BASED ON MINIO CLOUD DRIVE APPLICATION DESIGN AND DEVELIOPMENT

By

YI QIAO

A Project Report

Submitted to the Department of Computing & Software And the School of Graduate Studies

of McMaster University

in Partial Fulfilment of the Requirements for the degree

Master of Engineering

McMaster University

© Copyright by Prachi Goyal, December 2024

Master of Engineering (2024) (Computing & Software)

McMaster University Hamilton, Ontario, Canada

**TITLE:** BASED ON MINIO CLOUD DRIVE APPICATION DESIGN AND DEVELOPMENT

**AUTHOR:** Yi Qiao

M.Eng. Student (Computer & Software) McMaster University, Hamilton, Canada

**SUPERVISOR:** Dr. Richard Paige

**NUMBER OF PAGES:** 38

# ABSTRACT

OpenEMR is among the most popular open-source electronic health record and medical practice management solutions. It features fully integrated electronic health records, practice management, scheduling, electronic billing, internationalization, free support, and a vibrant community. It can run on Windows, Linux, Mac OS X, and many other platforms. Mobile application development is a fast-growing industry. As the number of mobiles and smartphones increases day by day, businesses are developing innovative mobile apps to attract their target customers. Using a mobile app to reach customers is ideally suited as it can easily keep the audience engaged and interested when using mobile apps. This report explains the design and development of a cross-platform mobile application for an E-Health patient using the OpenEMR 6.0.0 patient portal.

CONTENTS

[ABSTRACT 3](#_Toc757164666)

[1. INTRODUCTION 5](#_Toc1825932987)

[1.1 Background and Significance of the Study 5](#_Toc914396879)

[1.2 The Need for Research 5](#_Toc875367421)

[1.3 Research Content and Objectives 5](#_Toc2037262797)

[1.4 Research Methodology and Technical Route 6](#_Toc796561411)

[1.5 Organization of the paper 6](#_Toc394579279)

[2. RELATED WORK 8](#_Toc264440217)

[2.1 Overview of Cloud Storage Technology 8](#_Toc1303061476)

[2.2 Existing Cloud Storage Services 8](#_Toc515995026)

[2.3 Open-Source Cloud Storage Solutions 9](#_Toc789435396)

[2.4 MinIO: High Performance, Kubernetes-Native Object Storage 9](#_Toc886729406)

[2.5 Vue.js: An Incremental JavaScript Framework 10](#_Toc1872100109)

[3. REQUIREMENTS 12](#_Toc1603619766)

[4. DESIGN 13](#_Toc1117637312)

[4.1 Overall Design 13](#_Toc90842475)

[4.2 Domain Detail Design 13](#_Toc2076087955)

[4.2.1 User Domain 13](#_Toc495963229)

[5. IMPLEMENTATION 15](#_Toc1269955796)

[6. EVALUATION 16](#_Toc307095839)

[7. CONCLUSION 17](#_Toc956562332)

[A. REFERENCES 19](#_Toc880532482)

# INTRODUCTION

## 1.1 Background and Significance of the Study

In the contemporary era of rapid development of information technology, data, and information have become the blood of social operation. How to store and access these data efficiently and securely has become an important driving force for technological progress. Cloud storage service, as one of the solutions, is rapidly gaining popularity among individual and enterprise users with its unique advantages. In particular, open-source cloud storage solutions, such as Minio, provide users with customized service options to adapt to changing storage needs and challenges. Focusing on the application of open-source cloud storage services, this study explores the design and implementation of a cloud disk application based on Minio, aiming to provide users with a data storage option with high performance, high reliability, and easy management.

## 1.2 The Need for Research

Although there are many cloud disk services on the market today, they are often one-size-fits-all solutions that lack sufficient flexibility to meet the individual needs of specific user groups. For example, enterprise users may need to deploy cloud services in their internal network environment to ensure data privacy and security; research institutions may need customized data analysis tools combined with storage solutions; and individual users may seek more efficient data synchronization and backup functions. Existing cloud drive offerings often fail to provide adequate customization support in these areas.

In addition, many cloud disk services have limitations in terms of data sovereignty, with users' control over their own data restricted by the service provider's policies and technical architecture. Due to the ever-changing laws and regulations and the increasing demand for data sovereignty from enterprises and individuals, autonomous and controllable cloud disk services have become an inevitable trend. Based on these real-world needs, the development of a Minio-based autonomous cloud disk application not only provides customized services but also improves the flexibility and efficiency of data processing while ensuring data sovereignty and security.

## 1.3 Research Content and Objectives

The main goal of this thesis is to develop a web disk application that integrates the functions of disconnected transfer, file encryption, user management, and so on. The research covers the whole process from requirement analysis, system design, and interface implementation to functional testing. The back-end development of the system will be in Java and use the Spring Boot framework to improve development efficiency and simplify the deployment process. The front-end interface will be realized by the Vue.js framework to ensure the responsiveness and interactivity of the user interface. This research will also delve into the implementation of the breakpoint transfer technology and how to effectively manage files and user data in an online disk application to provide a secure data transfer and storage solution.

## 1.4 Research Methodology and Technical Route

Object Storage Service (OSS) is a massive, secure, low-cost, and highly reliable cloud storage service suitable for storing any type of files. Capacity and processing capacity are elastically expandable, and multiple storage types are available for selection, fully optimizing storage costs.AliCloud Object Storage OSS (Object Storage Service) is a massive, secure, low-cost, highly persistent cloud storage service provided by AliCloud. Its data is designed to be no less than 99.999999999999% (12 9s) persistent, and service availability (or business continuity) is no less than 99.995%.

MinIO is an object storage service based on the Apache License v2.0 open-source agreement. It is compatible with Amazon S3 cloud storage service interface, ideal for storing large-capacity unstructured data, such as images, videos, log files, backup data and containers/virtual machine images, etc., and an object file can be any size, from a few kilobytes to a maximum of 5T ranging.MinIO is a very lightweight service that can be easily integrated with other applications, such as NodeJS, Redis, or MySQL. For small and medium-sized enterprises, Minio is a good choice if you don't want to go to the cloud for storage. Minio can be used directly as object storage, but also as a gateway layer for object storage services on the cloud, seamlessly connecting to Amazon S3, and MicroSoft Azure.

In order to realize the research objectives, this paper adopts the method of combining theoretical research and empirical analysis. Firstly, a literature review is conducted to analyze the current state of development of cloud storage technology and netbook applications and determine the entry point of the research. Subsequently, the system functions are determined through requirement analysis, and the system architecture is designed based on the characteristics of Minio. In the implementation phase, this research will follow the agile development principle to iteratively complete the development and integration of each functional module. System testing will cover unit testing, integration testing, and performance testing to ensure the stability and reliability of the application.

## 1.5 Organization of the paper

This paper is organized as follows: chapter 1 introduces the background of the research, the need for the research, the content and objectives, and the research methodology. Chapter 2 overviews the related technologies and theoretical foundations, including cloud storage technologies, features of Minio, and the technology stack used for development. Chapter 3 analyzes the requirement points in detail. Chapter 4 discusses system design in detail, including architecture design, functional planning, interface definition, and data model. Chapter 5 shows the system implementation process, including development environment setup, code writing, functional implementation, and interface design. Chapter 6 conducts system testing, analyzes the test results, and evaluates the system performance. The last chapter summarizes the whole paper and presents an outlook on the future research direction.

# RELATED WORK

## 2.1 Overview of Cloud Storage Technology

Cloud storage technology has evolved dramatically since its inception, driven by the growing demand for data accessibility and disaster recovery options. Initially, the concept of cloud storage was to provide users with remote servers where they could store their data without having to worry about maintenance and physical hardware issues. Over time, these services have evolved to provide not only storage but also processing power, allowing complex applications and services to be fully hosted in the cloud. This evolution can be traced back to the development of virtualization technologies, which abstract the physical hardware to allow multiple virtual machines to run on a single physical server [[1]](#endnote-0)[1]. The scalability of these systems is made possible by distributed architectures, which allow data to be stored in multiple locations, thereby increasing redundancy and reliability. As the amount of data generated by organizations continues to grow exponentially, these cloud storage technologies have become an integral part of data management strategies. The shift from capital expenditure (CAPEX) to operational expenditure (OPEX) models has also been a significant factor in the adoption of cloud storage solutions, allowing businesses to pay only for the storage they use, rather than investing in expensive hardware infrastructures [[2]](#endnote-1)[2].

The fundamental properties of cloud storage, such as on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service, have been detailed by leading researchers in the field and form the basis for modern cloud computing paradigms [3][[3]](#endnote-2)[3]. These features contribute to the widespread adoption of cloud storage technologies as they offer flexibility and scalability that cannot be matched by traditional storage solutions. In addition, the ability to access data anytime anywhere has revolutionized the way businesses operate, enabling new methods of workflow and collaboration. The importance of these technological advances cannot be overemphasized, as they pave the way for the next generation of Internet services and applications.

## 2.2 Existing Cloud Storage Services

The landscape of cloud storage services is dominated by several key players, each offering their unique take on cloud storage. Amazon Web Services (AWS) introduced the concept of cloud storage to the masses with its Simple Storage Service (S3), which remains a benchmark for durability, availability, and scalability in the industry [[4]](#endnote-3)[4].AWS S3 provides an object storage service with an impressive 99.99999999999% durability as well as comprehensive security and compliance features, making it the storage solution of choice for organizations that need a robust storage solution for organizations that need a robust storage solution. Microsoft’s Azure Blob Storage complements its cloud offerings by providing a service that integrates seamlessly with other Azure services, offering options for hot, cool, and archive data storage, catering to various business needs[[5]](#endnote-4)[5]. Google Cloud Storage has made great strides by tightly integrating with its data processing services, especially in the area of data analytics, thus appealing to organizations looking to leverage big data[[6]](#endnote-5)[6].

Despite the power of these services, they are not without challenges. Vendor lock-in is a major issue, as the unique features and APIs offered by each vendor can make it difficult for customers to migrate data to different services. Additionally, the cost implications of data transfer and manipulation can be complex and sometimes unpredictable.Complicating the situation is the evolving nature of data protection laws, which require cloud storage providers to continually adapt their services to meet regulatory requirements. As a result, the balance between innovation, cost and compliance remains an evolving goal for the cloud storage industry.

## 2.3 Open-Source Cloud Storage Solutions

The proliferation of open-source cloud storage solutions has reinvigorated the storage market, providing alternatives that prioritize transparency, customizability, and community-driven development. These solutions cater to a diverse set of needs, from personal cloud storage to enterprise-level deployments. Ceph, for instance, is a unified, distributed storage system designed for excellent performance, reliability, and scalability. It is often used in situations that require highly scalable block, file, and object storage under a single whole-system namespace[[7]](#endnote-6)[7]. On the other hand, extends Kubernetes functionalities, turning distributed storage systems into self-managing, self-scaling, and self-healing storage services, thereby simplifying the deployment and management of storage solutions in cloud-native environments [[8]](#endnote-7)[8].

The open-source model also encourages innovation in cloud storage technology. It enables organizations to deploy and customize their own storage solutions to fit specific use cases, which is not always possible with proprietary services. The community development approach enables rapid iteration and incorporates cutting-edge features such as erasure coding and geo-replication, which improves data durability and availability. In addition, the ability to inspect and modify source code provides an additional layer of security and trust, as any security vulnerabilities are quickly identified and addressed by the community.

## 2.4 MinIO: High Performance, Kubernetes-Native Object Storage

MinIO has become an important player in the cloud-native object storage space, offering high performance and compatibility with the S3 API, which many organizations have found critical to meeting their cloud storage needs. Designed from the ground up to support private cloud and containerized environments, MinIO is well-suited for a wide range of data-intensive applications, from machine learning to big data analytics. Its design philosophy centers on simplicity and performance, with a single-layer architecture that facilitates straightforward scaling and management[[9]](#endnote-8)[9] .

MinIO is an object storage service based on the Apache License v2.0 open-source protocol that can be used for cloud storage solutions to save massive amounts of images, videos, and documents. The server side can work on Windows, Linux, OS X, and FreeBSD due to Golang implementation. Configuration is simple, basically copying the executable program, single line commands can be run up.MinIO is compatible with the Amazon S3 cloud storage service interface, which is ideal for storing large-capacity unstructured data, such as images, videos, log files, backup data, and container/virtual machine images, etc., and an object file can be of any size, ranging from a few kilobytes to a maximum of 5T. Its suitability for high-throughput, low-latency applications has been demonstrated in a variety of industry and academic environments, demonstrating its ability to handle the workloads required by modern applications while maintaining ease of use and deployability[[10]](#endnote-9)[9].

At the core of MinIO's support for distributed deployments and high availability of services and data is MinIO's Codec Correction feature. MinIO implements Codesmithing as a core component to provide data redundancy and availability. Assuming that MinIO divides an object into K data slices, and deletion correction generates M checksum slices based on the K data slices, MinIO needs at least K slices of any type to recover the original object. MinIO requires at least K slices of any type to recover the original object, meaning that M slices can be allowed to fail out of a total of K+M slices[[11]](#endnote-10)[10].

MinIO's approach to security is also worth mentioning. The service provides robust security features, including end-to-end encryption, identity and access management, and the ability to create fine-grained access control policies. These features ensure that MinIO can be deployed in sensitive environments where data security is critical. In addition, MinIO's open-source nature provides a vibrant community of developers and users who contribute to the ongoing development of MinIO and provide support through community forums and documentation[[12]](#endnote-11)[11].

## 2.5 Vue.js: An Incremental JavaScript Framework

Vue.js has emerged as an incremental JavaScript framework for building user interfaces. Unlike other monolithic frameworks, Vue was designed from the ground up to be incrementally adoptable. Its core library focuses only on the view layer, making it easy to integrate with other libraries or existing projects. Vue is also perfectly capable of powering sophisticated Single-Page Applications (SPAs) when used in combination with modern tooling and supporting libraries[[13]](#endnote-12)[12].

The flexibility of Vue.js allows developers to build applications to their liking, which has earned the framework a large following in the developer community. Vue.js has a gentle learning curve compared to more complex frameworks, making it popular with both novice and experienced developers, and its extensive documentation and active community support have further contributed to its widespread adoption. Inspired by Google developer Evan You, Vue.js was created in 2014 and inspired by Angular. Like Angular, it is a JavaScript-based toolkit system that is used to build a dynamic user interface. It is progressive, scalable, and best of all, open-source, so there are lots of third-party instruments to play with.[[14]](#endnote-13)[13].

# REQUIREMENTS

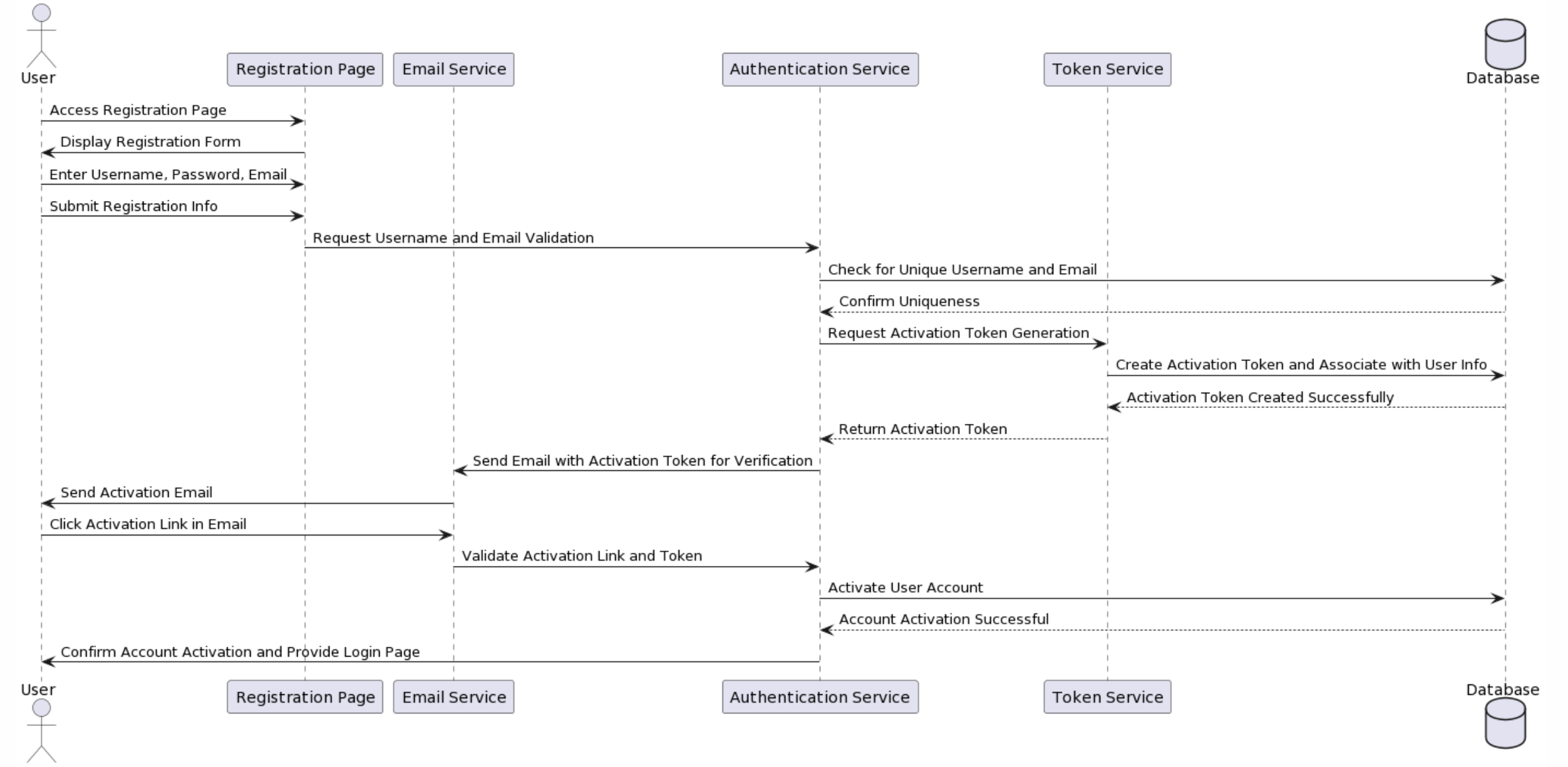
# DESIGN

## 4.1 Overall Design

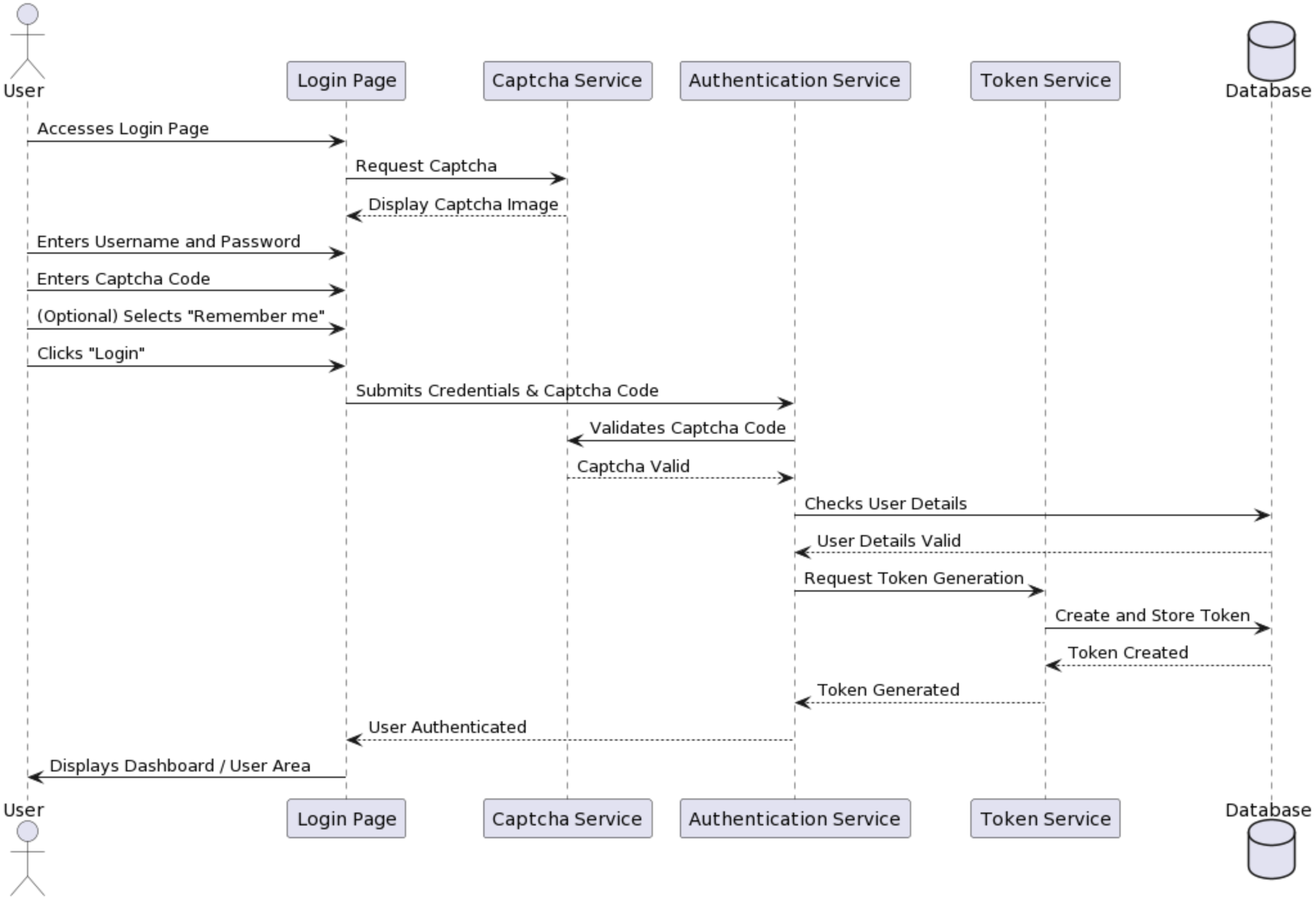
## 4.2 Domain Detail Design

### 4.2.1 User Domain

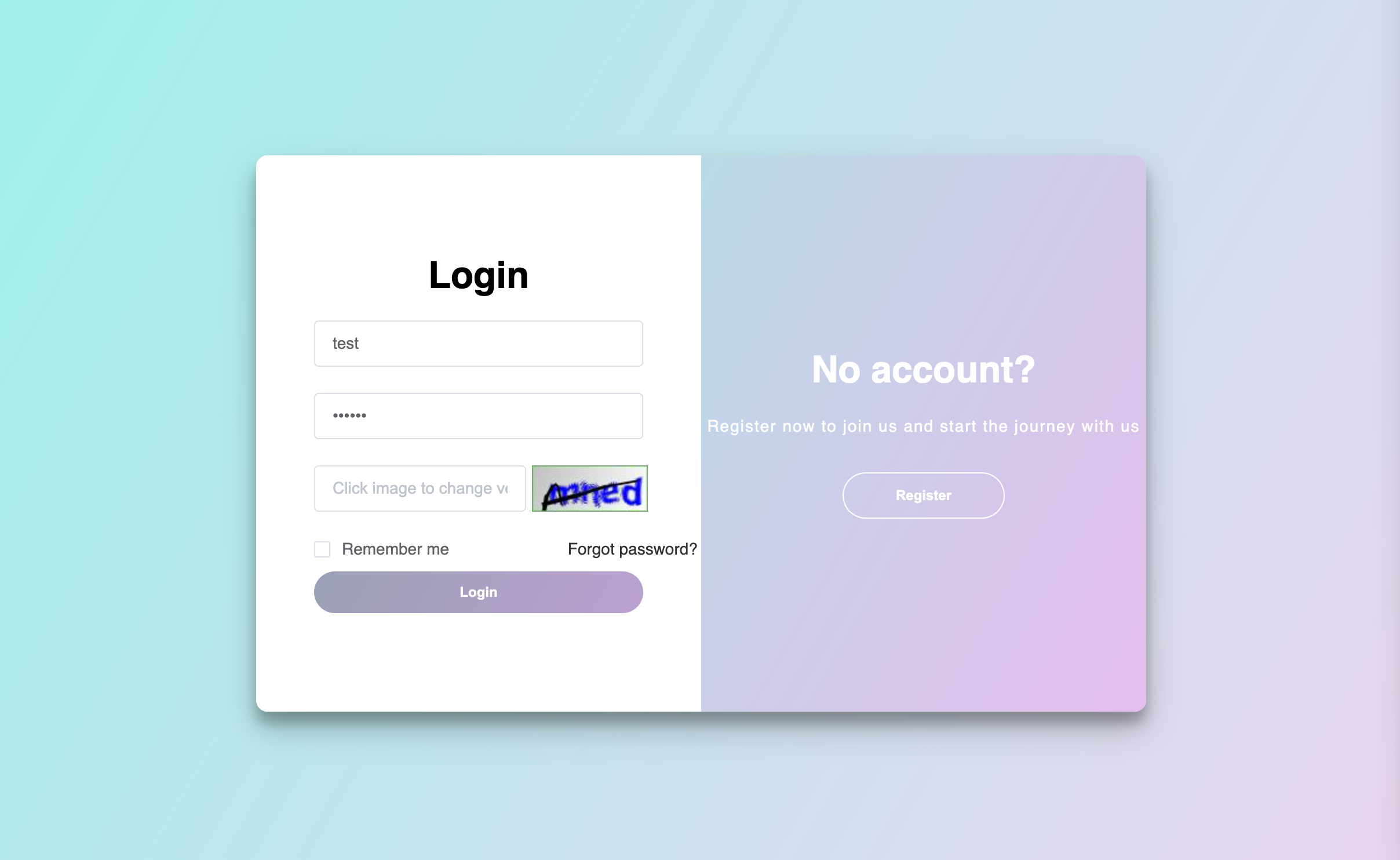
#### 4.2.1.1 User Register

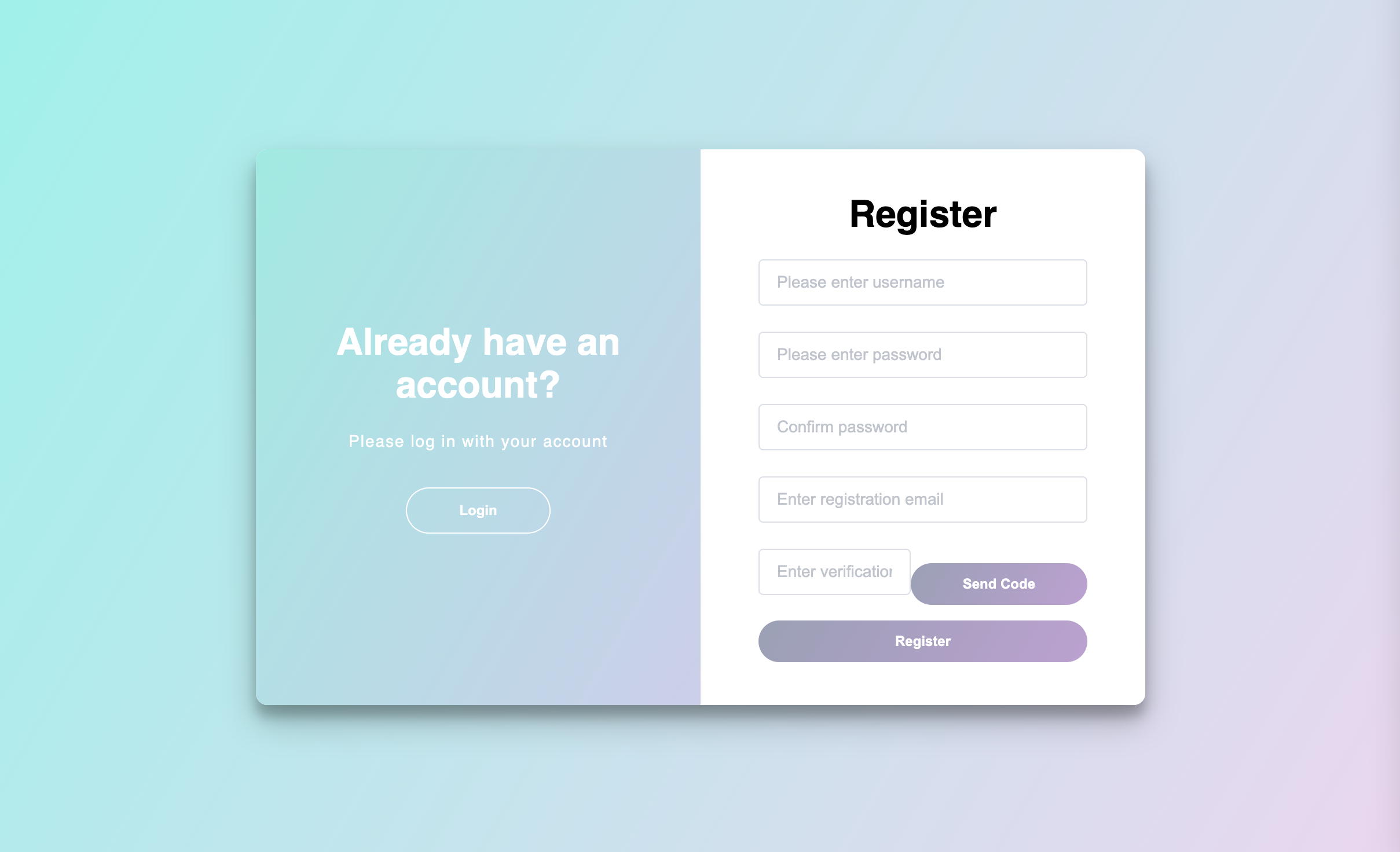


#### 4.2.1.1 User Login



# IMPLEMENTATION





# EVALUATION

# CONCLUSION

# REFERENCES

1. [1] Mell, P., & Grance, T. (2011). The NIST definition of cloud computing. [↑](#endnote-ref-0)
2. [2] Armbrust, M., et al. (2010). A view of cloud computing. Communications of the ACM, 53(4), 50-58. [↑](#endnote-ref-1)
3. [3] Bello, S. A., Oyedele, L. O., Akinade, O. O., Bilal, M., Davila Delgado, J. M., Akanbi, L. A., Ajayi, A. O., & Owolabi, H. A. (2021). Cloud computing in construction industry: Use cases, benefits, and challenges. Big Data Enterprise and Artificial Intelligence Lab (Big-DEAL), University of the West of England (UWE), Bristol, United Kingdom. Available online 17 December 2020, Version of Record 1 January 2021. [↑](#endnote-ref-2)
4. [4] Amazon Web Services, Inc. (2021). "Amazon Simple Storage Service (Amazon S3) documentation." [↑](#endnote-ref-3)
5. [5] Microsoft. (2021). "Azure Blob Storage documentation." [↑](#endnote-ref-4)
6. [6] Google Cloud. (2021). "Google Cloud Storage documentation." [↑](#endnote-ref-5)
7. [7] Weil, S. A., et al. (2006). "Ceph: A scalable, high-performance distributed file system." In Proceedings of the 7th symposium on Operating systems design and implementation. [↑](#endnote-ref-6)
8. [8] The Rook Authors. (2021). "Rook: Open-source, cloud-native storage for Kubernetes." [↑](#endnote-ref-7)
9. [9] Buuck, B. (2023, December 12). A Guide to Decoupled Storage with StarRocks and MinIO. MinIO Blog. Retrieved from https://blog.min.io/tag/open-source/ [↑](#endnote-ref-8)
10. [10] MinIO. (n.d.). Erasure Coding. MinIO Object Storage for Linux. Retrieved [the date you accessed the website], from https://min.io/docs/minio/linux/operations/concepts/erasure-coding.html [↑](#endnote-ref-9)
11. [11] Auernhammer, A. (2022, December 16). MinIO Best Practices - Security and Access Control. MinIO Blog. Retrieved from https://blog.min.io/s3-security-access-control/

    [12] You, E. (2014). "Vue.js: A progressive, incrementally-adoptable JavaScript framework for building UI on the web." [↑](#endnote-ref-10)
12. [13] DoNotApply(2022, Aug, 17) Vue vs Angular, which has the steeper learning.Medium. https://medium.com/@donotapply/vue-vs-angular-which-has-the-steeper-learning-curve-b75df9ab9a0a [↑](#endnote-ref-11)
13. [↑](#endnote-ref-12)
14. [↑](#endnote-ref-13)