynamic and interactive user interfaces for the ClosetCloud web application.
- The Multer library will be used to handle file uploads for outfit images, allowing users to save and manage their outfit files efficiently.
- These frameworks will enable the implementation of features such as real-time updates, responsive design elements, and smooth user interactions.

➢ **API Integration:**

- Integration with external APIs, such as speech-to-text APIs for transcribing user feedback APIs for interpreting gestures, will enhance application functionality and user experience.

➢ **Data Sharing Mechanisms:**

- Shared data items across software components include user authentication tokens, session identifiers, and cached data for improved performance.
- Implementation constraints may specify the use of secure data transmission protocols, such as HTTPS, to protect sensitive data shared between client and server components.

2.3.4 Communications Interfaces

➢ **HTTP/HTTPS Protocol:**

- The ClosetCloud web application will utilize the Hypertext Transfer Protocol (HTTP) or its secure variant HTTPS for communication between the client-side application and the web server.
- All client-server interactions, including user authentication, data retrieval, and API requests, will adhere to the HTTP/HTTPS protocol standards.

➢ **RESTful API Endpoints:**

- The application will communicate with external APIs, such as speech-to-text via Representational State Transfer (REST) architecture.

- RESTful API endpoints will define the communication interface for sending requests (GET, POST, PUT, DELETE) to the external services and receiving responses in a structured format, typically JSON or XML.

- **WebSockets for Real-time Communication:**
  - WebSockets will be employed for establishing bidirectional communication channels between the web application and the server, enabling real-time updates and notifications.
  - WebSocket connections will facilitate instantaneous data exchange for features like live chat support, real-time updates to virtual closets, and collaborative outfit planning.
- **Email Notifications:**
  - The application may incorporate email communication for sending notifications, alerts, or updates to users, such as account verification emails, password reset links, or promotional offers.
  - Email communication will adhere to standard email protocols (e.g., SMTP, IMAP) for message delivery and formatting, ensuring reliable and consistent communication with users.
- **Security and Encryption:**
  - Communication interfaces will implement Transport Layer Security (TLS) encryption to secure data transmission between the client and server components.
  - HTTPS will be enforced to encrypt data in transit, protecting sensitive information such as user credentials and personal data from unauthorized access or interception.
- **Data Transfer Rates and Synchronization:**
  - The application will support efficient data transfer rates to minimize latency and optimize performance for users accessing the ClosetCloud web application.
  - Synchronization mechanisms will be implemented to ensure consistency and data integrity across distributed components, such as caching mechanisms for frequently accessed data and synchronization protocols for real-time updates.

2.4 System Features

2.4.1 Virtual Closet Organization

2.4.1.1 Description and Priority

The Virtual Closet Organization feature allows users to organize their clothing items seamlessly within the application. This feature is of high priority as it forms the foundation of the wardrobe management functionality.

2.4.1.2 Stimulus/Response Sequences

- User selects the "Organize Closet" option from the main menu.
- System displays the virtual closet interface with options to categorize clothing items.
- User selects desired categories (e.g., type, color, season) and arranges items accordingly.
- System updates the virtual closet based on user actions and displays the organized layout.

2.4.1.3 Functional Requirements

- **REQ-SF1-1:** The system shall provide options for users to categorize clothing items based on type (e.g., shirts, pants, dresses).
- **REQ-SF1-2:** Users shall have the ability to arrange clothing items within each category by dragging and dropping.
- **REQ-SF1-3:** The system shall support the categorization of clothing items based on color, allowing users to group items by similar hues.
- **REQ-SF1-4:** Users shall be able to organize clothing items by season, enabling them to separate items suitable for different weather conditions.

2.4.2 Outfit Creation

2.4.2.1 Description and Priority

- The Outfit Creation feature empowers users to mix and match clothing items to create personalized outfits, enhancing their styling experience. This feature is of high priority as it promotes user engagement and satisfaction.

2.4.2.2 Stimulus/Response Sequences

- User selects the "Create Outfit" option from the main menu.
- System displays the outfit creation interface with options to select clothing items.
- User selects desired items and arranges them to form an outfit.
- System provides a preview of the outfit and allows users to save or discard it.

2.4.2.3 Functional Requirements

- **REQ-SF2-1:** The system shall present users with a visual interface for selecting clothing items from their virtual closet.
- **REQ-SF2-2:** Users shall be able to mix and match clothing items by layering and combining different pieces.
- **REQ-SF2-3:** The system shall allow users to save created outfits for future reference, including the option to name and categorize them.
- **REQ-SF2-4:** Users shall have the ability to edit and modify saved outfits, including adding or removing items as desired.

2.4.3 Outfit Recommendations

2.4.3.1 Description and Priority

- The Outfit Recommendations feature utilizes AI-driven algorithms to suggest outfit combinations tailored to user preferences, weather conditions, and current fashion trends. This feature is of high priority as it enhances user engagement and provides valuable styling assistance.

2.4.3.2 Stimulus/Response Sequences

- User accesses the "Outfit Recommendations" section within the application.
- System prompts users to input their style preferences, such as color palettes, clothing styles, and occasions.
- User selects relevant criteria, such as weather conditions or event types.
- System generates personalized outfit recommendations based on user inputs and current fashion trends.

2.4.3.3 Function Requirements

- **REQ-SF3-1:** The system shall provide a user-friendly interface for users to input their style preferences and criteria for outfit recommendations.
- **REQ-SF3-2:** Users shall have the option to specify contextual factors such as weather conditions, occasions, or preferred clothing styles.
- **REQ-SF3-3:** The system shall utilize AI algorithms to analyze user preferences and fashion trends to generate personalized outfit suggestions.
- **REQ-SF3-4:** Users shall be presented with a variety of outfit combinations, including suggestions for tops, bottoms, accessories, and footwear.

REQ-SF3-5: The system shall allow users to refine outfit recommendations by providing feedback or adjusting input criteria.

2.5 Nonfunctional Requirements

2.5.1 Performance Requirements

- The Closet Cloud application should load within 3 seconds of accessing the website.
- Response times for user interactions, such as adding items to the virtual closet or generating outfit recommendations, should be under 1 second.
- The system should be able to handle concurrent user requests without significant degradation in performance, supporting a minimum of 1000 simultaneous users.

2.5.2 Safety Requirements

- The application should not compromise user privacy or security, adhering to industry-standard encryption protocols (e.g., TLS) for data transmission.
- Safety measures should be in place to prevent unauthorized access to user accounts and sensitive information.
- User-generated content, such as uploaded images or videos, should be screened for inappropriate or harmful content to maintain a safe environment for all users.

2.5.3 Security Requirements

- User authentication and authorization mechanisms should be implemented securely to prevent unauthorized access to user accounts and data.

- The application should employ robust security measures, such as strong password hashing and salting techniques, to protect user credentials.
- Regular security audits and vulnerability assessments should be conducted to identify and mitigate potential security threats.

2.5.4 Usability Requirements

- The user interface should be intuitive and user-friendly, requiring minimal training for users to navigate and utilize the application's features.
- Clear and concise instructions should be provided to guide users through common tasks, such as adding items to the virtual closet or creating outfits.
- Accessibility features, such as keyboard navigation and screen reader compatibility, should be implemented to accommodate users with disabilities.

2.5.5 Reliability Requirements

- The application should maintain high availability, with an uptime of at least 99.9% to ensure uninterrupted access for users.
- Error handling mechanisms should be in place to gracefully handle exceptions and prevent system crashes or downtime.
- Regular backups of user data should be performed to safeguard against data loss in the event of system failures or disasters.

2.5.6 Maintainability/Supportability Requirements

- The codebase should be well-documented and maintainable, allowing for easy debugging, troubleshooting, and future enhancements by development teams.
- Comprehensive user support resources, such as FAQs, tutorials, and user manuals, should be provided to assist users in resolving common issues or inquiries.
- System administrators should have access to tools and utilities for monitoring system performance, diagnosing issues, and applying updates or patches as needed.

2.5.7 Portability Requirements

- The application should be compatible with a wide range of devices and platforms, including desktop computers, laptops, tablets, and smartphones.

- Cross-browser compatibility should be ensured to support major web browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.
- The application should be responsive and adaptive, adjusting its layout and functionality based on the screen size and device orientation.

2.5.8 Efficiency Requirements

- The application should utilize system resources efficiently, minimizing CPU and memory usage to optimize performance and scalability.
- Code optimization techniques should be employed to reduce load times and improve overall responsiveness, particularly for resource-intensive operations such as image processing and AI algorithms.

2.6 Domain Requirements

- **Fashion Industry Compliance:**
  - The application should adhere to industry standards and regulations related to fashion and clothing retail, including copyright laws for branded apparel and intellectual property rights.
- **Data Privacy and Protection:**
  - The application must comply with data protection regulations, such as the General Data Protection Regulation (GDPR) or California Consumer Privacy Act (CCPA), ensuring user data privacy and security.
  - Personal data collection and processing should be transparent, and users must provide explicit consent for data usage and storage.
- **E-commerce Integration:**
  - If the application incorporates e-commerce functionalities, it should comply with relevant e-commerce regulations and standards, including secure payment processing, refund policies, and customer data protection.
- **Accessibility Standards:**
  - The application should adhere to accessibility standards, such as the Web Content Accessibility Guidelines (WCAG), to ensure inclusivity and accessibility for users with disabilities.

- Features such as keyboard navigation, screen reader compatibility, and adjustable font sizes should be implemented to accommodate diverse user needs.

➢ **Localization and Internationalization:**

- The application should support multiple languages and cultural preferences to cater to a global user base.
- Localization features, such as language selection options, currency conversion, and region-specific content, should be provided to enhance user experience for international users.

➢ **Ethical Considerations:**

- The application should uphold ethical principles in content moderation and user interactions, promoting a positive and respectful online environment.
- Measures should be in place to prevent discriminatory practices, harassment, or offensive content within the application.

➢ **User Data Retention Policy:**

- A clear data retention policy should be defined, outlining the duration for which user data will be stored and the purposes for which it will be used.
- Users should have the option to delete their account and personal data permanently from the system if desired, in compliance with data privacy regulations.

➢ **Third-Party Integrations:**

- Any third-party services or APIs integrated into the application should comply with relevant terms of service, licensing agreements, and usage policies.
- Integration with external services should be documented, and legal considerations, such as data sharing agreements and liability clauses, should be addressed.

➢ **Scalability and Performance:**

- The application architecture should be designed to accommodate scalability requirements, allowing for seamless expansion to handle increased user traffic and data volume over time.

- Performance metrics should be monitored regularly, and scalability measures, such as load balancing and server optimization, should be implemented to maintain optimal performance levels.

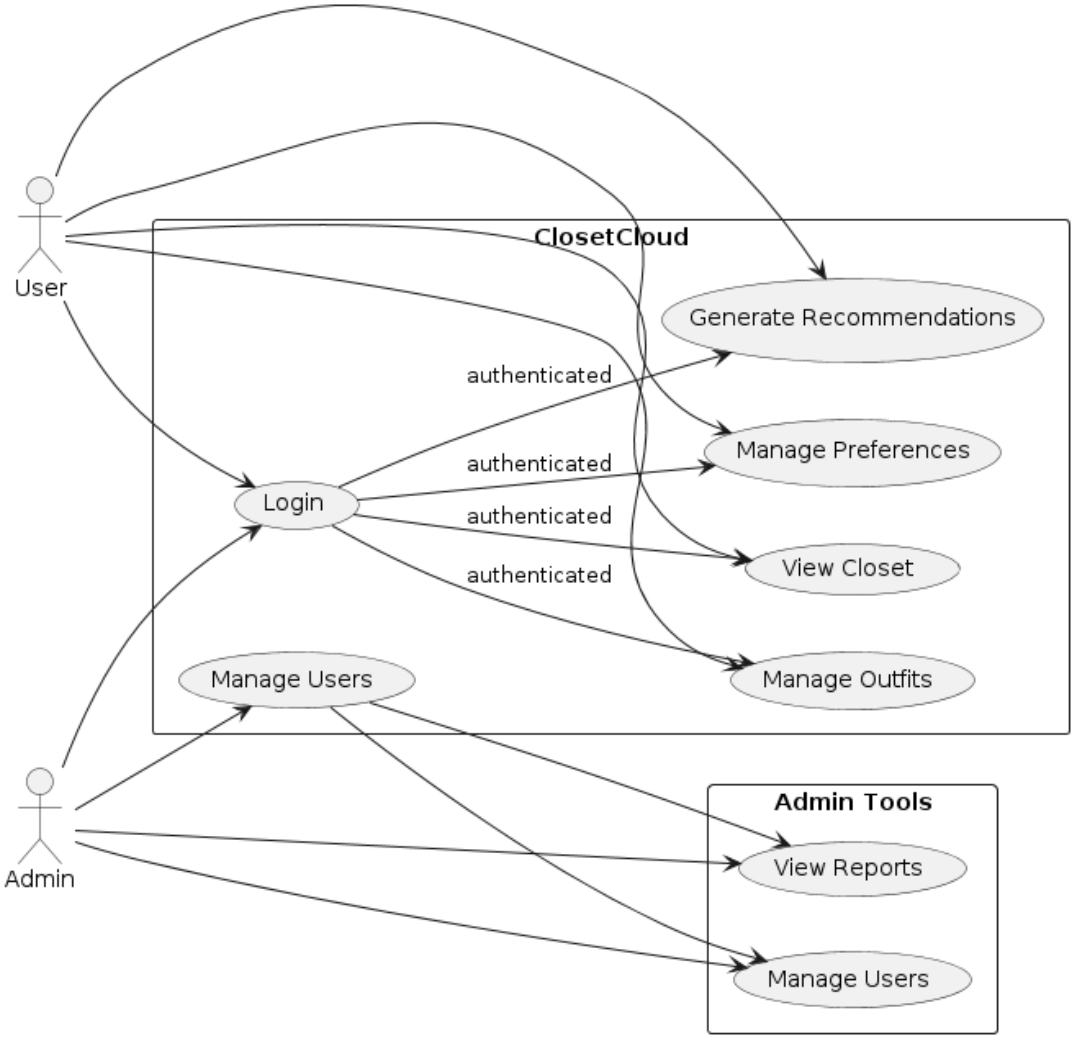
Chapter 3

Use Case Analysis

Chapter 3: System Analysis

This chapter covers the process of identifying actors (users or external systems), defining their goals, and specifying the sequence of steps required to accomplish those goals. It also emphasizes the importance of creating clear and concise use case diagrams and narratives to ensure effective communication between stakeholders, developers, and testers. Use case analysis serves as a foundation for designing and developing software systems that align with user needs and expectations, ultimately leading to successful and user-friendly applications.

3.1. Use Case Model



4 UseCase Diagram

3.2. Use Cases Description

Table 3 Use Case Login

Use Case Name	Login
Primary Actors	User, Admin
Goal in Context	Authenticate the user and provide access to the system.
Precondition	User has the application installed. User is not logged in.
Postcondition	User is logged into the system.
Triggers	User launches the application.
Scenario	1. User launches the app. 2. App prompts for credentials. 3. User enters valid credentials. 4. App authenticates and grants access.

Table 4 Manage Outfits

Use Case Name	Manage Outfits
Primary Actors	User
Goal in Context	To Manage outfits in the closets
Precondition	User must be logged in
Postcondition	Outfits are successfully managed
Triggers	User selects “Manage Outfits”
Scenario	User navigates to “Manage Outfits”

Table 5 Manage Preferences

Use Case Name	Manage Preferences
Primary Actors	User
Goal in Context	To manage the preferences
Precondition	User must be logged in
Postcondition	Preferences are successfully managed
Triggers	User selects “Manage Preferences”
Scenario	User navigates to “Manage Preferences”

Table 6 View Closets

Use Case Name	View Closets
Primary Actors	User
Goal in Context	To view the content of closet
Precondition	User must be logged in
Postcondition	Closet contents are displayed
Triggers	User selects “View Closets”
Scenario	User navigates to “View Closets”

Table 7 Generate Recommendations

Use Case Name	Generate Recommendations
Primary Actors	User
Goal in Context	To generate outfits recommendations
Precondition	User must be logged in
Postcondition	Recommendations are generated
Triggers	User selects “Generate Recommendations”
Scenario	User navigates to “Generate Recommendations”

Use Case Name	Manage User
Primary Actors	Admin
Goal in Context	To manage users
Precondition	Admin must be logged in
Postcondition	Users are successfully managed
Triggers	Admins selects “Manage Users”
Scenario	Admin navigates to “Manage Users”

Use Case Name	View Reports
Primary Actors	Admin
Goal in Context	To view reports
Precondition	Admin must be logged in
Postcondition	Reports are successfully viewed
Triggers	Admins selects “View Reports”
Scenario	Admin navigates to “View Reports”

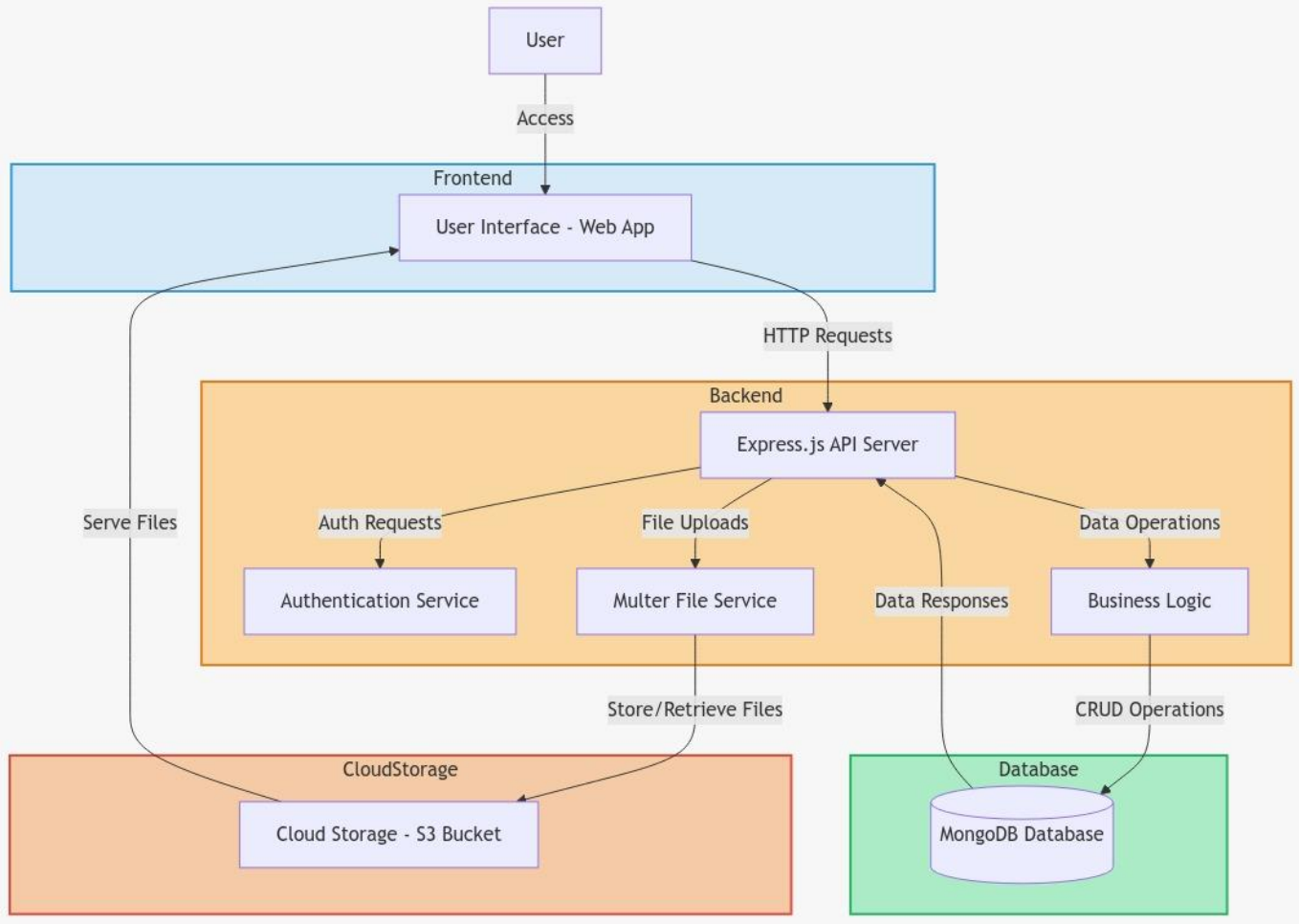
Chapter 4

System Design

Chapter 4: System Design

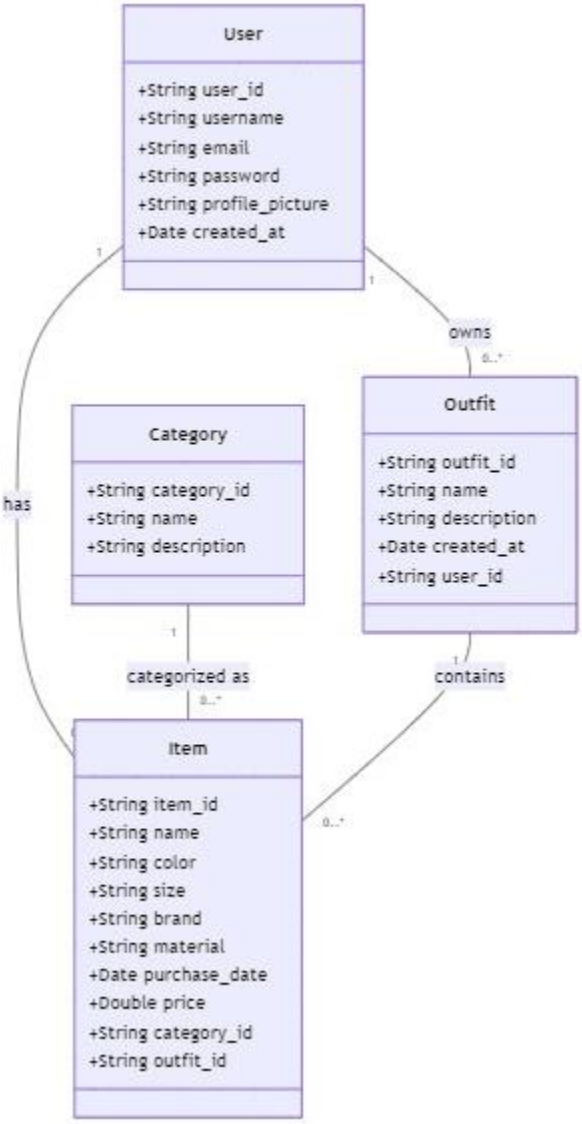
The System Design chapter provides a comprehensive overview of the architectural design, user interface design, and database design for the Closet Cloud project. It outlines the technical specifications and structural components that will facilitate the implementation of the application. This chapter delves into the organization of system modules, user interaction flows, and data storage mechanisms. By detailing the design considerations and decisions, it aims to provide a clear understanding of how the system will be structured and how different components will interact to deliver the desired functionality.

4.1. Architecture Diagram



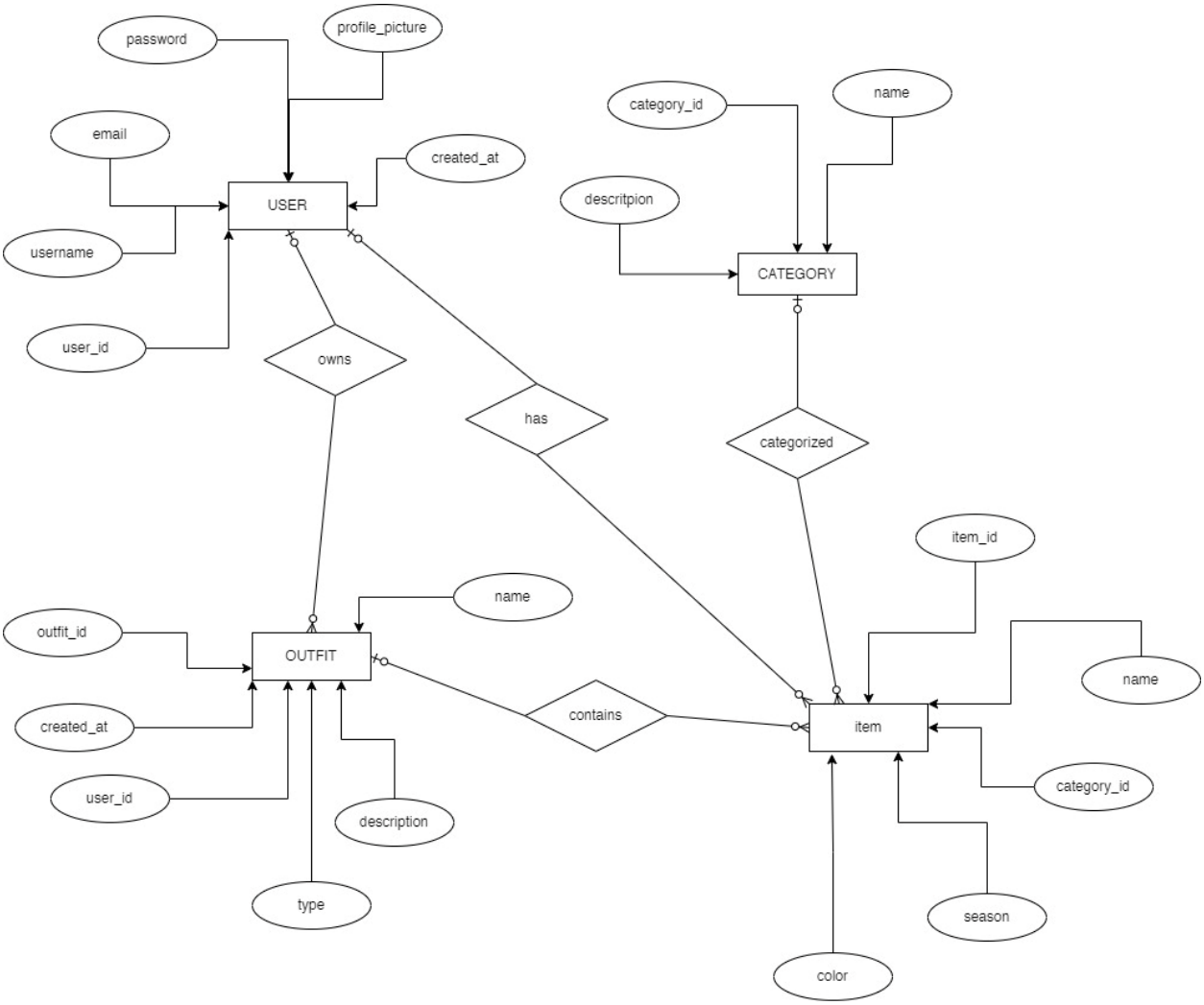
5 Architecture Diagram

4.2. Domain Model



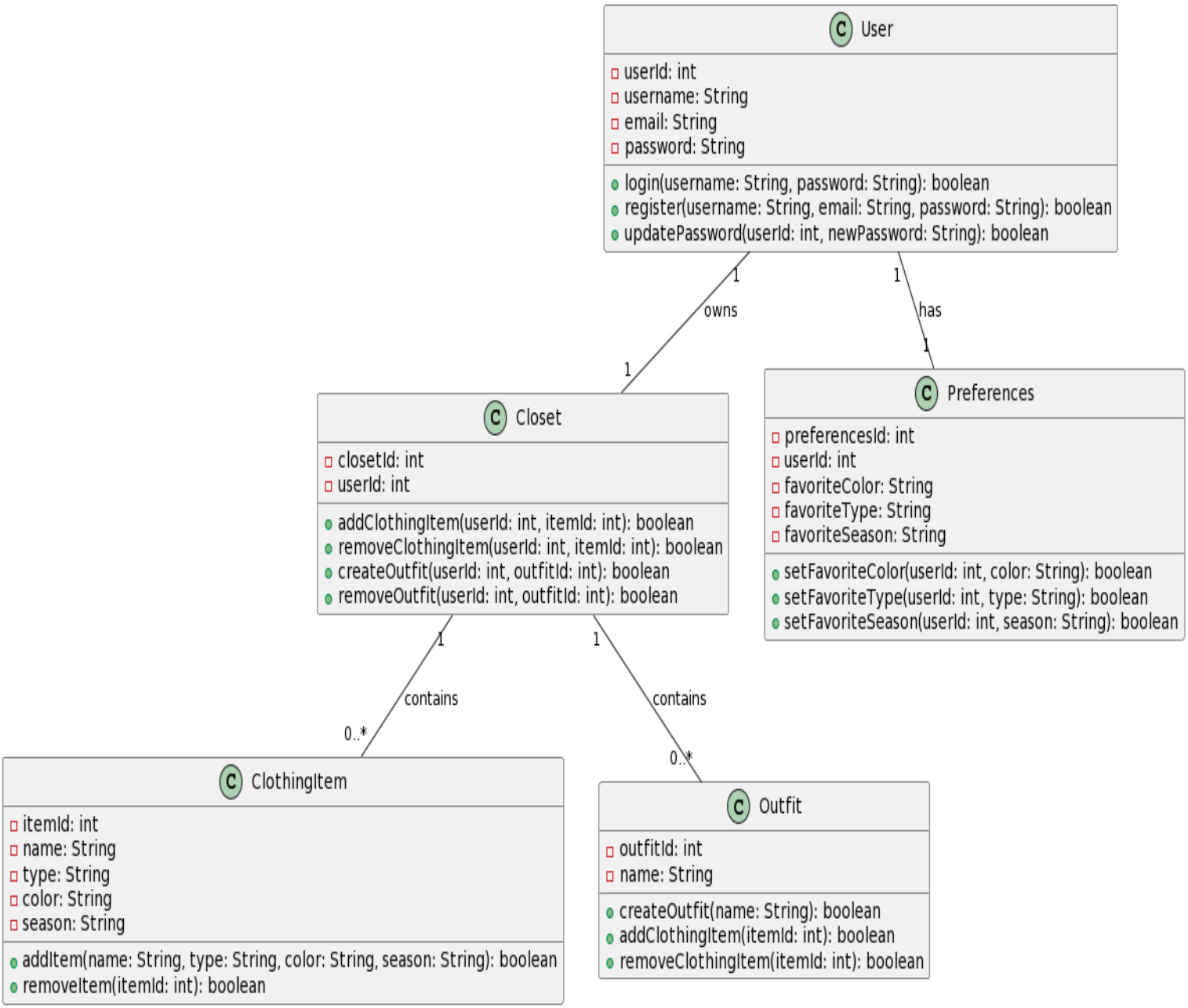
6 Domain Model

4.3. Entity Relationship Diagram with data dictionary



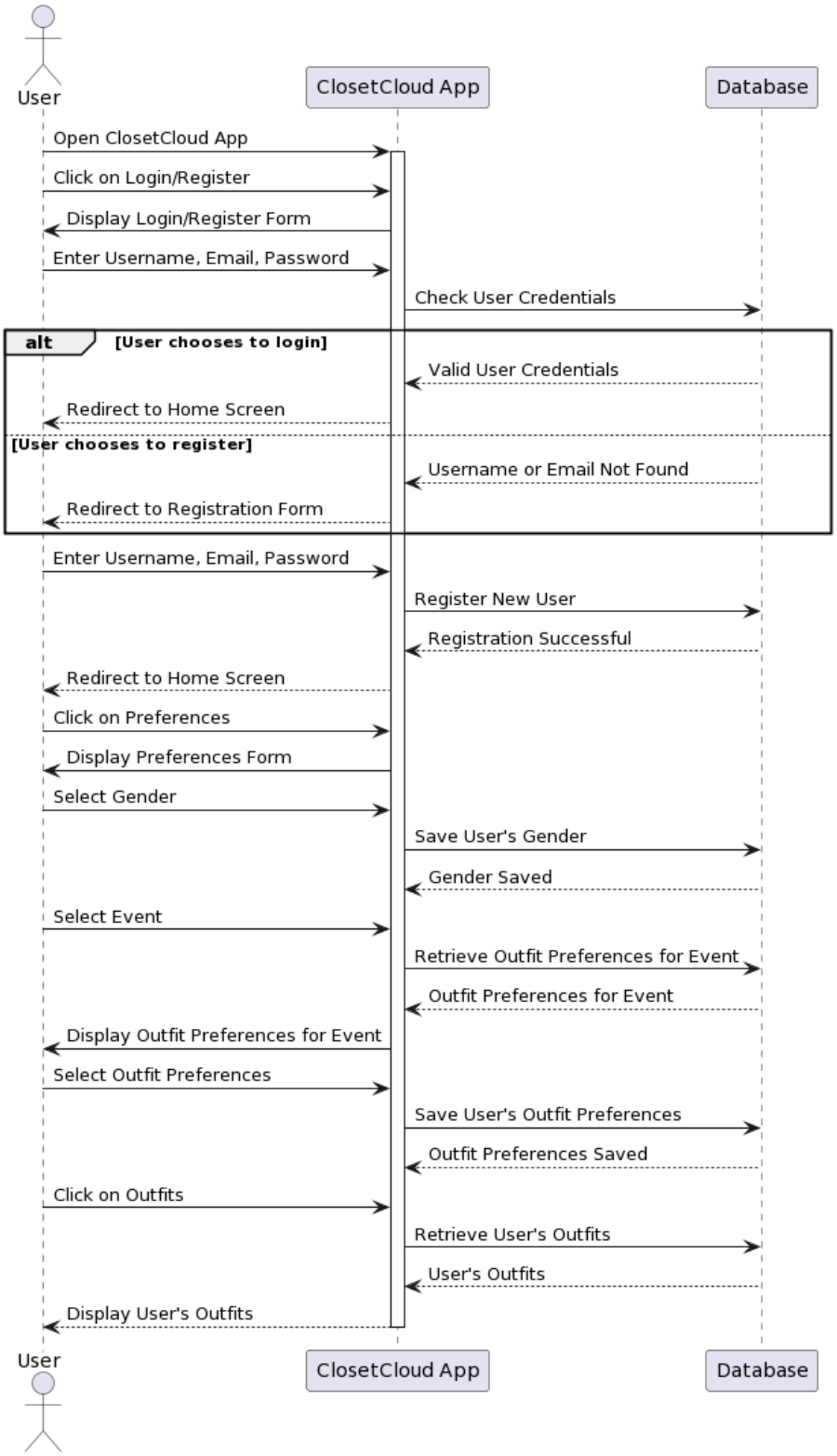
7 ERD Diagram

4.4. Class Diagram



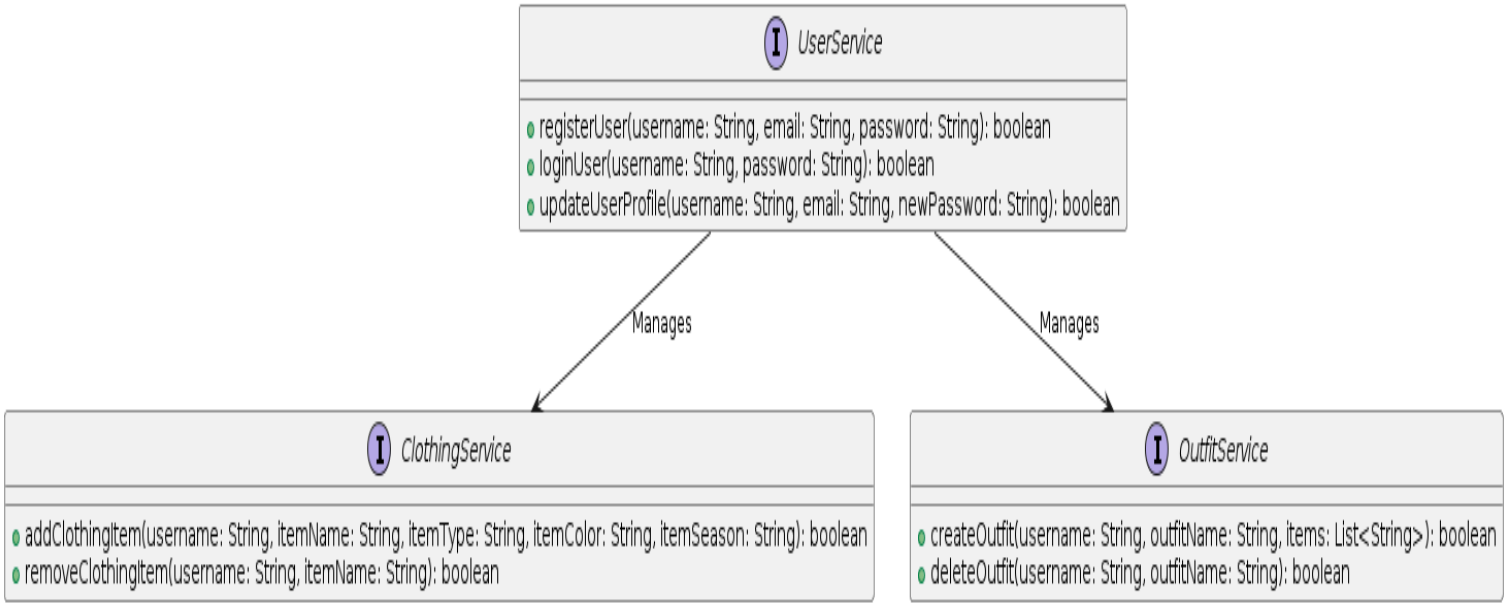
8 Class Diagram

4.5. Sequence / Collaboration Diagram



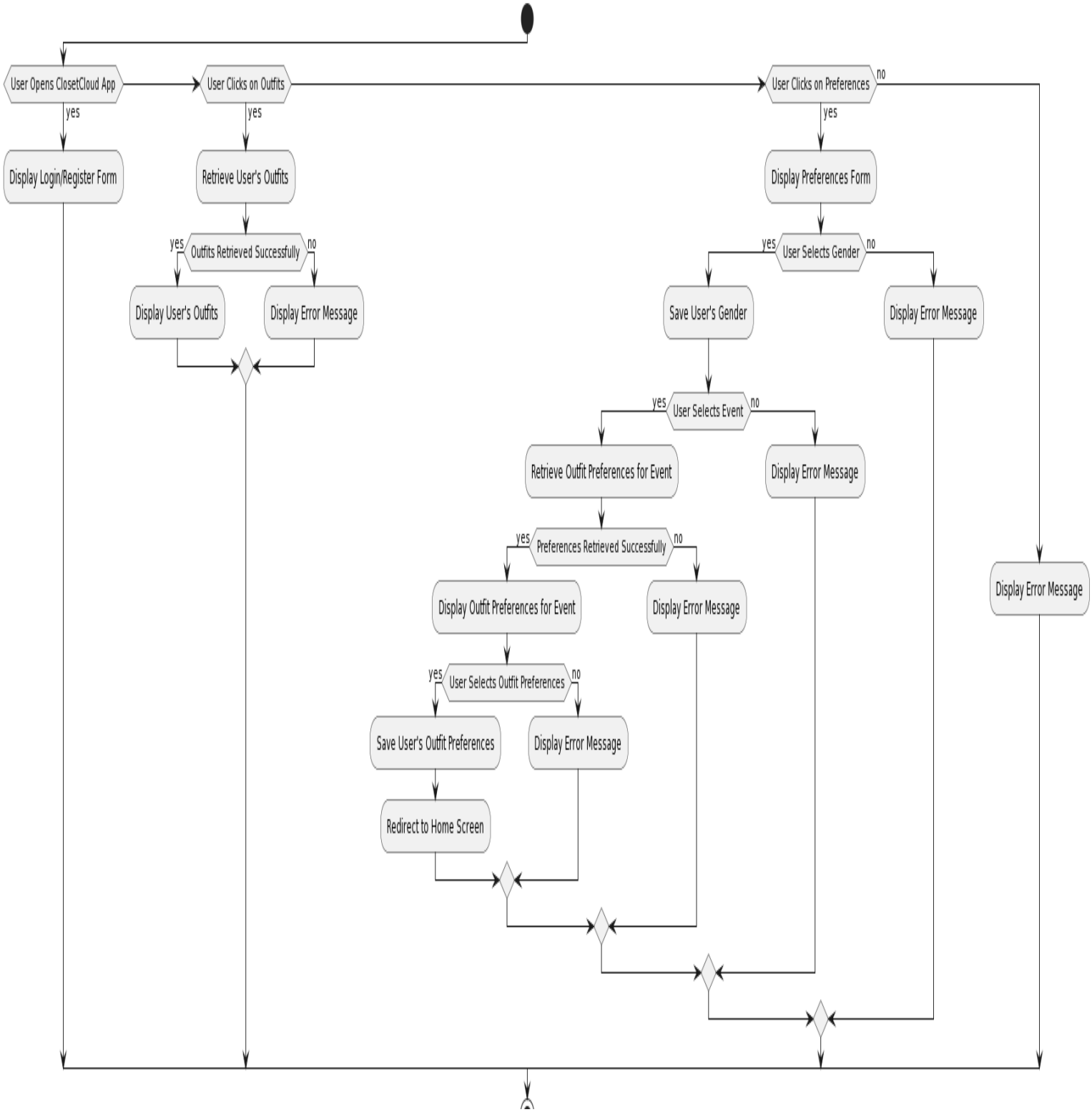
9 Sequence / Collaboration Diagram

4.6. Operation contracts



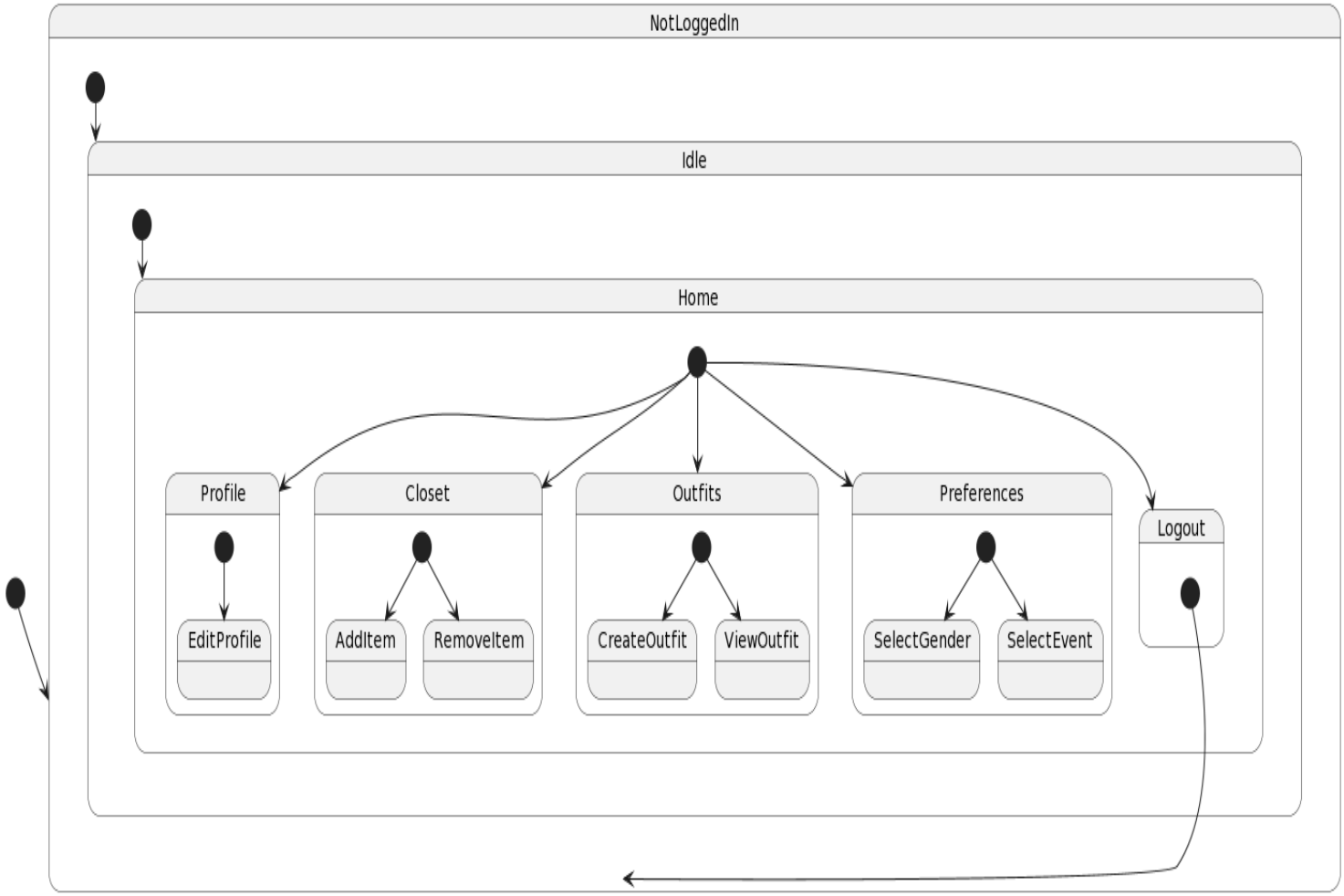
10 Operation Contracts

4.7. Activity Diagram



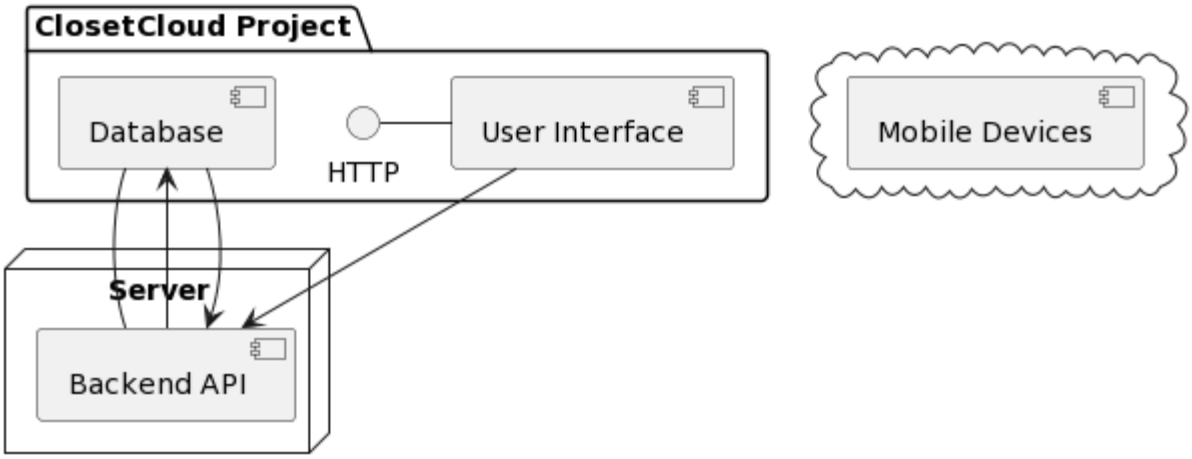
11 Activity Diagram

4.8. State Transition Diagram



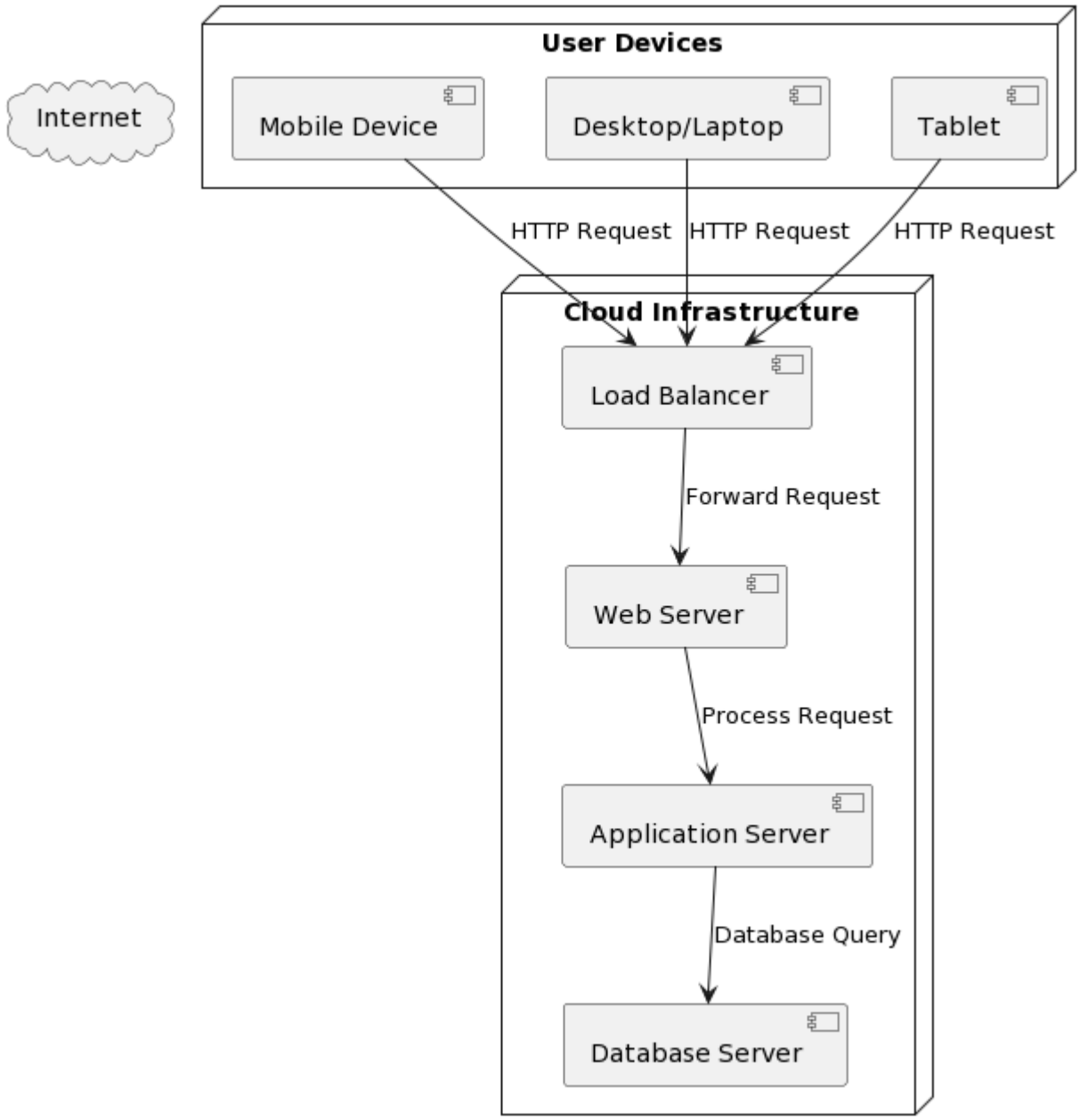
12 State Transition Diagram

4.9. Component Diagram



13 Component Diagram

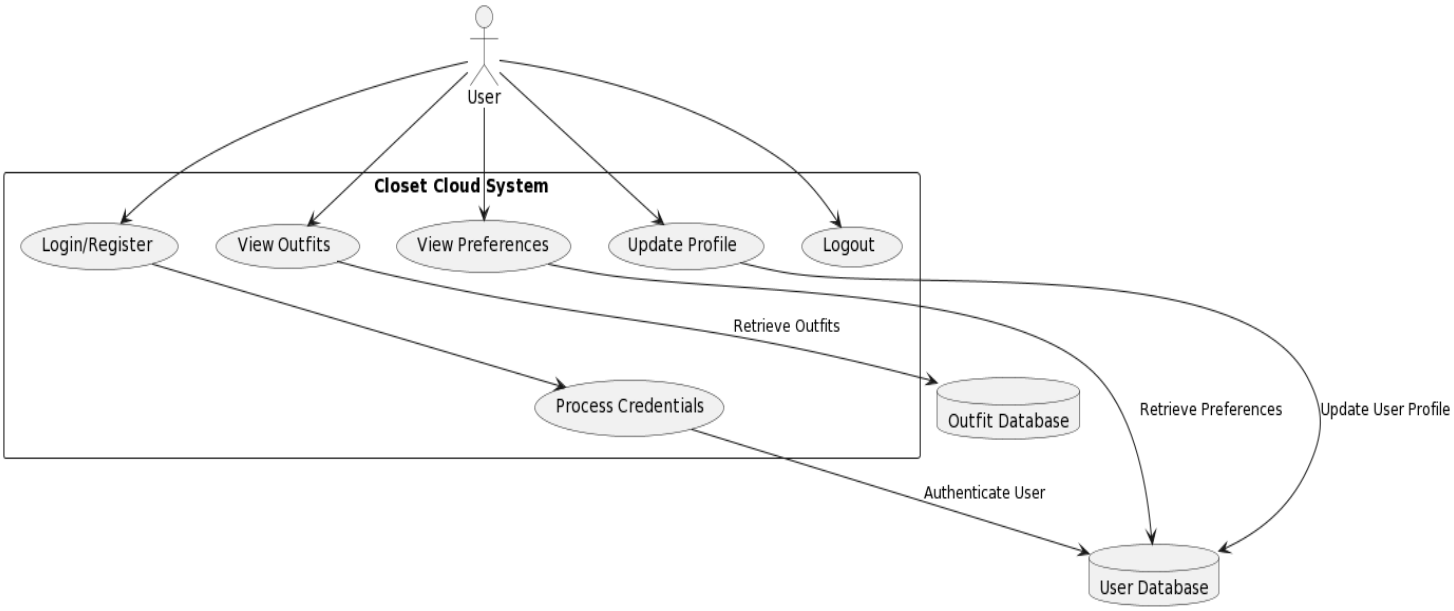
4.10. Deployment Diagram



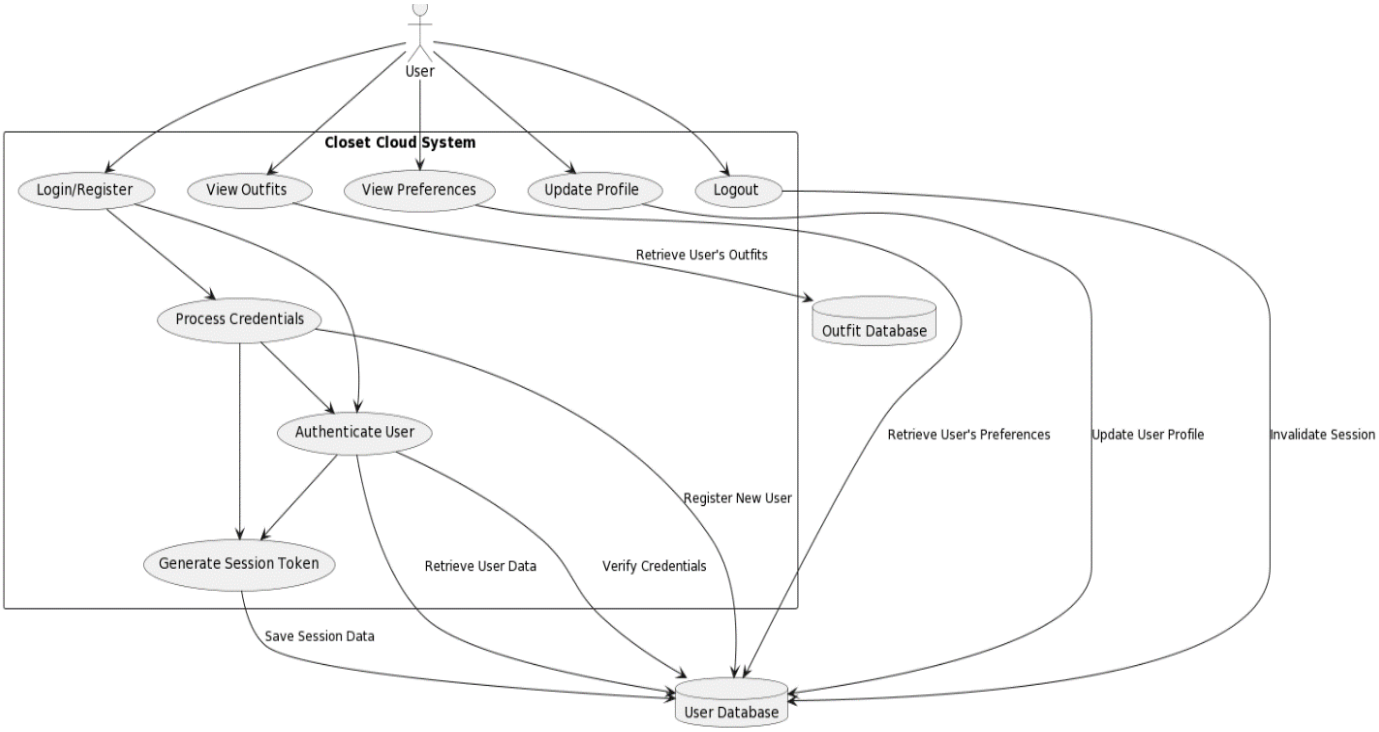
14 Deployment Diagram

4.11. Data Flow diagram [*only if structured approach is used - Level 0 and 1*]

Level 0



16 Data Flow Diagram (level-0)



15 Data Flow Diagram (level-1)

Chapter 5
Implementation

Chapter 5: Implementation

The implementation chapter of a project documentation serves as a detailed guide on how to turn the conceptual design into a functional system. It outlines the steps involved in coding, configuring, and deploying the software, providing developers with a roadmap for bringing the project to life. This chapter typically covers topics such as programming languages, frameworks, libraries, and tools used in development, as well as setup instructions, coding standards, version control practices, testing methodologies, and deployment strategies. It serves as a comprehensive reference for developers, ensuring consistency and efficiency throughout the implementation process.

5.1. Important Flow Control/Pseudo codes

In the implementation of the Closet Cloud project, several important flow control structures and pseudo-codes are essential to ensure the smooth execution of various functionalities. These include:

- **User Authentication:**

- Pseudo-code:

```
function authenticateUser(username, password) {
if (isValidCredentials(username, password)) {
   return true;
} else {
   return false;
}
}
```

- **Adding a Clothing Item:**

- Pseudo-code:

```
function addClothingItem(itemDetails) {
   if (isValidItemDetails(itemDetails)) {
      saveItemToDatabase(itemDetails);
      return true;
```

```
    } else {
        return false;
    }
}
```

- **Creating an Outfit:**

- Pseudo-code:

```
function createOutfit(outfitDetails) {
    if (isValidOutfitDetails(outfitDetails)) {
        assembleOutfit(outfitDetails);
        saveOutfitToDatabase(outfitDetails);
        return true;
    } else {
        return false;
    }
}
```

- **Viewing Preferences:**

- Pseudo-code:

```
function viewPreferences(userId) {
    preferences = getPreferencesFromDatabase(userId);
    if (preferences) {
        displayPreferences(preferences);
    } else {
        displayErrorMessage("Preferences not found.");
    }
}
```

- **Logging Out:**

- Pseudo-code:

```
function logout() {
    clearSessionData();
```

```
    redirectToLoginPage();
}
```

5.2. Components, Libraries, Web Services and stubs

In the implementation of the Closet Cloud project, various components, libraries, web services, and stubs play crucial roles in enhancing functionality and ensuring seamless integration. Here's an overview:

- **Components:**
  - **User Interface Components:** Frontend components built using frameworks like React.js to create interactive and responsive user interfaces.
  - **Backend Components:** Backend components implemented using technologies such as Node.js, Express.js, or MongoDB to handle business logic, data processing, and database interactions.
  - **Image Processing Component:** A component responsible for processing images uploaded by users, including resizing, cropping, and optimizing for storage and display purposes.
  - **Machine Learning Algorithm:** Supervised learning algorithms are used for implementing AI-driven solutions, such as outfit recommendations and image recognition.
- **Libraries:**
  - **Database Libraries:** Libraries such as Sequelize for interacting with databases like MongoDB to perform CRUD operations and manage data efficiently.
  - **Authentication Libraries:** Libraries like Passport.js for implementing user authentication and session management, while **Multer** is used for handling file uploads in Node.js applications.
- **Web Services:**
  - **Third-party APIs:** Integration with external APIs for functionalities such as data processing or machine learning tasks (e.g., using Supervised Learning).
- **Stubs:**
  - **Mock APIs:** Stubs or mock APIs created using tools like JSON Placeholder to simulate backend endpoints during frontend development and testing.
  - **Testing Stubs:** Mock implementations of external dependencies or services used during unit testing or integration testing to isolate components and ensure test reliability.

5.3. Deployment Environment

The deployment environment for the Closet Cloud project encompasses various elements to ensure the seamless deployment and operation of the application. Here's an overview of the key components:

- **Cloud Infrastructure:**
  - Utilization of cloud service providers such as Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform (GCP) for hosting the application's backend services, databases, and storage solutions.
  - Deployment of virtual servers or containers using services like Amazon EC2, Azure Virtual Machines, or Google Kubernetes Engine (GKE) to manage and scale the application's compute resources dynamically.
- **Database Hosting:**
  - Deployment of NoSQL databases like MongoDB for storing unstructured data, providing flexibility and scalability for data storage requirements.
- **Security Measures:**
  - Implementation of security best practices such as network segmentation, encryption at rest and in transit, and least privilege access controls to protect sensitive data and prevent unauthorized access.
  - Utilization of Web Application Firewalls (WAFs), DDoS protection services, and vulnerability scanning tools to enhance the application's resilience against cyber threats.
- **Scalability and High Availability:**
  - Configuration of auto-scaling policies to automatically adjust compute resources based on application demand, ensuring optimal performance during peak usage periods.
  - Deployment of load balancers and redundant infrastructure across multiple availability zones to achieve high availability and fault tolerance for the application.

5.4. Tools and Techniques

- **Integrated Development Environments (IDEs):**
  - Utilize IDEs such as Visual Studio Code, IntelliJ IDEA, or Eclipse for writing, debugging, and testing code efficiently.

- Leverage features like syntax highlighting, code completion, and integrated version control to streamline development workflows.

➢ **Version Control Systems (VCS):**

- Employ Git as a distributed version control system to track changes to project files, collaborate with team members, and manage codebase versions effectively.
- Utilize platforms like GitHub, GitLab, or Bitbucket for hosting repositories, managing pull requests, and facilitating code reviews.

➢ **Collaboration Tools:**

- Use communication and collaboration tools like Slack, Microsoft Teams, or Discord for real-time messaging, team discussions, and project coordination.
- Integrate project management platforms such as Jira, Trello, or Asana for task tracking, sprint planning, and progress monitoring.

5.5. Best Practices / Coding Standards

a) Consistent Naming Conventions:

- Use descriptive and meaningful names for variables, functions, classes, and files to enhance code readability.
- Follow naming conventions such as camelCase or snake_case consistently throughout the codebase.

b) Modular and Scalable Design:

- Encapsulate related functionality into modular components and classes to promote code reusability and maintainability.
- Design software architecture with scalability in mind, allowing the application to accommodate future growth and changes.

c) Clear and Concise Documentation:

- Document code using comments, docstrings, or README files to explain its purpose, functionality, and usage.
- Include information about parameters, return values, and any side effects to guide developers who interact with the code.
-

d) Version Control Best Practices:

- Use version control systems effectively to track changes, collaborate with team members, and manage codebase versions.
- Follow best practices such as committing small, atomic changes, writing informative commit messages, and branching for feature development or bug fixes.

e) Error Handling and Exception Management:

- Implement robust error handling mechanisms to gracefully handle exceptions, errors, and edge cases.
- Use try-catch blocks or error-handling middleware to handle exceptions effectively and prevent application crashes or unexpected behavior.

5.6. Version Control

Version control is crucial for managing the development of the Closet Cloud project effectively. Here are some key aspects of version control for this project:

1. **Git as Version Control System:**
   - Utilize Git as the primary version control system due to its popularity, robustness, and extensive feature set.
   - Establish a centralized Git repository to host the project's source code, allowing team members to collaborate efficiently.
2. **Code Reviews:**
   - Incorporate code reviews as an integral part of the development process to ensure code quality and maintain consistency.
   - Conduct peer reviews for proposed changes before merging them into the main branch, providing feedback and identifying potential issues.
3. **Continuous Integration (CI):**
   - Integrate version control with a CI/CD pipeline to automate build, test, and deployment processes.
   - Configure CI workflows to trigger on each commit or pull request, running automated tests and code quality checks to validate changes.

4. Documentation Management:

- Leverage version control for managing project documentation, including requirements, design documents, and user manuals.
- Store documentation alongside code in the repository, ensuring consistency and alignment between code changes and corresponding documentation updates.

Chapter 6

Testing and Evaluation

Chapter 6: Testing and Evaluation

This chapter discusses the **testing and evaluation** processes conducted to ensure the Cloud Closet system is reliable, efficient, and user-friendly. Various testing techniques were applied to validate different aspects of the system, such as **functionality, performance, and usability**. Each section highlights specific testing methods, including **use case testing, boundary value analysis, and stress testing**, to guarantee that the system meets user expectations and performs optimally under different scenarios. The chapter provides insights into the challenges encountered during testing, the methods used to address them, and recommendations for further improvement.

6.1. Use Case Testing

Use case testing evaluates the core functionalities of the CLOSET CLOUD to confirm that the platform work as intended.

6.1.1 Login Page Testing

Table Login Page Testing

Table 8 Login Page Testing

Test Case ID	TC-001
Test Case Summary	When admin click on login button it must be login and show the main page.
Prerequisites	User must be on login page and have username and password.
Test Procedure	Verify the user email and password.
Actual Result	User successful login.
Status	Successful.
Test Steps	• Navigate the login page • Enter the username • Enter the password • And click on login button • Logout

Table 6.1.1

6.1.2 Sign Up Testing

Table Sign Up Page Testing

Table 9 SignUp Page Testing

Test Case ID	TC-002
Test Case Summary	User must be signup after filling form.
Prerequisites	User must be on signup page.
Test Procedure	Verify the inputs given in form.
Expected Result	User must be allotted code and username, password and SignBot Dashboard.
Actual Result	User successful signup.
Status	Successful.
Test Steps	• Navigate the login page • Fill up the form. • And click on login button

Table 6.1.2

6.1.3 Extension Errors

Table Extension Error

Table 10 Extension Errors

No.	Step	Description
1-a	Login Failed	Login failed if user enter wrong username. System will display an error message.
1-b	Login Failed	Login failed if user enter wrong password. System will display an error message.
2	Registration Failed	Registration will have failed if a new user doesn't follow the rules. System will display an error message.

Table 6.1.3

6.2. Equivalence partitioning

This method divides input data into **equivalence classes** to identify errors efficiently.

- **Input Variables:**
    - Clothing categories (shirts, pants, accessories)
    - Image formats (JPEG, PNG, etc.)
    - Device types (mobile, tablet, desktop)
    - User roles (admin, guest, registered user)
    - Internet conditions (Wi-Fi, 4G, 3G)
- **Equivalence Classes:**
    - **Clothing Categories:**
        - Class 1: Outerwear (jackets, coats)
        - Class 2: Casual wear (T-shirts, jeans)
    - **Device Types:**
        - Class 3: Mobile devices
        - Class 4: Desktop browsers

This ensures the system handles various inputs correctly across all equivalence classes.

6.3. Boundary value analysis

Boundary value analysis focuses on testing the **limits of input values** to catch edge cases. For example:

- **Item Upload Feature:**
    - **Minimum Image Size:** 1 MB (valid)
    - **Maximum Image Size:** 5 MB (valid)
    - **Boundary Test:** Image size of 0.9 MB (invalid) and 5.1 MB (invalid)
- **Closet Capacity:**
    - **Min Limit:** 1 item
    - **Max Limit:** 1000 items
    - **Boundary Test:** Attempt to upload 0 items (invalid) and 1001 items (invalid)

6.4. Data flow testing

Data flow testing examines how **data moves through the system** and identifies any anomalies. For Cloud Closet, the focus was on how clothing data is uploaded, stored, and retrieved:

1. **Upload Process:** Image and metadata are uploaded to the server.
2. **Storage Verification:** Data is stored in the database without loss.
3. **Retrieval Process:** The system retrieves items accurately for the outfit creation feature.

6.5. Unit testing

Table 11 Unit Testing

Test Case ID	Test Details	Input	Expected Result	Status
TC-001	Upload Item to Closet	Image File	Item Uploaded Successfully	Passed
TC-002	Generate Outfit Recommendation	Clothing Items	Suggested outfit displayed	Passed
TC-003	User Login Authentication	Credentials	User Authenticated	Passed
TC-004	Delete Item from Closet	Item Id	Item Deleted Successfully	Passed

6.6. Integration testing

Table 12 Integration Testing

Test Case ID	IT-001
Test Summary	Verify integration between the outfit creation.
Test Procedure	Create an outfit.
Expected Result	Outfit is created.
Status	Passed

6.7. Performance testing

Table 13 Performance Testing

Test Case ID	PT-001
Test Type	Response Time Testing
Required Performance	Less than 2 second
Actual Performance	1.5 seconds
Status	Passed

6.8. Stress Testing

Stress testing assesses how well the system performs under **extreme load** conditions.

- **Scenario:** Upload 500 clothing items simultaneously.
- **Outcome:** The system remained stable, with a minor delay in response time (5 seconds).
- **Result:** The system successfully handled the load, ensuring **data integrity** and functionality.

Additional stress testing included:

- **Multiple Users:** 50 users creating outfits at the same time.
- **Network Load:** Testing on low-bandwidth networks to assess system behavior.

Chapter 7

Summary, Conclusion and Future Enhancements

Chapter 7: Summary, Conclusion & Future Enhancements

7.1. Project Summary

The Cloud Closet project is an innovative concept that occupies different areas of a person's wardrobe and styling process to achieve successful client image transformation. It presents a professional and user-oriented approach that enables users to control an outfit, organize their clothes and learn more about styles. The functionalities of Cloud Closet, outfit building, closet organization, and user login, as well as item uploading that closes the gap between the traditional approach to handling wardrobe and bringing it into the digital era.

This platform is not a simple space of 'digital closet utility', but one that is activated by extant AI algorithms. There are features below that allow the users to see a plethora of styling options and plan their outfits: The user authentication system allows for the creation of a safe shopping environment, and virtual closet extension is easy due to the recognizable item upload option.

Cloud Closet is a complete assistant covering all aspects of fashion for your wardrobe. It fits small clerking chores such as reminding one what to wear in the next day and large weekly chores such as organizing seasonal exchanges of clothes. Translating to a user-friendly application interface used with advanced technologies today while being designed to accommodate future trends in fashion and technology, the project has the flexibility of a growth model.

7.2. Achievements and Improvements

Throughout the project, several key achievements were realized:

- **Implementation of Core Functionalities:** We managed to incorporate most of the components including item upload, closet management and outfit planning.
- **AI-based Recommendations:** The AI algorithms enabled the system to suggest a set of outfits and combinations which helped the users to gain ideas about what they might wear.
- **User Authentication and Security:** An effective means of user's identification guarantees data of the Internet resource and wardrobe information from unauthorized use.
- **Scalable Infrastructure:** Overall, the project architecture caters for future enhancement work: extending with third-party fashion services, shopping or a stylist check-out buttons.

Changes introduced during development were, for instance, enhancing the UI so users have a good experience to go with the application. Furthermore, the system's latency was improved to make it equally usable across mobile click based and mouse based devices. Ideas and reviews received were timely in helping to develop further features since Cloud Closet was to address real user requirements efficiently.

7.3. Critical Review

Despite the successful implementation of major features, the project faced a few challenges:

- **AI Recommendation Limitations**: In fact, the use of AI in outfit proposing is effective, however, it relies on the item tags or categories from the users in terms of quality and accuracy. It is conceivable subsequent iterations may need more complex mathematical algorithms, as well as application of machine learning models for optimal outcome.
- **Feature Complexity:** A few of the users complained of complexity in creating their outfits. This is going to be critical in the next steps of the evolution process because reducing the complexity of the interaction while not limiting options will become a key area of focus for growth.
- **Limited Fashion Integration:** What is more, the current platform provides wardrobe planning that is vitally important, but it is even more beneficial to be based on the integration with the hot e-shopping platforms or the fashion brands.

These challenges point out areas that, if refined to a greater degree, can add a lot of value and magnetism to the platform.

7.4. Lessons Learnt

The development of Cloud Closet provided several valuable insights and lessons:

- **User-Centric Design:** Listening to early users is important because when designing features, you must address real user issues. The enhancements from the iteration based on the condition result of user needs and behavior enhanced the platform usability.
- **Balancing Functionality with Simplicity:** Creating a feature-rich platform as the Cloud Closet implies that there is always a fine line between offering utility and making it intuitive. Having too many features leads to the fact that interfaces are overloaded and excessive functionality

is not actively used, so software development should take into account the identified core needs and provide an opportunity to perform additional optional tasks.

- **AI Adaptability:** Artificial Intelligence really depends on the inputs it receives. We also found that recommendation systems with neatly defined categories in aggregated data drastically enhance the performance.
- **Project Management and Collaboration:** Another factor that influenced the project success was communication and application of the use of the agile models. Integration of the various work stages was facilitated through effective teamwork whereby design and development process as well as the testing processes were well coordinated.
- **Scalability Considerations:** Future scalability considerations were viewed as paramount when building the platform. As much as the services and technology advance the fashion industry, the capacities for adding new services or partners will be the key to Cloud Closet's sustainability.

7.5. Future Enhancements/Recommendations

To further enhance Cloud Closet and address identified challenges, the following future improvements are recommended:

- **Enhanced AI and Machine Learning Models:** By incorporating such higher level AI architectures it is possible to integrate personalized aspects to the application that include weather advice on what to wear, appropriate clothing for a certain occasion and so on.
- **Third-Party Integrations:** Fashion retailer or e-commerce integration would provide missing wardrobe items at direct shopping convenience, more users are familiar with Amazon, Zalando, etc.
- **Virtual Try-On and AR Features:** Adding the aspect of AR as a feature to try on outfits with, would contribute significantly to a better general use of the application as people will be able to see how an outfit will look on them before they wear the clothes.
- **Mobile-First Development:** As it is evident, the platform is responsive, but a dedicated mobile application would be more convenient to use especially for anyone on the go.

- **Sustainable Fashion Integration:** For sustainable fashion, the platform can modify the platform added to track the use of the wardrobe, or popping up the suggestions for buying from the thrift store.
- **Community Engagement:** Installing chatter boxes or, specially constructed style sharing boxes in the app would enable users to share styling advice or tips leading to increased fashion engagement.

These improvements will also make Cloud Closet even grow to be more effective, providing the users with the effective, fun and progressive interface to help them keep track of their persona image. Given the current trends in which fashion is going more digital, Cloud Closet is sure to make a perfect fit as a tool that everybody who needs a platform with which to express their style seamlessly and with ease would opt for.

Reference and Bibliography

Reference and Bibliography

I. Al-Omar, N. N., Al-Rashed, N. M., Al-Fantoukh, H. I., al-Osaimi, R. M., Al-Dayel, A. H. A., & Mostefai, S. (2013). The design and development of a web-based virtual closet: The smart closet project. *Journal of Advanced Management Science*, *1*(1), 124-128.

II. Li, Y., & Hu, W. (2014). The Conceptual Design of" Smart Closet" Fashion Consultant Expert System. *International Journal of Business and Social Science*, *5*(10).

III. Drushel, B. E. (2012). Virtual Closets: Strategic Identity. *Social networking and impression management: self-presentation in the digital age*, 149.

IV. Fukuda, M., & Nakatani, Y. (2011, October). Clothes recommend themselves: A new approach to a fashion coordinate support system. In *Proceedings of the World Congress on Engineering and Computer Science* (Vol. 1, pp. 19-21).

V. Perry, A. (2016). Consumers' acceptance of smart virtual closets. *Journal of Retailing and Consumer Services*, *33*, 171-177.

VI. 안지윤. (2020). *UX Design for an Effective Closet Browsing Experience through Smart Closet* (Doctoral dissertation, 서울대학교 대학원).

VII. Huang, W. (2021). *Closet Go: A Data-Driven Digital Closet System to Improve the Dressing Experience* (Doctoral dissertation, Purdue University).

VIII. Fernandes, C. E., & Morais, R. (2021). A review on potential technological advances for fashion retail: smart fitting rooms, augmented and virtual realities. *dObra [s]–revista da Associação Brasileira de Estudos de Pesquisas em Moda*, (32), 168-186.

IX. Perry, A. (2017). Factors that influence consumers' purchase intention of smart closets. *International Journal of Fashion Design, Technology and Education*, *10*(1), 91-100.

X. Rode, J. A., Magee, R., Sebastian, M., Black, A., Yudell, R., Gibran, A., ... & Zimmerman, J. (2012, September). Rethinking the smart closet as an opportunity to enhance the social

currency of clothing. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (pp. 183-192).

XI. Matsumoto, T., Wada, Y., Norimatsu, M., & Okude, N. (2004, January). Smart-media applications created by Scenario based modeling. In *2004 International Symposium on Applications and the Internet Workshops. 2004 Workshops.* (pp. 473-479). IEEE.

XII. Wanick, V., Stallwood, J., & Bazaki, E. (2023). The use of augmented reality to enhance consumer experience: The case of kohl's snapchat virtual closet and sephora virtual artist. In *Reinventing Fashion Retailing: Digitalising, Gamifying, Entrepreneuring* (pp. 55-71). Cham: Springer International Publishing.

XIII. Pillis, D., Pataranutaporn, P., Maes, P., & Sra, M. (2024, March). AI Comes Out of the Closet: Using AI-Generated Virtual Characters to Help Individuals Practice LGBTQIA+ Advocacy. In *Proceedings of the 29th International Conference on Intelligent User Interfaces* (pp. 686-698).

XIV. Li, Y., & Hu, W. (2014). The Conceptual Design of" Smart Closet" Fashion Consultant Expert System. *International Journal of Business and Social Science*, *5*(10).

XV. Case, S. E. (2007). *Performing science and the virtual*. Routledge.

FYP II Report-1.pdf

ORIGINALITY REPORT

16%	11%	7%	13%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Higher Education Commission Pakistan Student Paper	11%
2	www.slideshare.net Internet Source	1%
3	www.coursehero.com Internet Source	1%
4	skemman.is Internet Source	1%
5	research.superior.edu.pk Internet Source	1%
6	Submitted to CSU, Los Angeles Student Paper	1%
7	Submitted to Oulu University of Applied Sciences Student Paper	<1%
8	Submitted to Gulf College Oman Student Paper	<1%
9	fastercapital.com	

	Internet Source	<1%
10	srmiliss2024.srmist.edu.in Internet Source	<1%
11	www.joams.com Internet Source	<1%
12	krisp.ai Internet Source	<1%
13	pdfcoffee.com Internet Source	<1%
14	slashdev.io Internet Source	<1%
15	www.wetterradar.uni-hamburg.de Internet Source	<1%
16	Jimut Bahan Pal. "Wisp: A preference based location finder application", Open Science Framework, 2019 Publication	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		