Main Report submitted to
The Department of Computer Engineering of Çankaya University
in fulfillment of the requirements for
CENG 407 Innovative System Design and Development I

PLATFORM FOR MUSİC, BOOK AND MOVİE RECOMMENDATİON

By

Ege Varolgüneş	201611062
Enzel Ege Sarı	201612057
Yaren İpek	201711038
Pelinsu Serimer	201711058

Group No: 07

Supervisor: Assist. Prof. Abdül Kadir Görür

Çankaya University
Department of Computer Engineering
Fall 2020

STATEMENT OF NON-PLAGIARISM

I hereby declare that all information in this assignment has been obtained and presented in accordance with academic rules and ethical conduct and the work I am submitting in this document, except where I have indicated, is my own work.

Date: 05.01.2021

Group Members’ Name and Surname	**Signatures**
Ege Varolgüneş	
Enzel Ege Sarı	
Yaren İpek	
Pelinsu Serimer	

TABLE OF CONTENTS

TABLE OF FIGURES

ABSTRACT

Several modern recommender systems are developed to help users to devote more time to themselves by proposing relevant suggestions in the requested context. This project is designed with the aim of improving present recommender systems, by bringing multiple recommender systems which focus on different areas together in one platform. Conducting a literature review provides an opportunity to learn how to design an innovative system and understand the challenges along the way.

KEY WORDS: Recommender System, Software Development, Deep Learning.

ÖZ

Kullanıcıların talep edilen bağlamda ilgili öneriler sunarak kendilerine daha fazla zaman ayırmalarına yardımcı olmak için birkaç modern tavsiye sistemi geliştirilmiştir. Bu proje, farklı alanlara odaklanan birden fazla öneri sistemini tek bir platformda bir araya getirerek mevcut öneri sistemlerini geliştirmek amacıyla tasarlanmıştır. Bir literatür taraması yapmak, yenilikçi bir sistemin nasıl tasarlanacağını öğrenmek ve yol boyunca karşılaşılan zorlukları anlamak için bir fırsat sağlar.

ANAHTAR KELİMELER: Öneri Sistemi, Yazılım Geliştirme, Derin öğrenme

1.INTRODUCTION

The popularization of different types of web services leads the recommender system concept to become popular and varied. The concept of recommender system aims to suggest relevant items to users. Various item examples can be given depending on industries. (Rocca, 2019) The scope of this project is to develop a platform which involves multiple recommender systems with various suggestion items. The platform will focus on recommendations of the following items: movies to watch, books to read, music to listen to. Diverseness of the items will not only offer variety to the options proposed to the users, but also reduce the item searching period required by the users to find the right option. The platform developed will be more flexible for the users decision changes. In addition to these advantages, it is expected that the platform will be more appealing to users having different requests in context. Within the scope of this work, The necessary literature review (Chapter 2), software requirement specifications (Chapter 3) and the software design description (Chapter 4) is discussed and ended with a brief conclusion.

2. Literature Review

2.1. Main Findings

2.1.1 Recommender Systems

The recommender systems are more valuable and important attributes to our daily lives. The rise of services such as Netflix, Spotify, Soundwave or even shopping pages like Amazon or Trendyol made the customized recommendations more and more popular. Recommender systems help the companies and customers tremendously. Recommender systems are basically algorithms that makes suggestions to the users according to their preferences. (Rocca, 2019)

The recommender systems use special algorithms to figure out what the customer likes and provides relevant suggestions. Few real-life examples for such systems are Soundwave, GoodReads and MovieLens. Soundwave makes suggestions based on the location of the user and gives the music trends that the user is currently at. (O'Hear, 2013) GoodReads gives out recommendations according to genres of books the user enjoyed previously. (Strickland, 2009) MovieLens gives the user lots of movie options to rate and from the users ratings it understands the users behavior and recommends the user other movies they will enjoy to watch.

These recommender systems use different recommender systems such as, but not limited to, collaborative filtering-based system, content-based system, hybrid system, Popularity based system etc.

Collaborative Filtering (CF) based methods uses the ratings given by the users as the main source of the behaviors to make a recommendation. One downside is that the rating may be sparse which causes the CF based methods to decrease their performance. The CF based methods gather information on different users and make a well calculated recommendation with the other user’s behaviors included. (Wang, Wang, & Yeung, 2015)

Content Based Filtering method uses the previous behaviors of the user such as purchases or consumptions and makes recommendations based on those historical data. Content based filtering methods match up the attributes of a user behavior, in other word users interests, with the attributes of the items and the system makes the recommendations regarding these comparisons. (Lops, de Gemmis, & Semeraro, 2010)

Hybrid methods usually combine the content-based methods with collaboration-based methods to come up with a recommendation for the user. This method gives much more accurate recommendations and is currently used by most of the industries. There are two approaches one way is to create two different recommender system and combine the suggestions or combine the methods into one single recommender system. (Rocca, 2019)

2.1.2 Web Development

Web development is the work of developing and maintenance websites for the Internet or an intranet. Since developing a website for this project is one of the main goals, research conducted for some necessary terms and concepts regarding web development. This section of the literature review has the related research that helped us to choose the related architecture and protocol.

IBM defines the web services as a generic term for an interoperable machine-to-machine software function that is hosted at a network addressable location. (IBM, 2020) When creating a web service there is a need for an architecture and a protocol. An architecture describes how the databases, applications and middleware systems work with each other. (Banga, 2020)A protocol is an agreement on how the web services will be exchanging data.

Service-Oriented Architecture (SOA) is an architectural style that uses services of the network such as the web for building software applications. Applications in SOA ,as the name suggests, are based on services. SOA enables the users to use the existing assets of the already existing IT infrastructures. (Mahmoud, 2005) (Bean, 2009) OA has 6 main components. These are service, policies, endpoints, contracts, messages and service consumers. Services are the most important component of the SOA. Each service offers a different functionality. Policies define the restrictions and the terms such as security, auditing, etc. Endpoint is a URL which identifies the location on the built-in HTTP service where the web services listener listens for incoming requests. (IBM) Messages are the communication unit of the SOA. There are various kinds of Messages such as REST, SOAP, JMS. Consumers communicate with services by messaging. A consumer can be another application or a service.

RedHat defines API (Application Programming Interface) as a set of definitions and protocols for building and integrating application software. (Red Hat) Main purpose of an API is to let an application's methods to other applications and respond to data requests easily and quickly. It provides saving time and money. APIs give us flexibility and provide opportunities for innovation.

REST (Representational State Transfer) is an architectural style which provides a fast client-server communication . It is not a standard, so it supports different data formats such as JSON, XML, HTML, etc. REST is stateless; it provides scalability and visibility. It is cacheable. REST has a layered system because of the client-server architecture. REST lets developers to use HTTP methods such as GET, POST, HEAD, PUT, DELETE and PATCH.
SOAP is a protocol for the exchange of information in a distributed environment. (IBM, 2020) SOAP is an XML based messaging protocol. It only supports XML format and these messages are human-readable. For this reason, it is in need of more bandwidth and compute power. SOAP can ride on different transfer protocols such as HTTP, SMTP, UDP, etc.

2.1.3 Mobile Development

Mobile development is the process of creating software applications that run on a mobile device. (Amazon) This section of the literature review focuses on mobile development strategies. These strategies are native development, hybrid development, cross-platform development and progressive web applications approach.
Native development aims to develop software applications for use on a particular target platform or device. Major mobile development platforms are iOS by Apple, Android by Google, and others, such as Microsoft Windows phones or RIM Blackberry. (Latif, Lakhrissi, Nfaoui, & Es-Sbai, 2016). In this approach of mobile development, native mobile developers choose the platform to be worked on by deciding how they link the application with the underlying operating system. Native mobile developers use a specific Software Development Kit (SDK) and programming language for each target platform. For example, to develop a native platform for the Android operating system, developers use Android Studio for SDK, and Java as programming language, while another developer uses XCode for SDK, and Objective-C/Swift as programming language to develop a native platform for iOS.

Hybrid development is the blend solution that consists of both native and web approaches. Hybrid platform developers embed the code written in HTML, CSS, and JavaScript into a native application. (Kidecha, 2020)

Cross-platform development aims to develop software applications for use on different target platforms or devices. Cross-platform approach has been motivated by the difficulty occurred in the native approach. Development phase is easier and more efficient in the cross-platform approach, compared to the native approach. Because in the cross-platform approach, developers create a code base, then only make small changes on the code, if it is required by the platform. They do not need to use different SDK's, platforms or programming languages for multiple platform development. (Latif, Lakhrissi, Nfaoui, & Es-Sbai, 2016)

There is another option called "progressive web applications". This approach consists of browser-based HTML, CSS, and JavaScript codes and accessed over the Internet. (Summerfield, -)Since development of a web application is one of the goals of the project, development of a progressive web application would be redundant. Hence, this approach will not be reviewed any further.

2.1.4 Deep Learning

While implementing a recommender system, usage of deep learning algorithms are more common to use simply because deep learning can efficiently learn the underlying behaviors of the input data and give related outputs. Deep learning has a high level of flexibility and is really powerful for sequentially modeling tasks or decisions. Also, Deep learning is really good option to model the interactions of the non-linear data which a person's behaviors can be explained by non-linear equations most of the time. (Le, 2019)

Through in-depth research on the techniques of deep learning that can be used in this project 5 suitable techniques found. Multi-Layer Perceptron based recommendation, autoencoder based recommendation, Convolutional Neural Networks Based Recommendation, Recurrent Neural Networks Based Recommendation and Restricted Boltzmann Machines Based Recommendation are techniques that are going to be discussed in this section.

Multi-Layer Perceptron (MLP) based recommendation method is a multiple hidden layer feedforward neural network. The hidden layers are placed between the input layer and the output layer. MLP is an effective and concise network model that can make an approximation of any measurable function to any needed accuracy. The feature representation of MLP is straightforward even if it is not expressive as the other recommendation methods.

Autoencoder based recommendation method is one of the unsupervised models. Autoencoder attempts to reconstruct input data in the output layer. For the important feature representation of the input data, the bottleneck layer is used. autoencoders used to learn the lower dimensional feature representations at bottleneck layer.

Convolutional Neural Networks (CNN) based recommendation is also a feed forward neural network like the MLPs but there are convolution layers and pooling operations included. CNN can capture the global features as well as the local features. This increases the model's efficiency and accuracy remarkably. CNN's mostly used to extract features from texts, audios, videos and images. The usage of CNN is more common than the other methods. (Ying , et al., 2018)

Recurrent Neural Networks (RNN) based recommendation method is suitable for sequential data modelling. It has memory and loop attributes to remember the previous computations. Therefore, they can be used for temporal dynamics of interactions and sequential patterns of user behaviors. But this method can be used for session-based recommendations most of the time. (Hidasi, Baltrunas, Karatzoglu, & Tikk, 2015)

Restricted Boltzmann Machines (RBM) based recommendation method is a two layered neural network model. It has one visible and one hidden layer. It can easily be loaded to a deep network. The visible layer of the RBM consists of binary values, therefore the ratings shown in a vector to fulfill this restriction. In a recommendation system each user has a unique RBM with a shared parameter. This means that each user will tell what they want and rate those needs and the RBM will gather the behavior information. (Ali, 2019)

2.2. Decision

Deciding upon the methodologies that will be used in the development phase of the project has a great value. The correct decision of the methodologies will lead to a correctly working recommendation platform, hence a correctly completed project. The system is going to be a standalone system.

First decision is to decide on the recommender system algorithms from the options that are mentioned in the previous section. Each algorithm has its perks and drawbacks but using a content-based filtering system will benefit the platform better than the other options since conducting a questionnaire to retrieve the customers behaviors and then giving out the recommendations from the outcome will be much faster in receiving the recommendations from the platform.

Deciding on the web services were challenging because the perks of each method were too many. In the end the decision made upon the usage of REST API. An API enables us to use a full functionality of another application while the web services only allows us to use specific tasks. Therefore, APIs are a better option. REST APIs use fewer resources and support various formats of messages. For these reasons, development of web services will be done by using REST API.

To decide which mobile development approach to be used, options that are mentioned in the previous section are evaluated. Since in native development for each platform is a costly and time-consuming process, most popular mobile development companies use the cross-platform approach. They avoid rewriting the code for different platforms. Even the hybrid approach has the same code-shareability with the cross-platform approach, it is mostly used when user experience and performance is not a priority for the application, hence it will not be a great option to consider in our project. However, native approach is more appropriate in our case. In the scope of mobile development, our platform has been planned as only an Android application to limit the scope of the project and reduce the cost of training. Also, it has been seen that native approach has richer libraries, higher performance, and higher user experience.

Decision of the deep learning technique to be used is more crucial than the decisions of the other components of the project. Hence, the decision for the deep learning method needed more thinking and trial and error to make sure of the performance. Even though the decision upon the technique for deep learning is to be decided, some of the techniques are eliminated down to two possible techniques. Further in the development phase the decision for the algorithm will be made between CNN and RBM methods.

3. SOFTWARE REQUIREMENTS SPECIFICATIONS

3.1. Introduction

3.1.1. Purpose

The purpose of this document is to describe the system which is called QuiRec. This platform aims to make recommendations for books to read, movies to watch and music to listen to. This documentation includes detailed information about requirements of the project. It showcases the function and interface requirements as well as the performance requirements and the security requirements. The constraints of this project are also added to the description of the requirements.

3.1.2. Scope of Project

In the existing recommender systems, the users who want to get personalized recommendations upon the books, movies and music tend to use multiple different platforms. This situation leads up to the waste of time for the user. This project comes in handy to reduce the time wasted on getting the recommendations. Since the project combines 3 different recommendation mediums into one single recommendation platform called QuiRec.

The purpose of QuiRec is to combine 3 different recommendation systems into one to give simultaneous recommendations for books, movies and music. With this the expected time spent on getting the personalized recommendation will be diminished.

The platform will be implemented on a web application and a native android application to support all types of users.

3.1.3. References

Varolgunes, E., Sarı, E. E., İpek Y., Serimer P. (2020). *Literature Review for a Platform for Music, Book and Movie Recommendation*. Retrieved from Github:https://github.com/CankayaUniversity/ceng-407-408-2020-2021-A-Platform-for-Music-Book-and-Movie-Recommendation/blob/main/LiteratureReview_Group07.pdf

3.1.4. Overview of the Document

The documentation consists of 3 chapters. The first chapter is for the introduction and the main aspects of the project. Second and third chapters have more detail upon the details about the project. Each chapter focuses on different areas of the project. Second chapter is dedicated to the explanation of the product perspectives and user characteristics. Meaning the non-technical requirements of the project is explained in chapter 2. In chapter 3 the technical requirements of the project is listed within five different aspects of the technicalities. These details are listed for the software engineers and to guide them through the project to achieve the correct product.

3.2. Overall Description

3.2.1. Product Perspective

QuiRec is a recommendation platform which will be developed both as a web application and a mobile application for Android Operating System (OS). With this platform, users can receive personalized recommendations upon the books, music, and movies without using three different platforms to receive recommendations on three different items. (Varolgüneş, Sarı, İpek, & Serimer, 2020)

3.2.1.1. Development Methodology

Software Development Life Cycle (SDLC) is a process for analyzing and planning, designing, developing, testing and deploying a system. There are multiple SDLC models. (Balaji & Murugaiyan, 2012) Waterfall software development model is one of the most-used SDLC models in the software development industry. It is the linear development model because it consists of sequential phases. In this approach, the entire process of software development is considered as a single project. (Aparna, 2020) Besides, the waterfall development model has faster project deliverance. Figure 1 shows the implementation of waterfall development model.

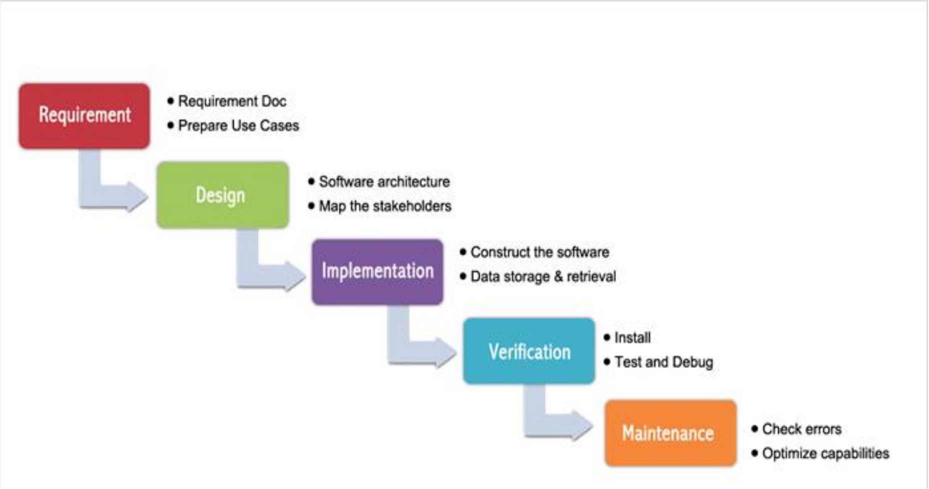


Figure 1 Waterfall Model

Waterfall development model requires that project requirements are clear at the very beginning. This methodology divides project development time within the phases, and each phase should be completed before moving to the next phase. Each phase is completed with a specific software product. At the end of all phases, it is expected that the software project has become available and applicable.

The waterfall development model has been chosen as the development methodology of the QuiRec project because the project' requirements were clear from the project planning phase. Also, since the project development time is divided between the phases, this helps focus better on each phase individually.

3.2.2. User Characteristic

3.2.2.1. Users

- Users must have a browser on a personal computer (PC) or another smart device which has a browser to use the QuiRec platform as a web application.
- Users must have a smart device which has an Android OS having version over 8 to use the QuiRec platform as a mobile application.
- Users must have access to the internet and must have any other devices required to use applications.
- Users must know how to use a web browser and/or mobile applications.

3.2.2.2. Admin

- The admin must have a browser on a personal computer (PC) or another smart device which has a browser to use the QuiRec platform as a web application.
- The admin must have a smart device which has an Android OS having version over 8 to use the QuiRec platform as a mobile application.
- The admin must have access to the internet and must have any other devices required to use applications.
- The admin must know how to use a web browser and/or mobile applications.

3.3. REQUIREMENTS SPECIFICATION

3.3.1. Interface Requirements

3.3.1.1. User Interfaces

The user can get recommendations for any kind of music, books and movies, using the recommendations page in the application. The user must have an account to use the application.

If not, users can easily register from the registration page. If a user needs to change something from their profile he/she can change easily. The user can logout from his/her account after his/her job is finished.

The admin will be responsible for observing the recommendation performances and maintaining the quality of the platform by operating necessary updates on the system. For these operations, the admin will have an administration page. The admin will be registered to the system by the system developer.

3.3.1.2. Hardware Interfaces

Server Side

For the first stage, the server will have a large hard disk enough to hold 5000 user data and up to 10000 music, book and movie information gathered online. It can be increased later if necessary.

Client Side

Any personal computer which supports any type of browser would be acceptable to use QuiRec as a web application.

Any smart device which supports any type of browser or supports mobile applications would be acceptable to use QuiRec as a mobile application.

3.3.1.3. Software Interfaces

Server Side

For the first stage, the database is designed enough to keep records of 5000 users, and up to 10000 music, book and movie records. It can be increased later if necessary. Any other software interface is inessential.

Client Side

The web application developed will be accessible over any browsers that run on any OS, supporting at least HTML version 5 and having Java enabled.

The mobile application developed will be accessible over any browsers that run on Android OS, supporting at least HTML version 5 and having Java enabled. It also will be accessible on the application store supporting Android version 8.

3.3.1.4. Communication Interfaces

The application’s server and client sides will communicate to each other to transmit data over the Transmission Control Protocol/Internet Protocol (TCP/IP), which is the default protocol, and over the HyperText Transfer Protocol (HTTP).

3.3.2. Functional Requirements

3.3.2.1. Main Page

The first page that will meet the user in the application will be the main page. This page will introduce the application to the user and with the links, the user is directed to the login or registration pages.

3.3.2.2. Login Page

The user on the login page is directed to the recommendations page by activating her/his account with her/his username or email and password. The other option in this page is, if the user does not have an account, he/she can use the register link in this page and directed to the registration page to create an account for the application. Also, if the user forgets his/her password, he/she can easily change his/her password with clicking the link displayed on the login page.

3.3.2.3. Registration Page

In this page, the user can register by filling the required information which are email address, username, and password. After filling in the required information, the user can continue and register to the application.

3.3.2.4. Forgot Password Page

The forgot password page, which is accessed over clicking the link displayed when the user clicked to the “forgot password” button on the login page, allows the user to change his/her password. The user will be directed to this page using the link sent by the system to the user’s mail address.

3.3.2.5. Change Password Page

In this page, there are three text areas that the user needs to fill. The first area is for the user to type old password s/he has. The second area is for the user to type the new password he/she sets. The third area is for the user to re-enter the new password specified by the user for confirmation. After changing the password, the user can activate his/her account with the new password.

3.3.2.6. Questionnaire Page

This page is where the user’s preferences are gotten. First, the user must select at least one of the songs, movie or book genres. Then the user will choose which style of the selected genre they want to get recommendations for. After, the system will direct to the recommendations page to show the list of recommendations.

3.3.2.7. Profile Page

The profile page shows the user’s account details. The user can access to his/her questionnaire page to change their preferences by clicking the related button from this page. Also, the user can change his/her password from this page by clicking the related button.

3.3.2.8. Recommendations Page

Recommendations based on the preferences made by the user on the questionnaire page are shown to the user on this page. From this page, the user can view the music, books and/or movies recommended by the application. If the user asks for a new list of recommendations, he/she can go to the questionnaire page with the “change my recommendation” option, answer the questions again, and get the type and genres of recommendations he/she want.

3.3.2.9. Browser Page

This page helps users to see all movies, books, and music in the database of the application. They can select and see details of an item, or search an item using the search bar on this page, and get recommendations without changing their preferences.

3.3.2.10. Details Page

This page will open when the user clicks onto an item from the recommendations page or browser page. Detailed information related to the clicked item will be displayed on this page. Also, if the user has accessed this page over the recommendations page, the user will see a rating option where they will rate the recommendation if it meets their preferences.

3.3.2.11. Administration Page

This page will only be available for the admin, as well as the admin will have only this page. The admin will see and monitor the recommendation ratings from this page. Also, the page will contain options which helps admin to make operations over the recommendation system.

3.3.2. Performance Requirement

In order to accomplish user satisfaction, the pages of the system needs to be changed smoothly. If the necessary information is provided; sign up, login, logout, and listing recommendations operations will take no more than 5 seconds. Stable internet connection should be established. The minimum requirements for the system to run smoothly is:

1. CPU: Intel i5-8250U or upper version.
2. RAM: 8GB or more.
3. Operating System: Windows7 or upper. GNU/Linux.

3.3.3. Software System Attributes

3.3.3.1. Usability

GUI will be designed to make users comfortable and familiar with the mobile and web application. Applications will be simple and comprehensible so that people from different groups of ages can use the applications.

3.3.3.2. Adaptability

The user interfaces for two different user types, which are user and admin, will be different from each other.

3.3.3.3. Maintainability

Having regard to the received feedback, new features and improvements can be implemented to the project.

3.3.3.4. Portability

The web application of the project will run properly on all web browsers. The mobile application of the project will run on Android versions 8 and higher.

3.3.3.5. Performance

The recommendation system should not be working unless the user filled out the questionnaire to get recommendations.

3.3.4. Safety Requirement

To provide the security of the application, the users will not have direct access to the database. Users have to choose their passwords at least 8 characters long and the passwords must include upper-lower case letters, a number, and a symbol. All passwords will be hashed, and the hashed forms will be stored in the database. Authentication will be observed.

Also, secure network protocol TCP/IP protocol in PostgreSQL will be used. With access tokens, the users will only have access to their own accounts, not the other users'.

4.Software Design Description

4.1. Introduction

4.1.1 Purpose

The purpose of this document is to describe the QuiRec system. This platform aims to make recommendations for books to read, movies to watch and music to listen to. This documentation includes detailed information about the designing aspects of the project. It shows the architectural design and the use case realization of the QuiRec system.

4.1.2 Scope

The QuiRec will be the first recommender system that combines the mostly used recommender systems. The complete design descriptions of the system will be given in this documentation.

4.1.3 Overview of document

This document consists of 3 chapters. The chapter one gives out the scope and the purpose of the document. Chapter two gives detailed information about the architecture design. The subchapters of chapter two consists of the design approach, used technologies, class diagrams of the system, architecture design of QuiRec and last but not least the activity diagram of the system. Chapter three has further information on the user related design descriptions. User Interface (UI) Design Descriptions and the screen shots of the UI system is given.

4.2. ARCHITECTURE DESIGN

4.2.1 Architecture Design Approach

Primary purpose of the QuiRec project is to build a platform which combines recommendations of multiple items and presents them to users. The architecture of the project is designed with the aim of building and developing a system which meets users with maximum capacity and performance and presents them fulfilment recommendations.

Functionality of the system will be conducted with focusing to the user interfaces and focusing to the communication between the client and server. The user interfaces are designed considering the user experiences, simplicity, effectiveness, and comprehensibility. Also, for performance monitoring it is important to understand the user's satisfaction. In further sections, designs are explained in detail.

4.2.2 Technologies Used

Web application of the QuiRec project will be developed:

- using the Java programming language and Spring Boot framework in the backend part, with using the PostgreSQL for database management and Hibernate for ORM framework.
- using the HTML, CSS, JavaScript programming language and the Vue.JS and Vuetify JS frameworks in the frontend part.
- using the Python programming language and TensorFlow machine learning platform for recommendation models and functions.

Mobile application of the QuiRec project will be developed:

- using the same recommendation models and functions with the web application.
- using the Kotlin for Android development.

4.2.3 Class diagram

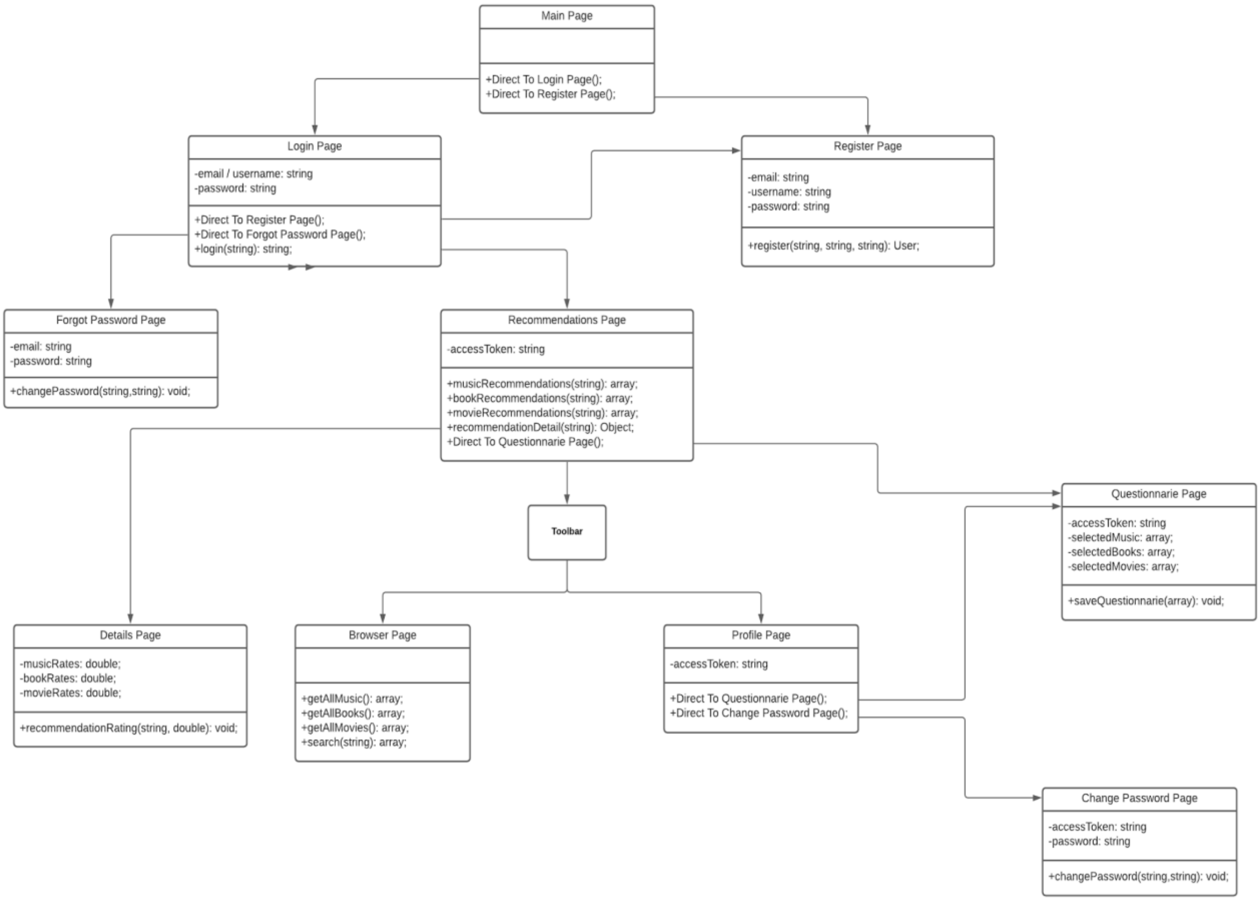


Figure 2 Class Diagram of The User

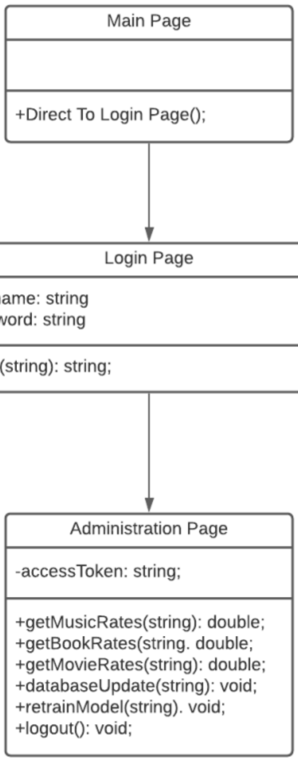


Figure 3 Class Diagram of The Admin

4.2.4 Architecture Design of QuiRec System

The QuiRec platform will serve up to users and admins. The main purpose of the system is to recommend their favorite genre of movie, books and music to users. The admin will use the system for monitoring the effectiveness of the recommendation model and making necessary updates on the database and recommendation model.

Users can register and login to the system; select or change genre preferences, receive recommendations, see recommendations in detail and rate the recommendations from 1 to 5 (1->Not related, 5-> Highly related compared to user preferences), which will be used for measuring the recommendation performance, change password information in the system, and logout from the system. Using the system, users can also see all movies, books and music which the system has, and perform a search operation without changing their preferences.

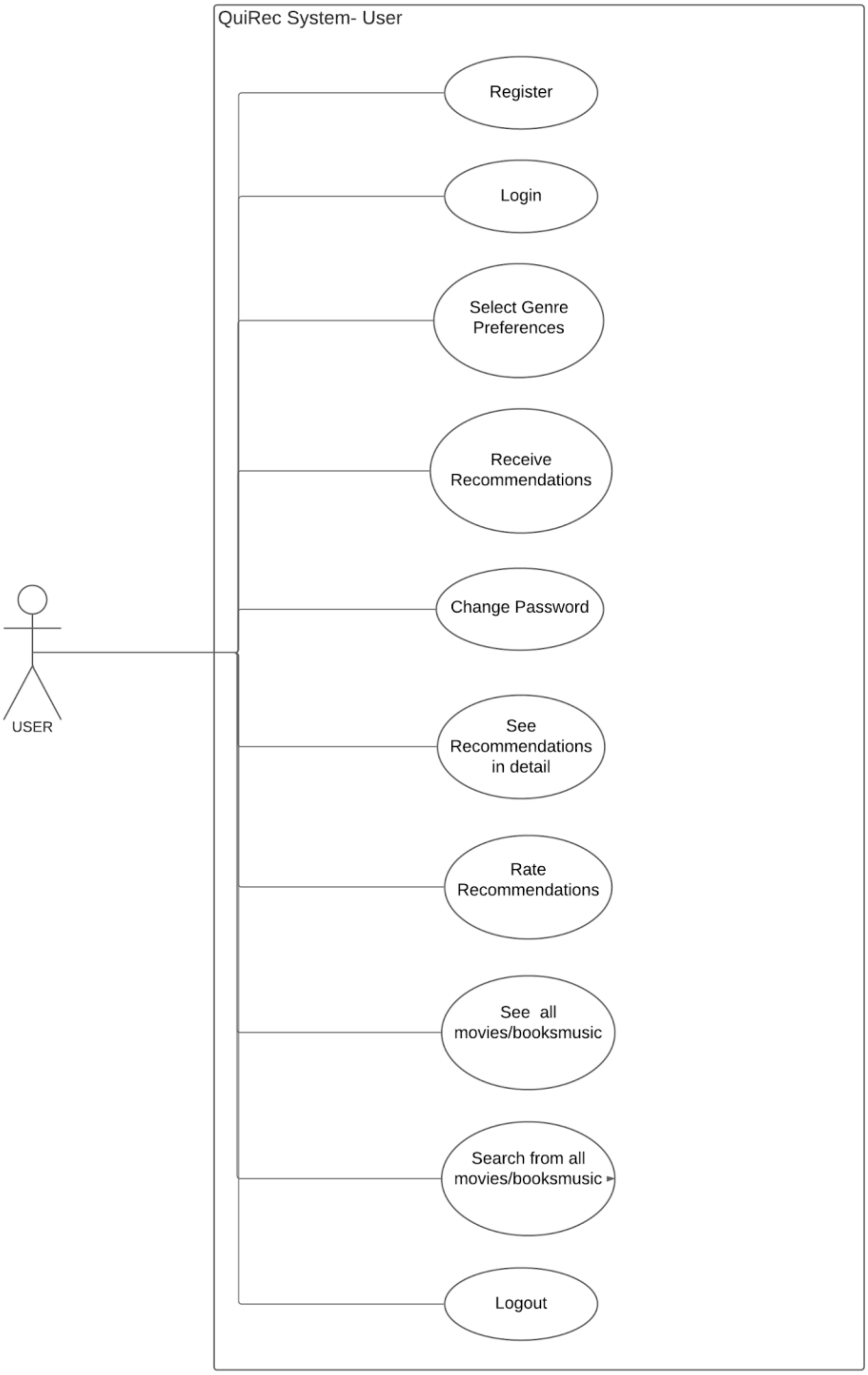


Figure 4 Use Case of the User

Table 1 The Position-Action Diagram for User

User Position	**Action**
Main page	The user can go to the login page.
Main page	The user can go to the registration page.
Login Page	The user can log into the system.
Registration Page	The user can register to the system.
Forgot Password Page	If the user forgets their account password, they can change it.
Change Password Page	The user can change the password of his/her account.
Questionnaire Page	The user selects genre preferences by answering the questions of the system to be used for recommendations.
Recommendations Page	The user can receive the movies, books or music recommended by the system.
Browser Page	The user can see all the movies, books and music which the system has in the database.
Browser Page	The user can perform a search operation inside the displayed data.
Profile Page	The user can see the information of his/her own account and can change password or recommendation preferences through this page.
Details Page	The user can see the recommendation in detail and rate the recommendations from 1 to 5. (1->Not related, 5-> Highly related)

The system is responsible for making recommendations according to user preferences.

Initially, the admin will be registered to the system automatically by the system developer. The admin will login to the system, monitor the performance of the model, update the database and retrain the recommendation when it is required, and logout from the system.

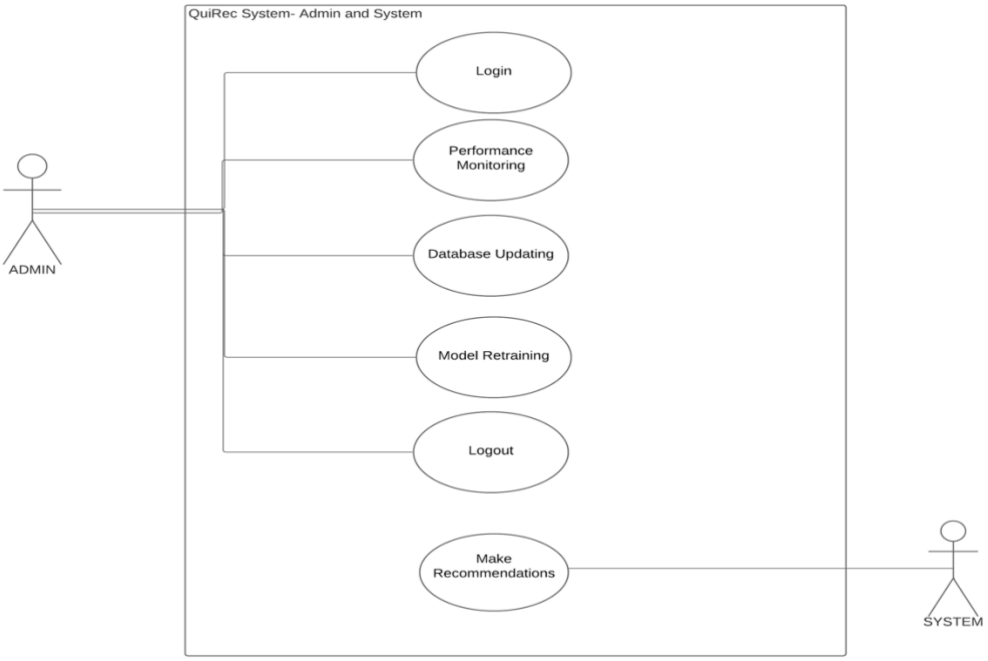


Figure 5 The Use Case Diagram of The Admin and System

Table 2 The Position-Action Diagram of Admin

Admin Position	**Action**
Main Page	The admin can go to the login page.
Login Page	The admin can log into the system.
Administration Page	The admin can monitor the performance of recommendations, update the database used for recommendations and retrain the recommendation model. The admin also can logout from the system from this page.

4.2.5 Activity diagram

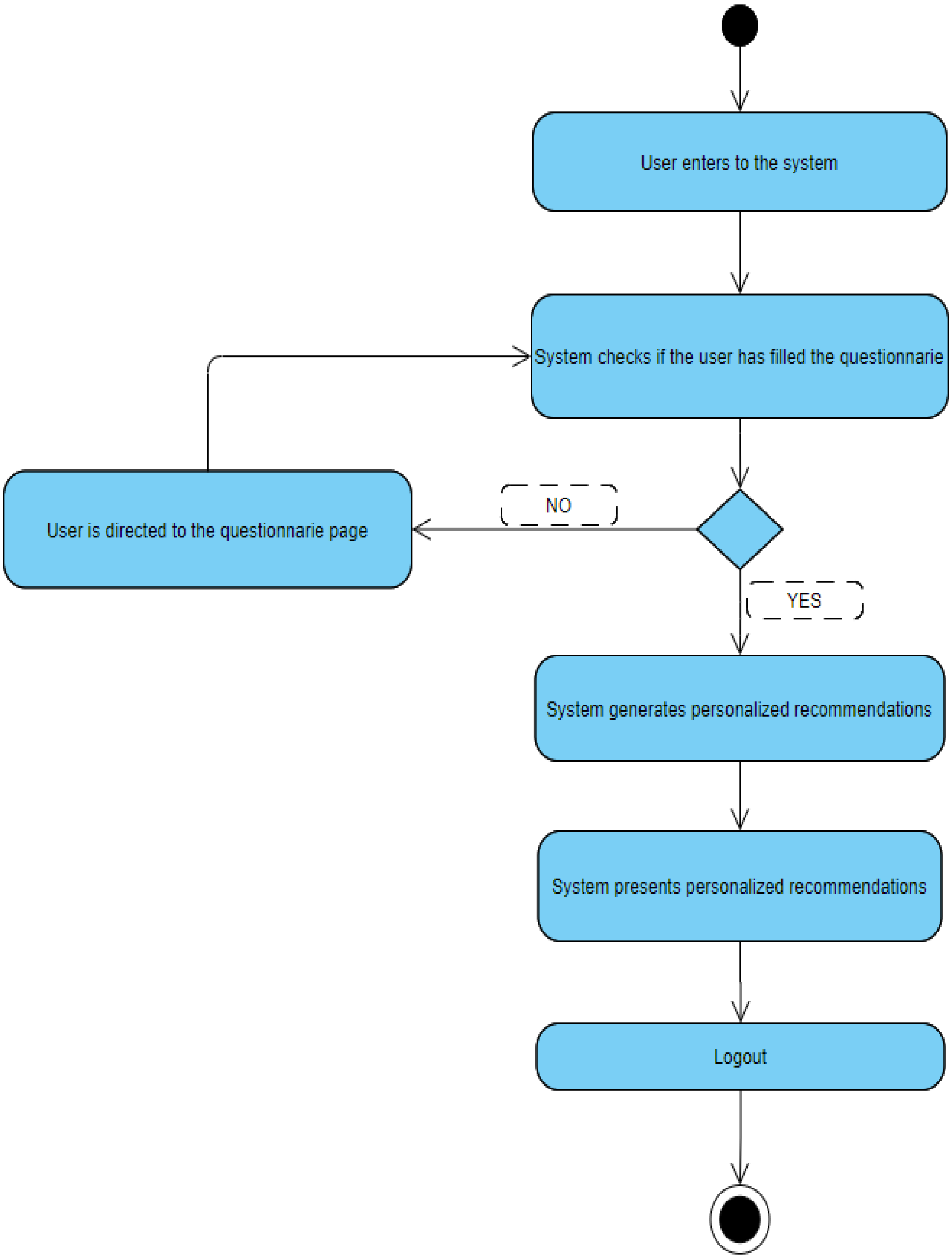

User enters to the system
System checks if the user has filled the questionnarie
NO
User is directed to the questionnarie page
YES
System generates personalized recommendations
System presents personalized recommendations
Logout

Figure 6 Activity Diagram of The User

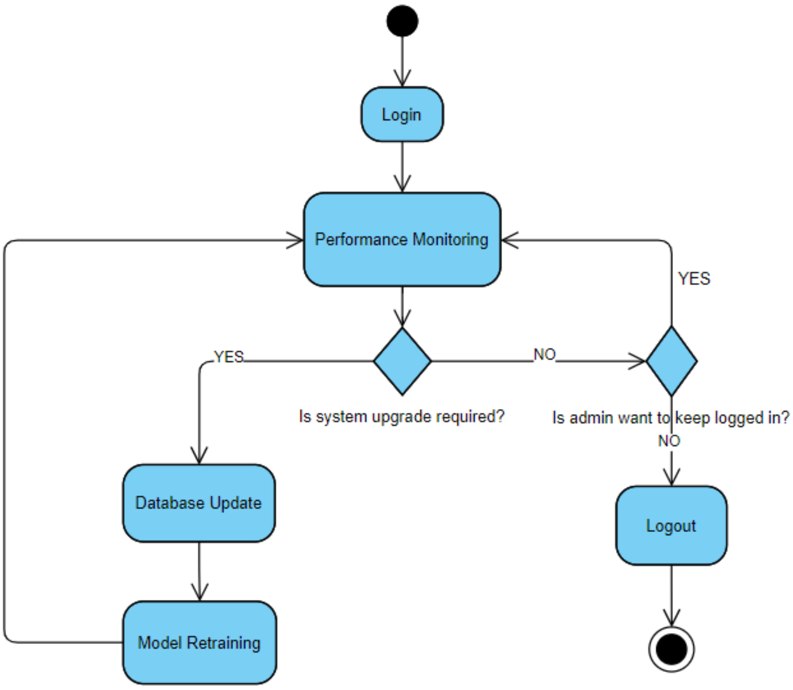


Figure 7 Activity Diagram of The Admin

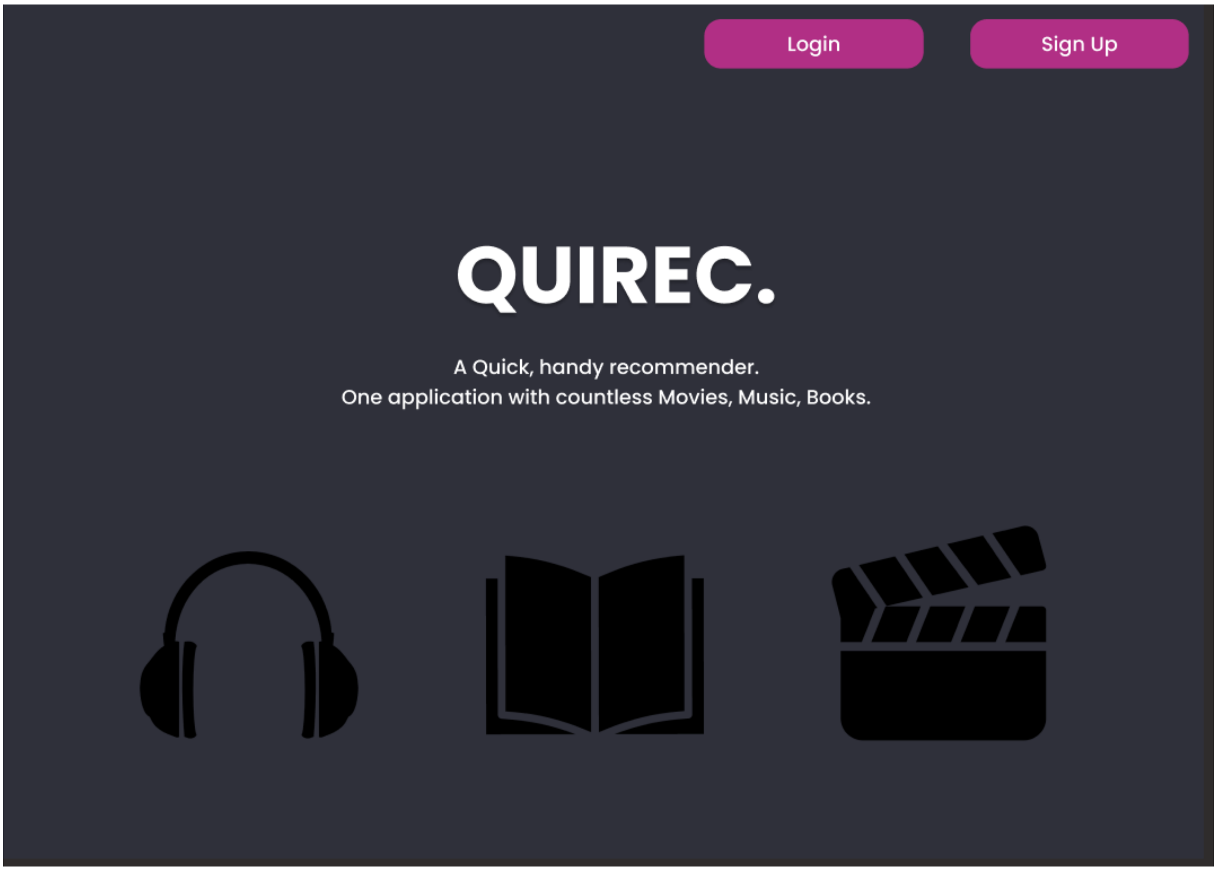
4.3. USE CASE REALIZATION

4.3.1. UI Design Descriptions

The QuiRec project has eleven different pages which are exemplified with UI screenshots and explained below. Both the web application and mobile application of the project will carry the same UI designs and descriptions. This part of the document contains only the screenshots designed for web application, but the mobile application will be developed with the same design, just by converting the designs applicable for a mobile device. Explanations of the designs will be the same for both the web and mobile applications.

Main Page:

The application meets the user on this page. In this page, the main description of the platform is displayed to the user, and with "login" and "sign up" buttons, the user is directed to a related page.



QUIREC.

A Quick, handy recommender.
One application with countless Movies, Music, Books.

Figure 8 The Main Page of QuiRec

Registration Page:

It is required for users to register to the system to use it. If they are not already registered, they will register to the system by providing necessary information displayed on the screen, which are "username", "email address", "password", and "password(again)". If the user is already registered, they will easily switch to the login page by clicking the "login" button.

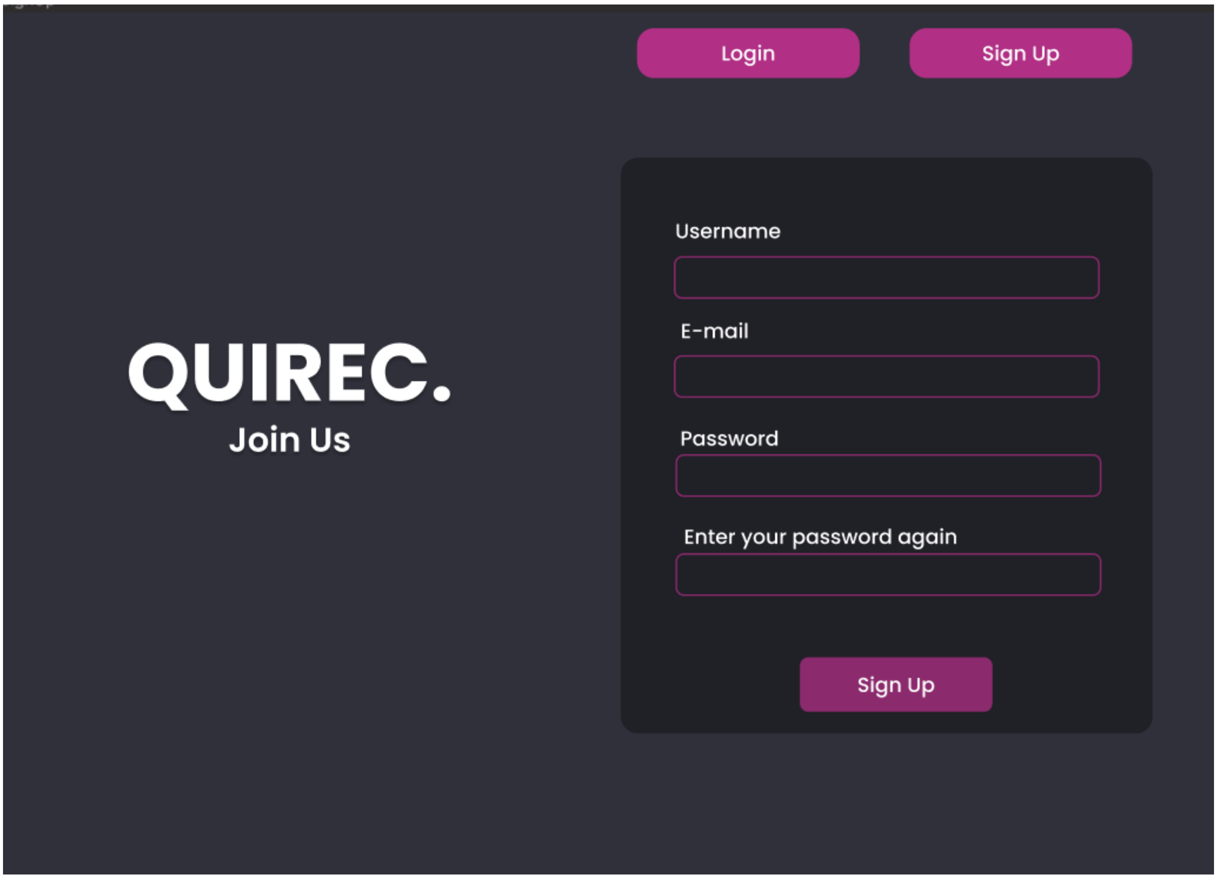
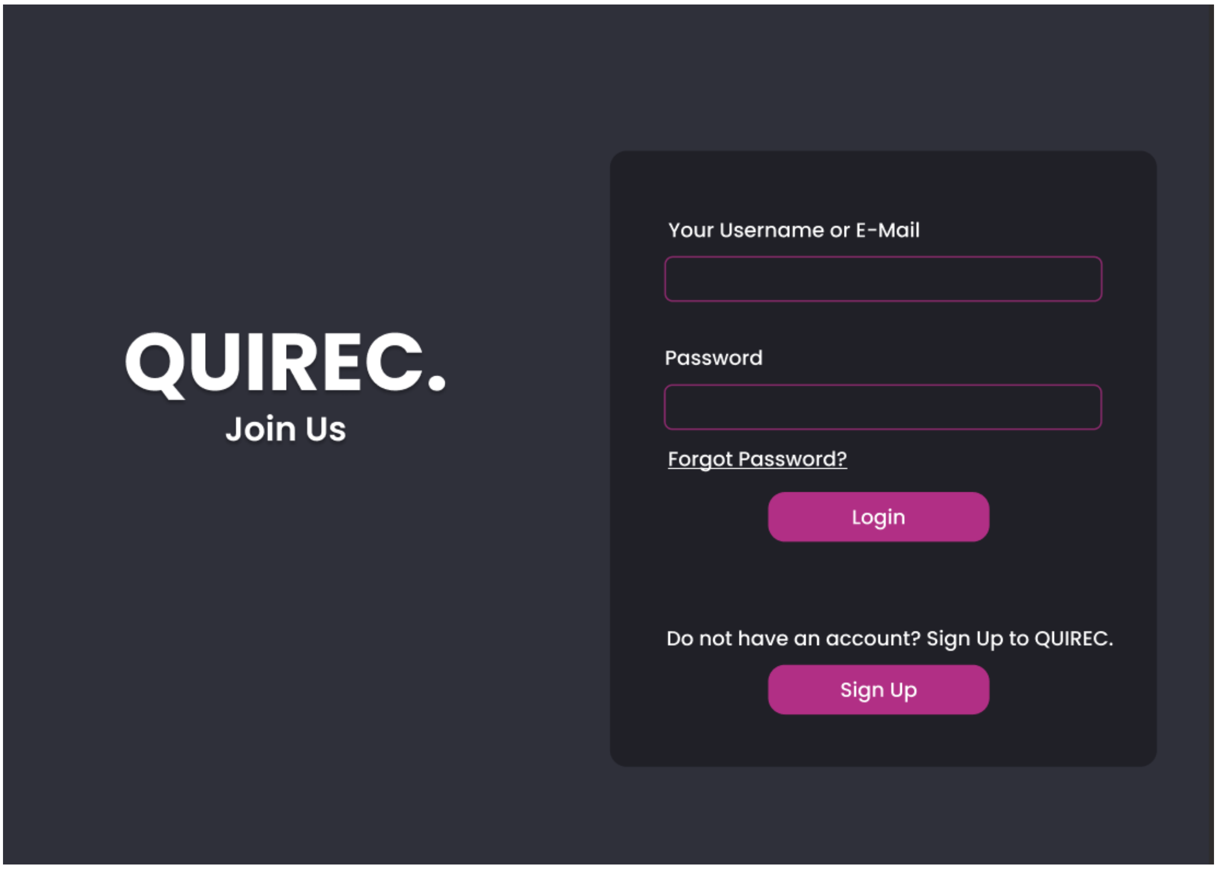


Figure 9 The Registration Page of QuiRec

Login Page:

Registered users will use this page to enter the system by filling two text areas: first one is “email/username” information, which allows users to enter the system whether with email addresses or usernames; second one is “password” information. If the user forgets his/her password, they will click to the “forgot password” button. And if a user is directed to the login page, but is not registered to the system, they will easily switch to the registration page by clicking the “sign up” button.

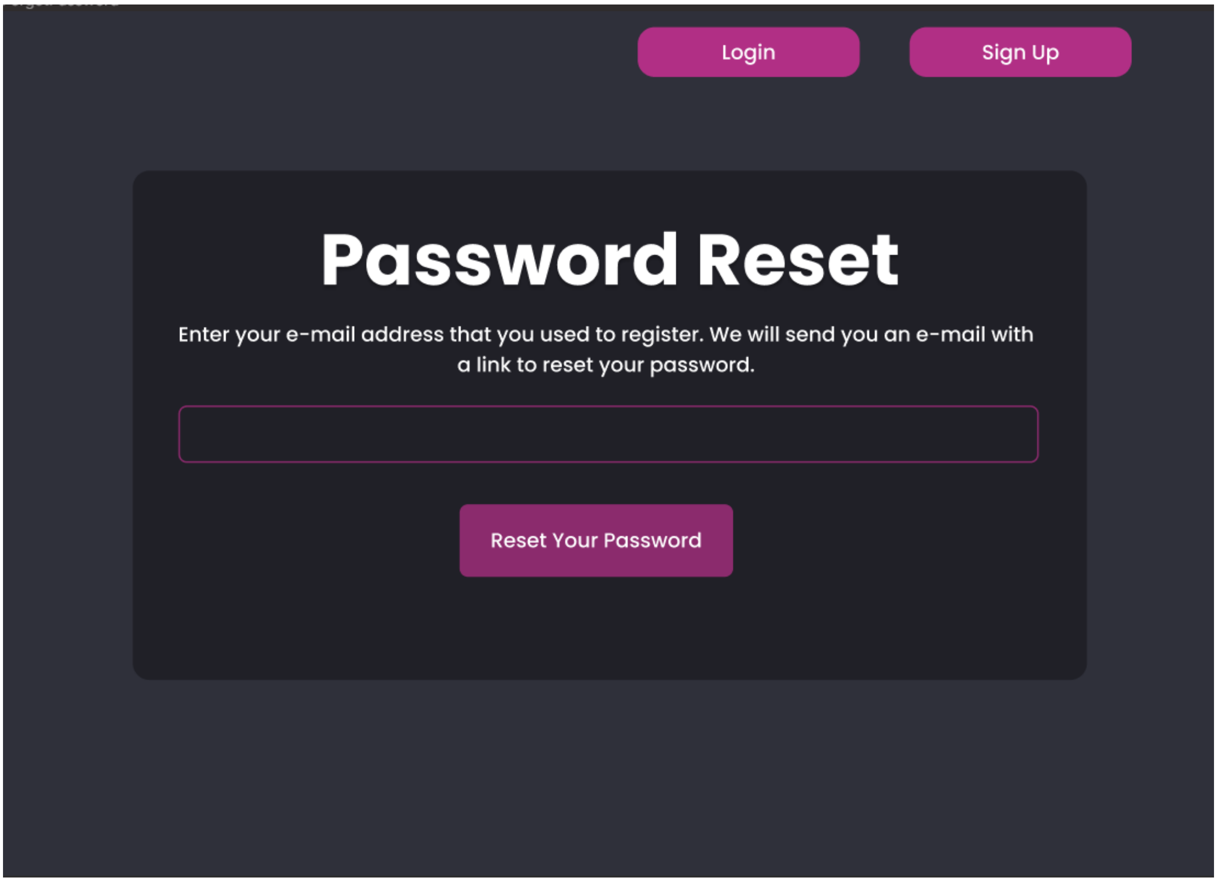


Do not have an account? Sign Up to QUIREC.

Figure 10 The Login Page of QuiRec

Forgot Password Page:

When the user clicks to the “forgot password” button on the login page, a pop-up text area will be displayed on the screen. The user will enter his/her email address in this page and will click the “reset your password” button. And the system automatically sends an email to the user. By clicking on the link sent on the email, the user will be directed to the forgot password page. In this page, the user will change his/her password by entering a new password two times and save by clicking the save button.



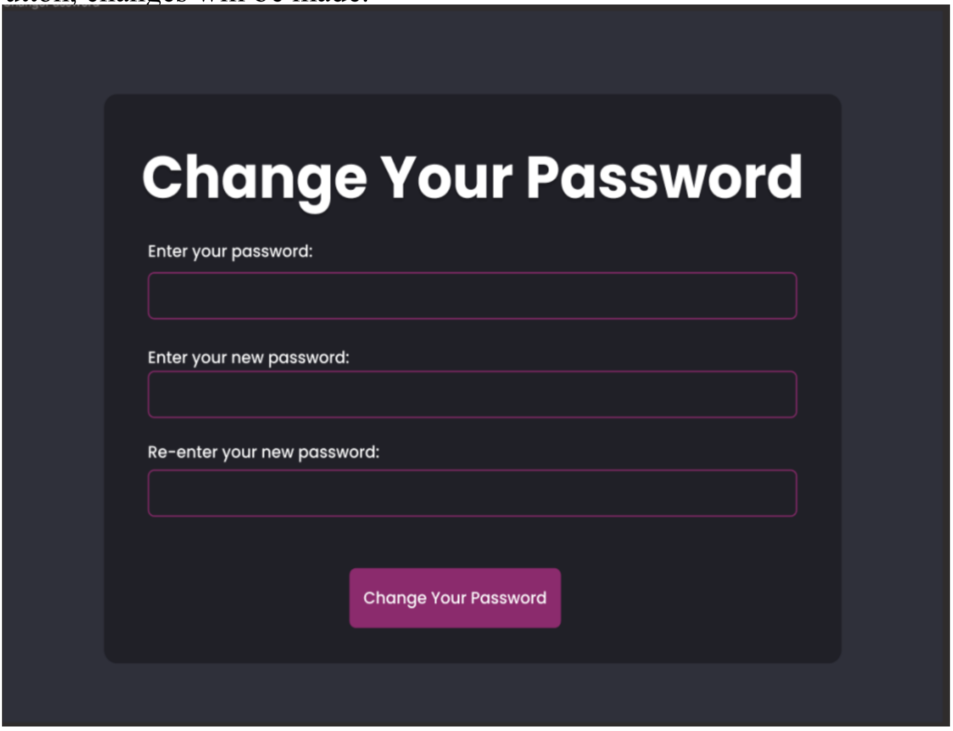
Password Reset

Enter your e-mail address that you used to register. We will send you an e-mail with a link to reset your password.

Figure 11 The Forgot Password Page of The QuiRec

Change Password Page:

This page contains 3 text-boxes. User fills the first box with his/her current password. After that s/he enters his/her new password and re-enters it. When s/he clicks the "Change Your Password" button, changes will be made.



Change Your Password

Enter your password:

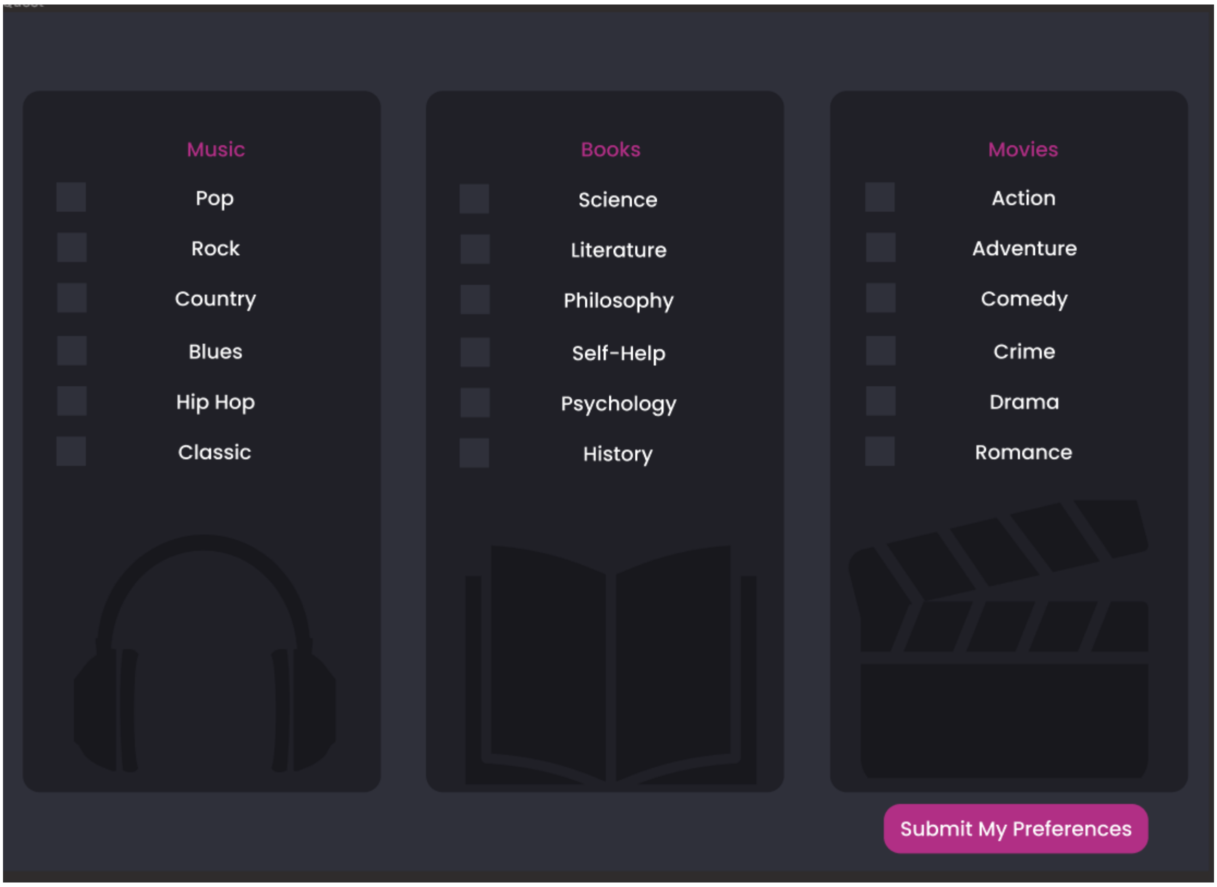
Enter your new password:

Re-enter your new password:

Figure 12 Change Password Page

Questionnaire Page:

This page contains 3 parts with checkboxes and genres. User will check the related boxes to specify his/her preferences. When s/he clicks the ‘Submit My Preferences’ button, database will be updated, and the necessary changes will be saved. User will be directed to Recommendations Page.



Music

□ Pop

□ Rock

□ Country

□ Blues

□ Hip Hop

□ Classic

Books

□ Science

□ Literature

□ Philosophy

□ Self-Help

□ Psychology

□ History

Movies

□ Action

□ Adventure

□ Comedy

□ Crime

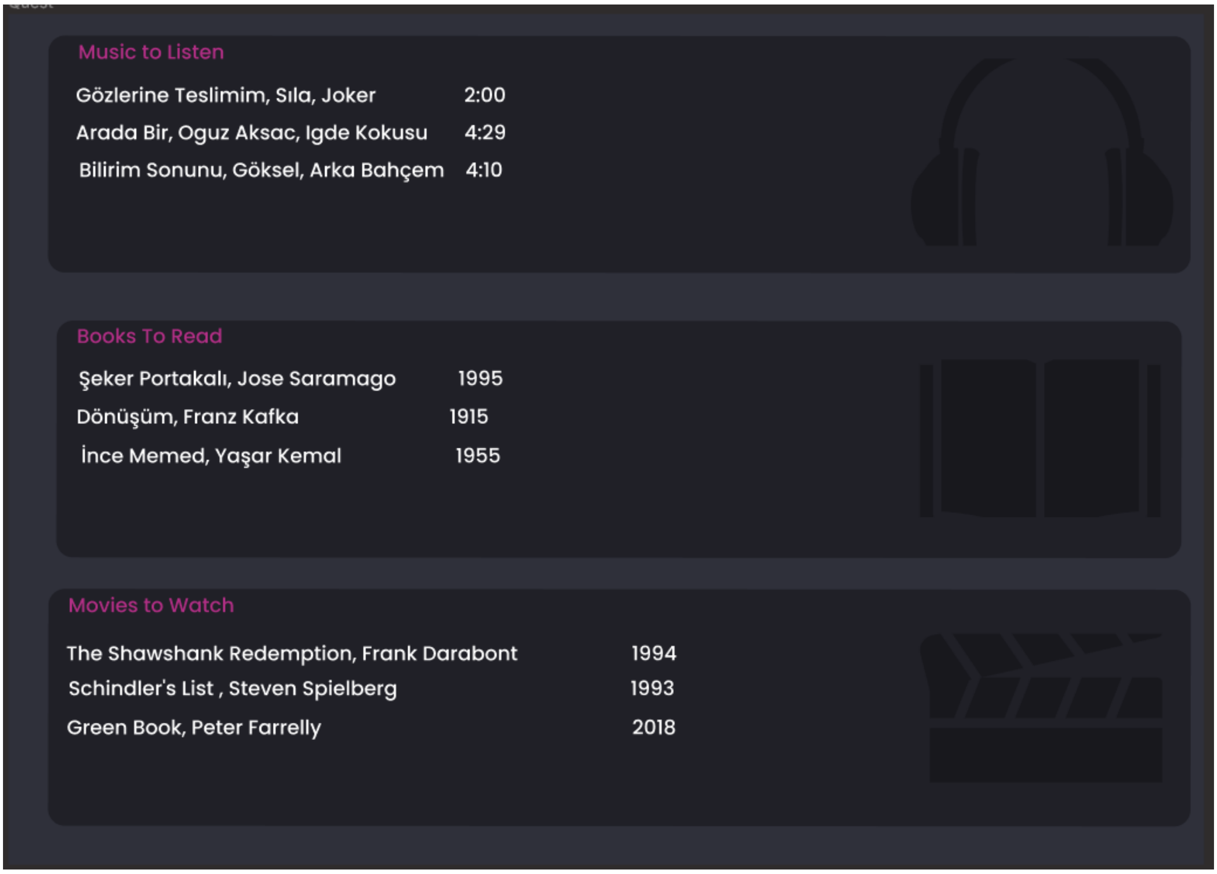
□ Drama

□ Romance

Figure 13 The Questionnaire Page of QuiRec

Recommendations Page:

The recommended music, movies and books are displayed on this page. User will be directed to the details page which contains detailed information of these when s/he clicks on them.



Music to Listen

Gözlerine Teslimim, Sıla, Joker	2:00
Arada Bir, Oguz Aksac, Igde Kokusu	4:29
Bilirim Sonunu, Göksel, Arka Bahçem	4:10

Books To Read

Şeker Portakalı, Jose Saramago	1995
Dönüşüm, Franz Kafka	1915
İnce Memed, Yaşar Kemal	1955

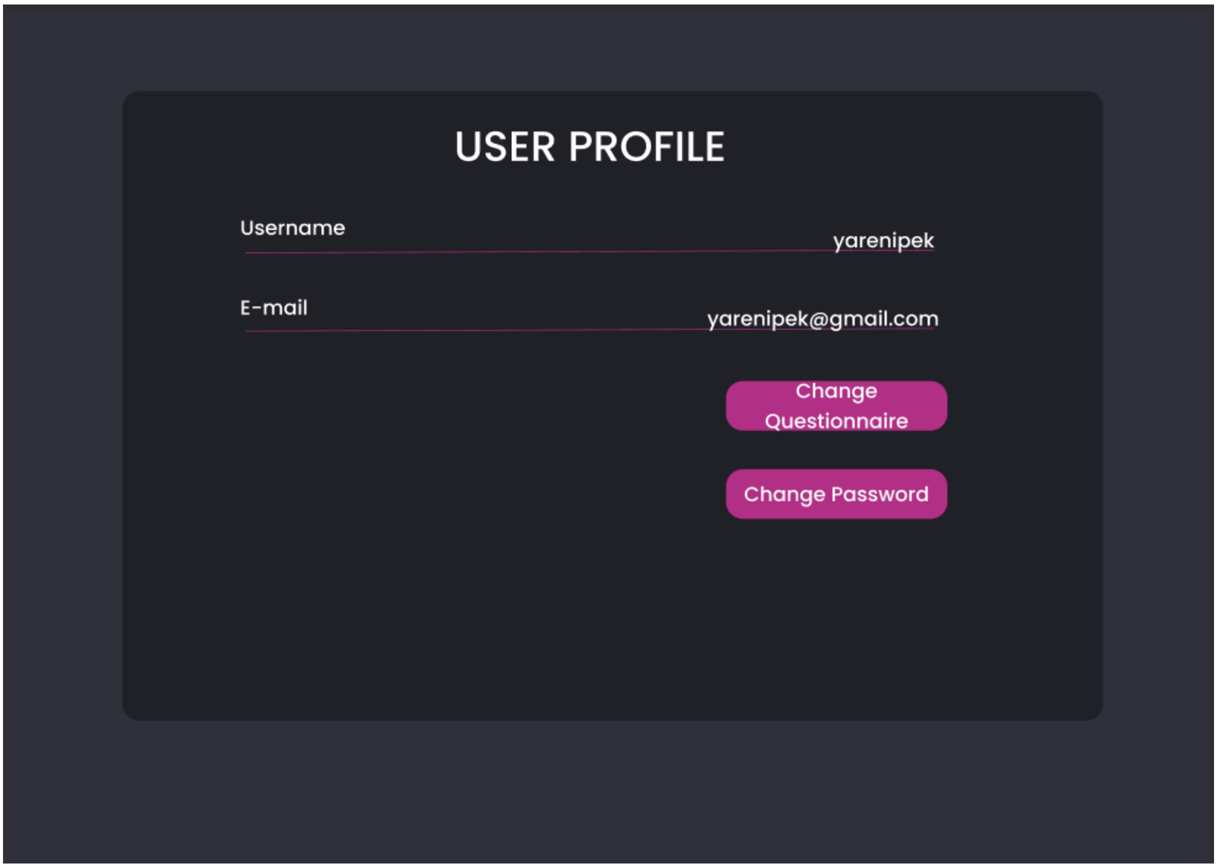
Movies to Watch

The Shawshank Redemption, Frank Darabont	1994
Schindler's List , Steven Spielberg	1993
Green Book, Peter Farrelly	2018

Figure 14 The Recommendations Page of QuiRec

Profile Page:

The user can show his/her personal information and change his/her password or preferences, after these processes s/he will be directed to the recommendations page.



USER PROFILE

Username: yarenipek

E-mail: yarenipek@gmail.com

Figure 15 Profile Page of The QuiRec

Browser Page:

Music, movies and books are displayed on this page. S/he enters a string to the Search Button and the related music, movie or book will be displayed on the screen. S/he will be directed to the details page if s/he clicks on them.

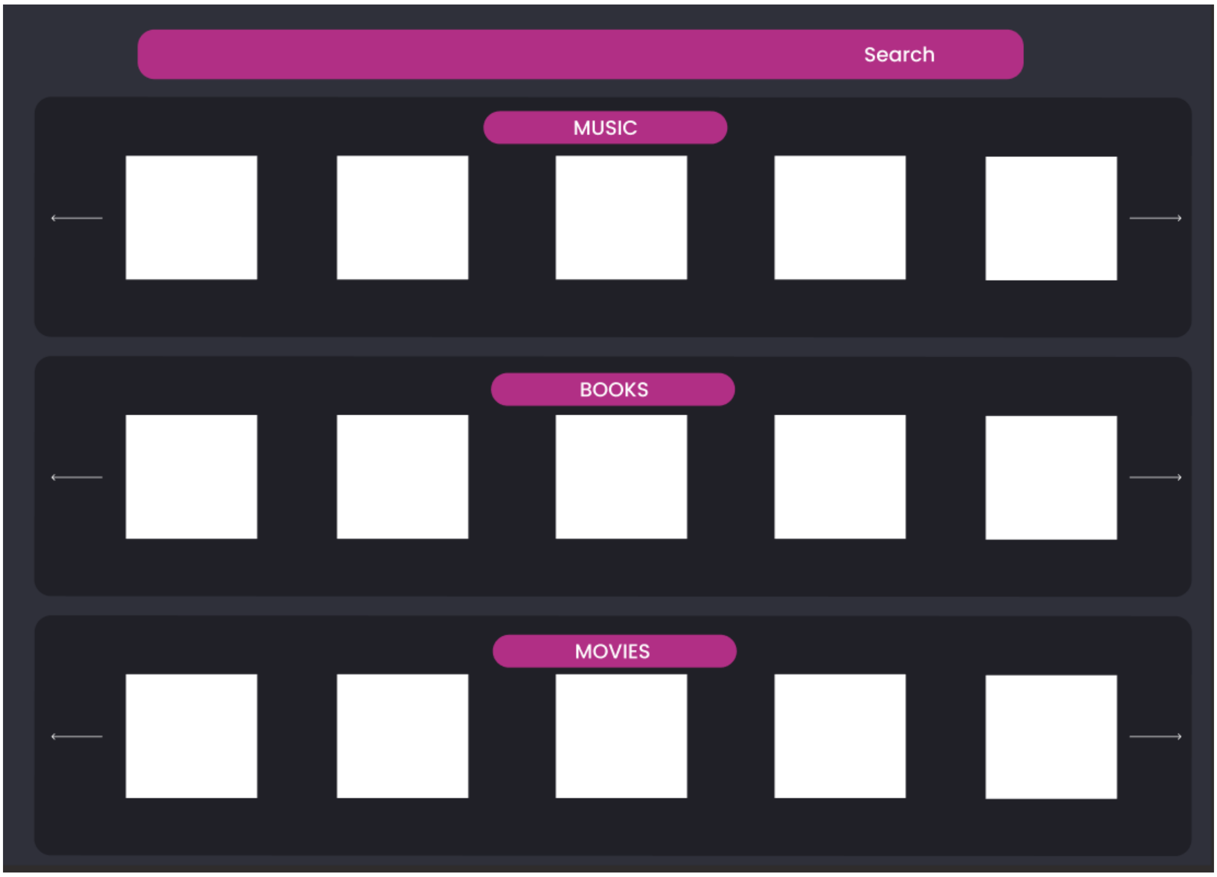


Figure 16 The Browser Page of The QuiRec

Administration Page:

The admin will make the monitoring of the recommendation model's performance. The admin can update the database, retrain the model by clicking the related buttons. S/he can log out from the page.

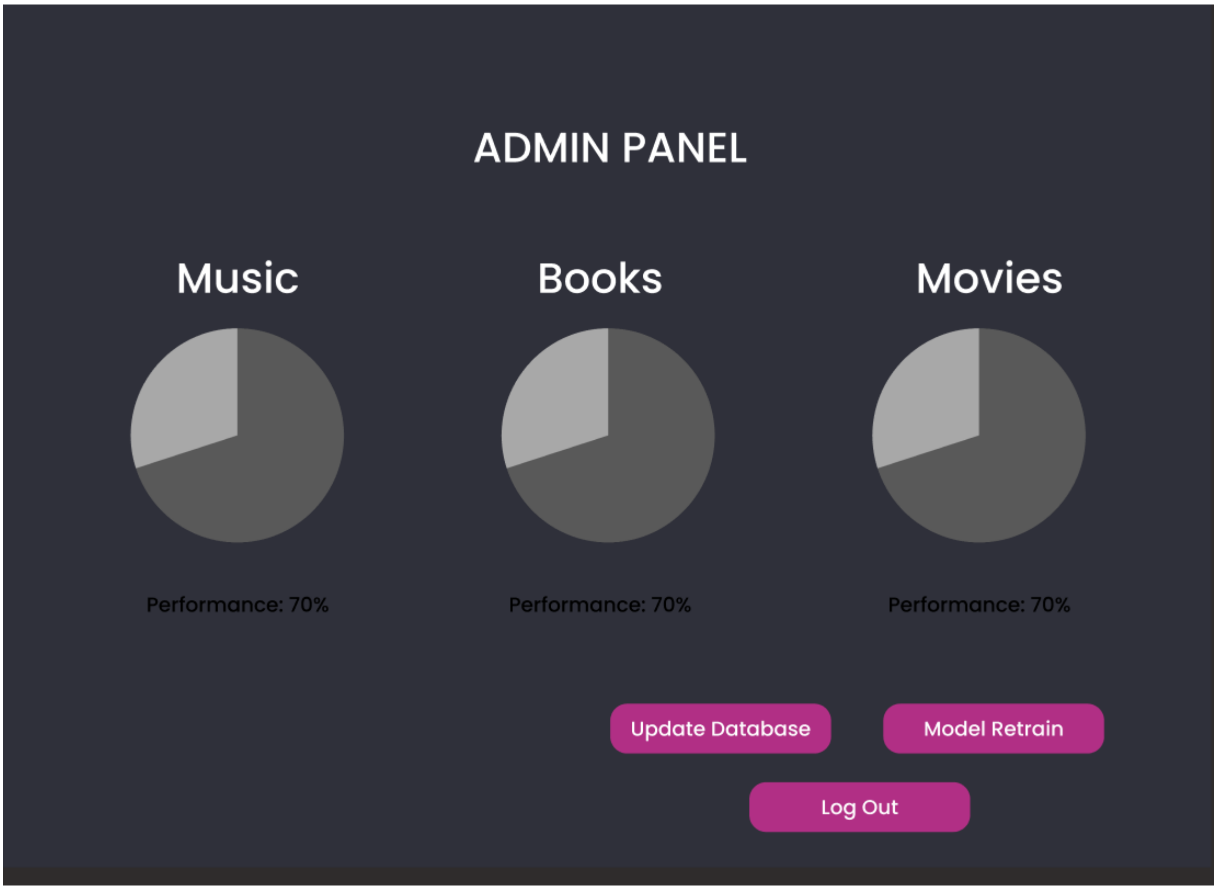
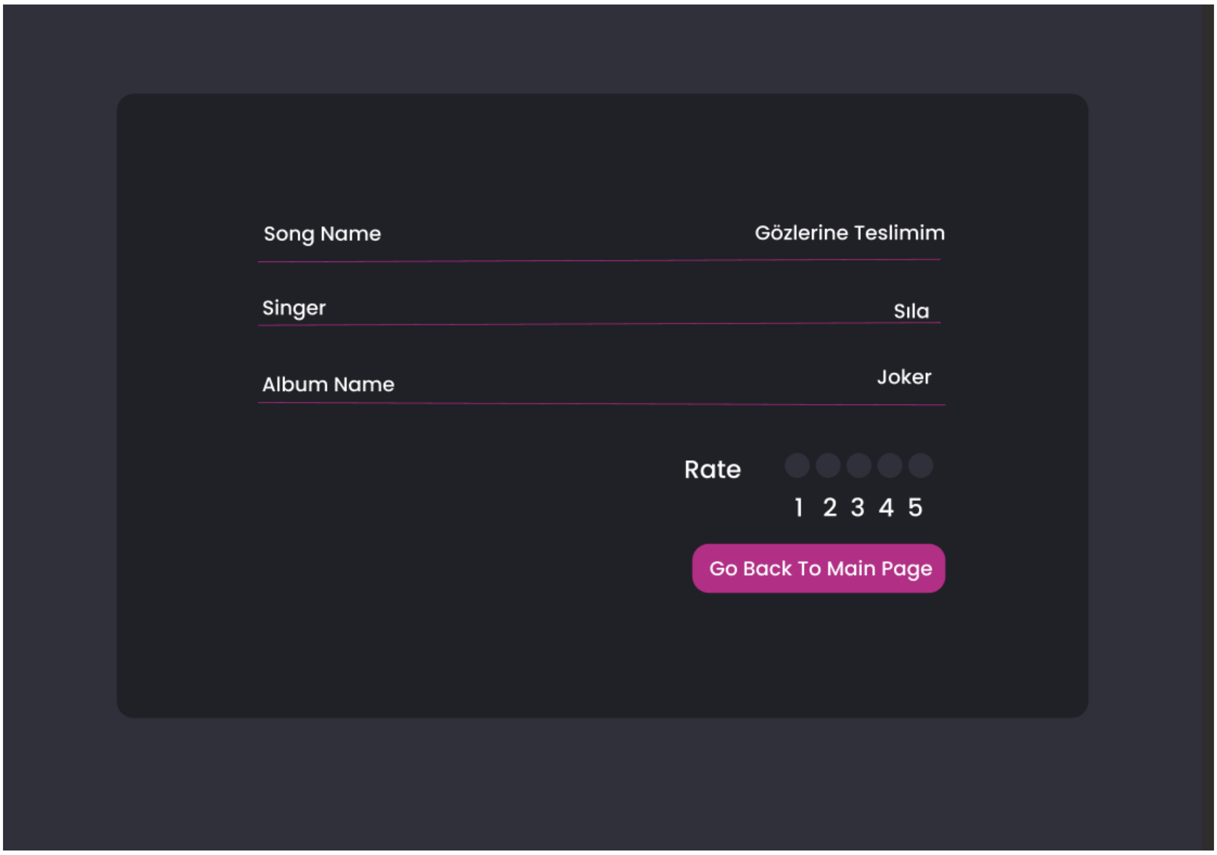


Figure 17 The Administration Page of QuiRec

Details Page:

The user can see the detailed information of the music, movies or books. S/he can rate them from 1 up to 5. S/he will be directed to the main page when s/he clicks on the 'Go Back To Main Page' button.



Song Name Gözlerine Teslimim

Singer Sıla

Album Name Joker

Figure 18 Details Page of The QuiRec

4.3.2. Sequence Diagrams

The user provides necessary information, which is displayed on the screen and required by the server, to register to the system. The interface sends that information to the server. The server makes verification of the information provided by the user using the database. And if required conditions for registration are held, the server inserts the user to the database, and returns an OK message in the verification acknowledgement. Then the user is registered to the system. Else if the required conditions are not held, the server does not save the user to the database and returns an ERROR message in the verification acknowledgement. Then the user is not registered to the system. With return messages, proper messages will be displayed on the screen so that the user understands the response of their requests.

Initially, the admin will be registered to the system automatically by the system developer. So, s/he does not have to use and make requests from the registration page.

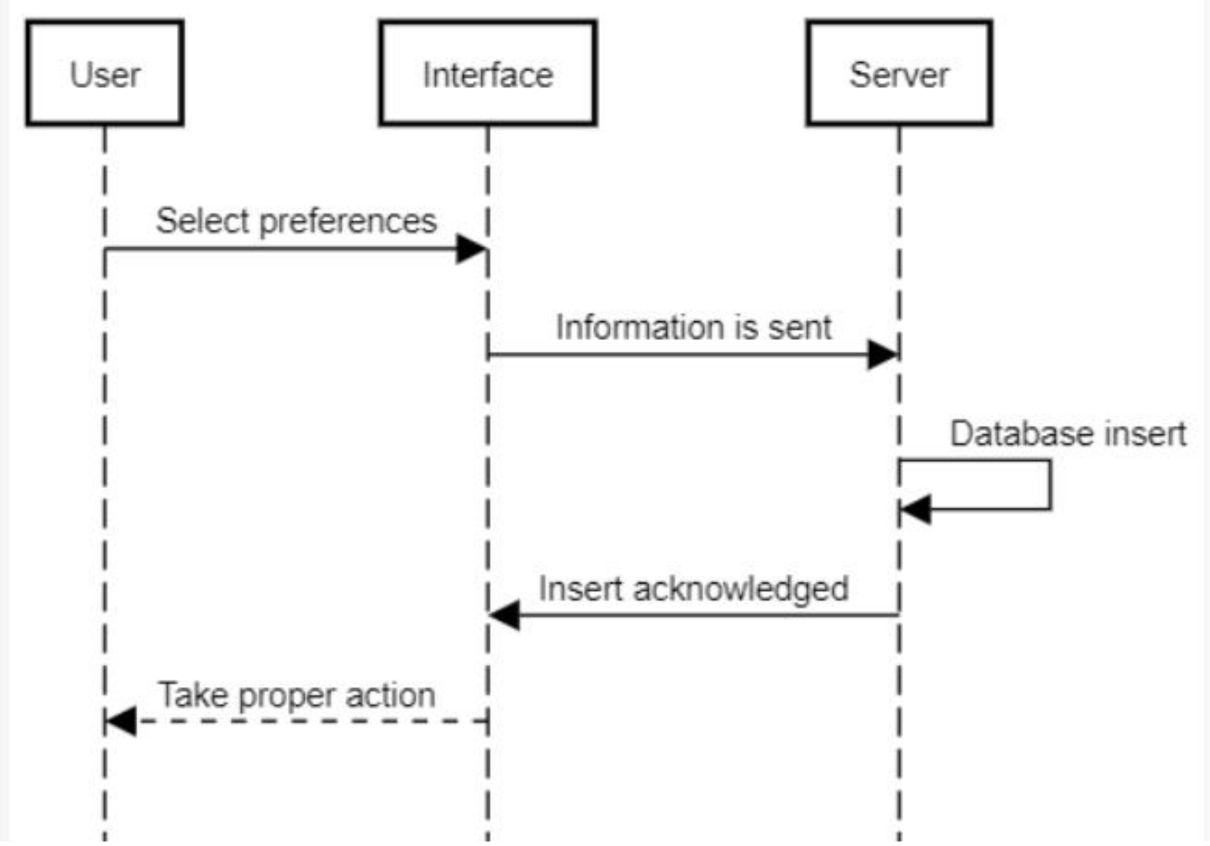


Figure 19 Sequence Diagram of Registration

The user provides necessary information, which is displayed on the screen and required by the server, to login to the system. The interface sends that information to the server. The server makes verification of the information provided by the user using the database. If the user is registered to the system, which means s/he has information already saved in the database, and if the information coming from the interface matches with the information in the database, the server returns an OK message in the verification acknowledgement. Then the user is logged in to the system. Else if the required comparisons are not matched, the server returns an ERROR message in the verification acknowledgement. Then the user is not logged in to the system. With OK return messages, the user is directed to their recommendations page, and with ERROR messages, proper messages will be displayed on the screen so that the user understands that they are not logged in to the system.

The admin also logs to the system with the process and sequence explained above. Only difference between the user and admin is, while users are directed to the recommendations page when the return messages are OK, the admin will be directed to the administration page.

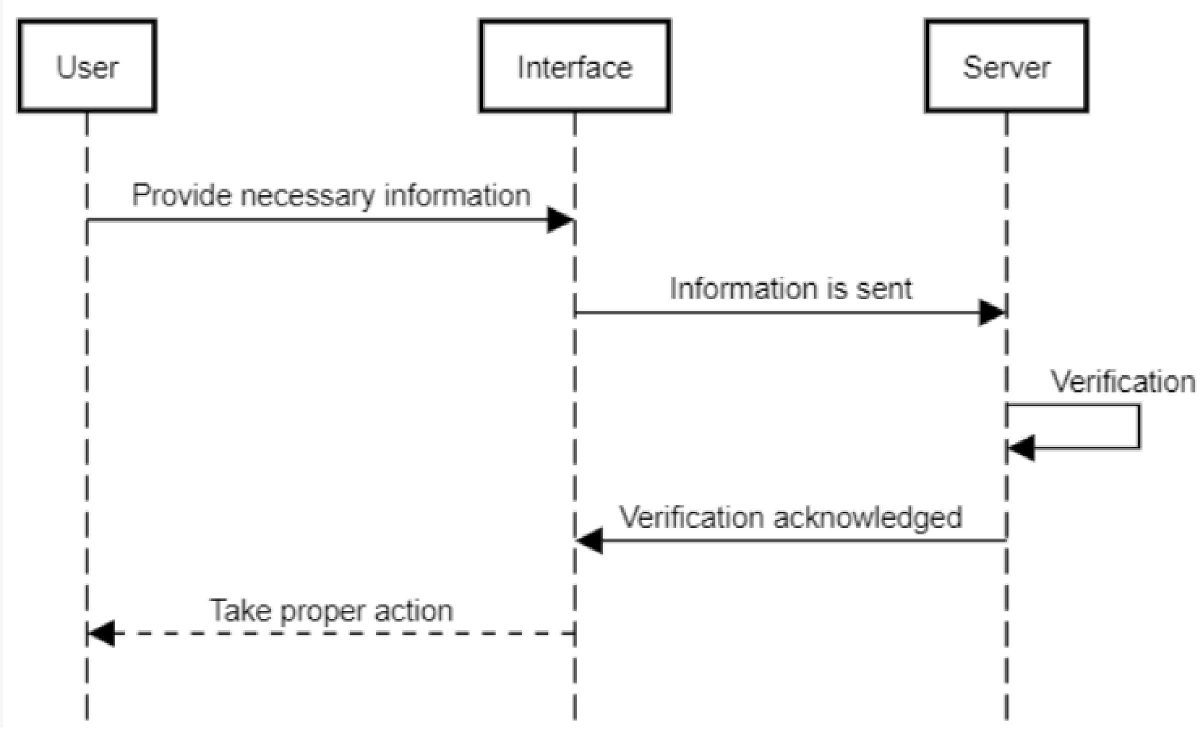


Figure 20 : Sequence Diagram of The Login

The user enters necessary email information, which is displayed on the pop-up screen in the login page and required by the server, to send a mail changing link to the user. The interface sends that information to the server. The server makes verification of the information provided by the user making data querying inside the user’s table. As a result of the query, if a matching data is found, which means the user is registered to the system, the server will send a mail to the user’s address which includes a routing link to change his/her password, and the server returns an OK message in the mail sent acknowledgement. And if no matching data is found, which means the user is not registered to the system, the mail will not be sent, and the server returns an ERROR message in the mail sent acknowledgement. With return messages, proper messages will be displayed on the screen so that the user understands the response of their requests.

If the user receives a mail, they will click to the link inside the mail, and will be directed to the password changing page. The user enters necessary password information, which is displayed on the screen and required by the server, to change his/her password. The interface sends that information to the server. And if required conditions for insert operation are held, such as user verification of user’s operation, the database will be updated with new password information. So, the user’s password will be changed. And if verification is failed, the database will not be updated. Then the result of the request will be sent to the interface in the update acknowledgement. With return messages, proper messages will be displayed on the screen so that the user understands the response of their requests.

Since the admin will be registered to the system automatically by the system developer, the change password operations also will be held by the system developer. So, password change using the interface will not be available for the admin.

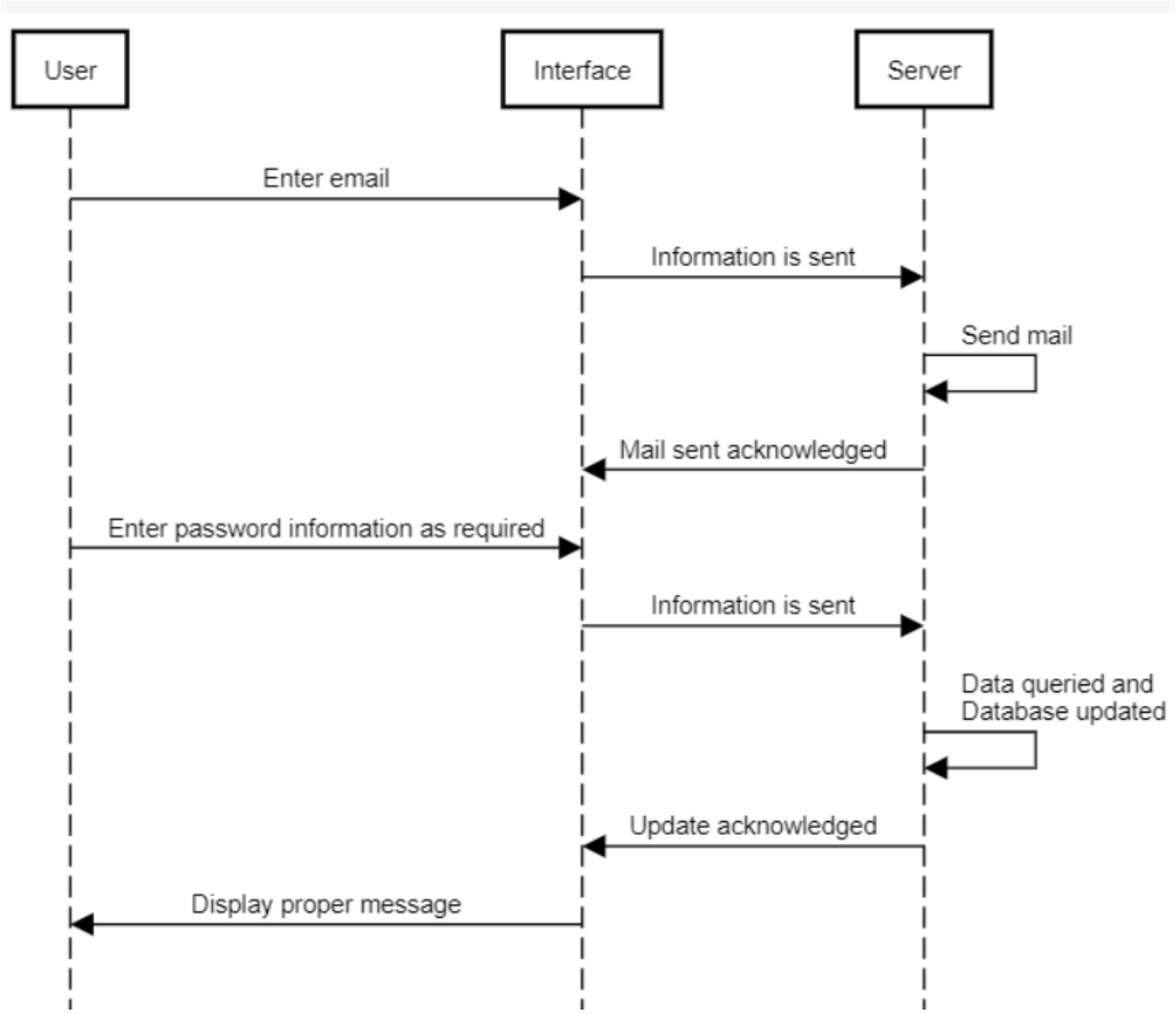


Figure 21 Sequence Diagram of Forgot Password

The user enters necessary password information, which is displayed on the screen and required by the server, to change his/her password. The interface sends that information to the server. The server makes verification of the information provided by the user making data querying inside the user's table. As a result of the query, if a matching data is found, it will be updated with new password information. So, the user's password will be changed. And if no matching data is found, the database will not be updated. Then the result of the request will be sent to the interface in the verification acknowledgement. With return messages, proper messages will be displayed on the screen so that the user understands the response of their requests.

Since the admin will be registered to the system automatically by the system developer, the change password operations also will be held by the system developer. So, password change using the interface will not be available for the admin.

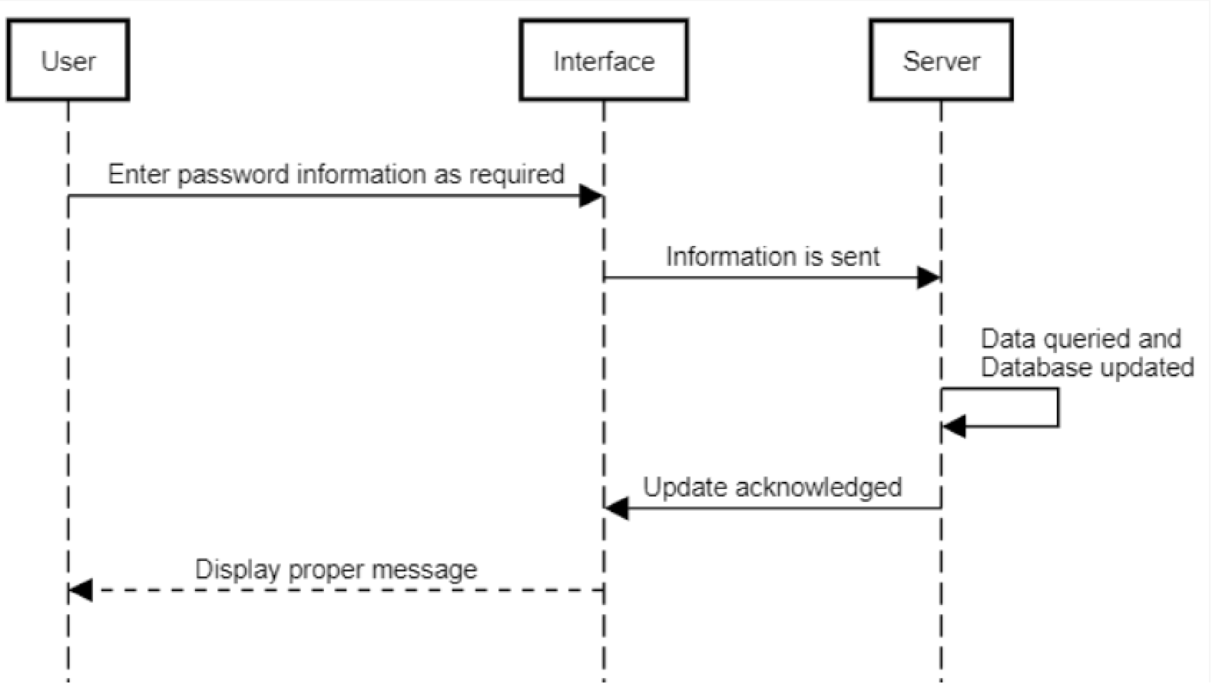


Figure 22 Sequence Diagram of Change Password

When the user opens the recommendations page, the interface sends the user information to the server. The server makes required verifications, such as access token verification, to allow users to see personalized recommendations. Result of the verification will be sent to the interface by the system in the verification acknowledgement. If an error is returned from the server, the information sending step will be skipped by the interface and a proper error message will be displayed. If no error is returned from the server, the interface will send the user information again to the server. With this information, the server will query the database to find a matching record in the database. Then, the result of the query, the server either produces and sends recommendation items to the interface or returns an empty response. If the interface receives an empty response, it will display a proper message on the screen which routes the user to the questionnaire page. Else if the interface receives recommendation items, the user will see these recommendations on the screen.

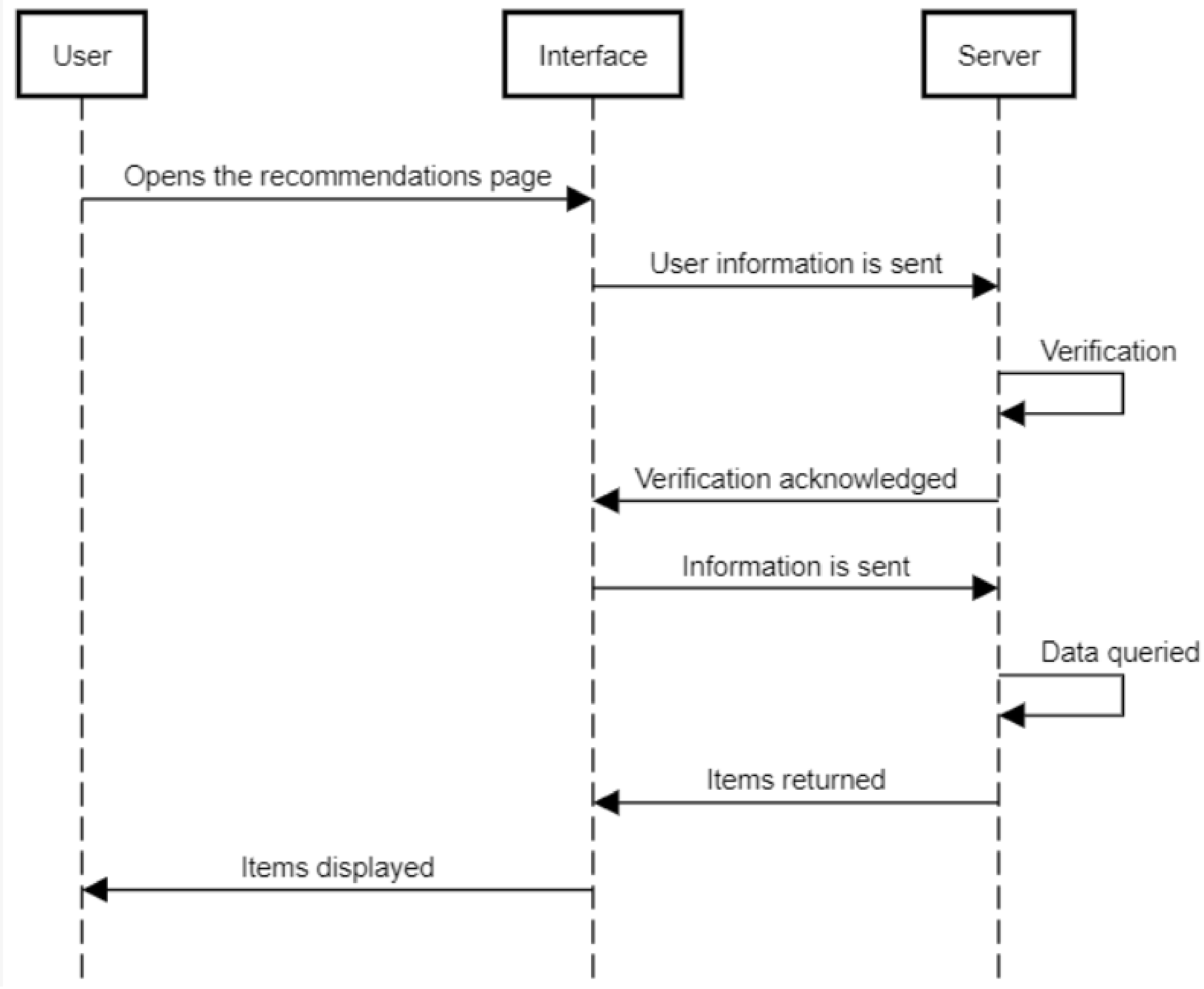


Figure 23 Sequence Diagram of Recommendation

From the details page, the user will see the detailed information about the selected movie, book or music record. To achieve this, the user selects an item from the interface, the interface transfers selected record information to the server, and the server will query the database to find a matching record in the database. Then, the result of the query will be passed to the interface. If any matching is achieved, the user will see the selected item details on the screen, but if no matching is achieved, a proper message for the situation will be displayed.

If the item is displayed properly to the user, the user will see the rating area on the screen. From this area, users will rate the recommendation from 1 to 5, 1 means not related with preferences, 5 means highly related with preferences, by selecting the related rate option. Then, the interface will send the rate information to the server, and the server inserts the rate information in the related database table and returns an OK message in the insert acknowledgement. Then the rating information of the recommendation is saved to the system. Else if any problem occurs during the insertion process, the server does not save the rating information to the database and returns an ERROR message in the insert acknowledgement. With return messages, proper messages will be displayed on the screen so that the user understands the response of their requests.

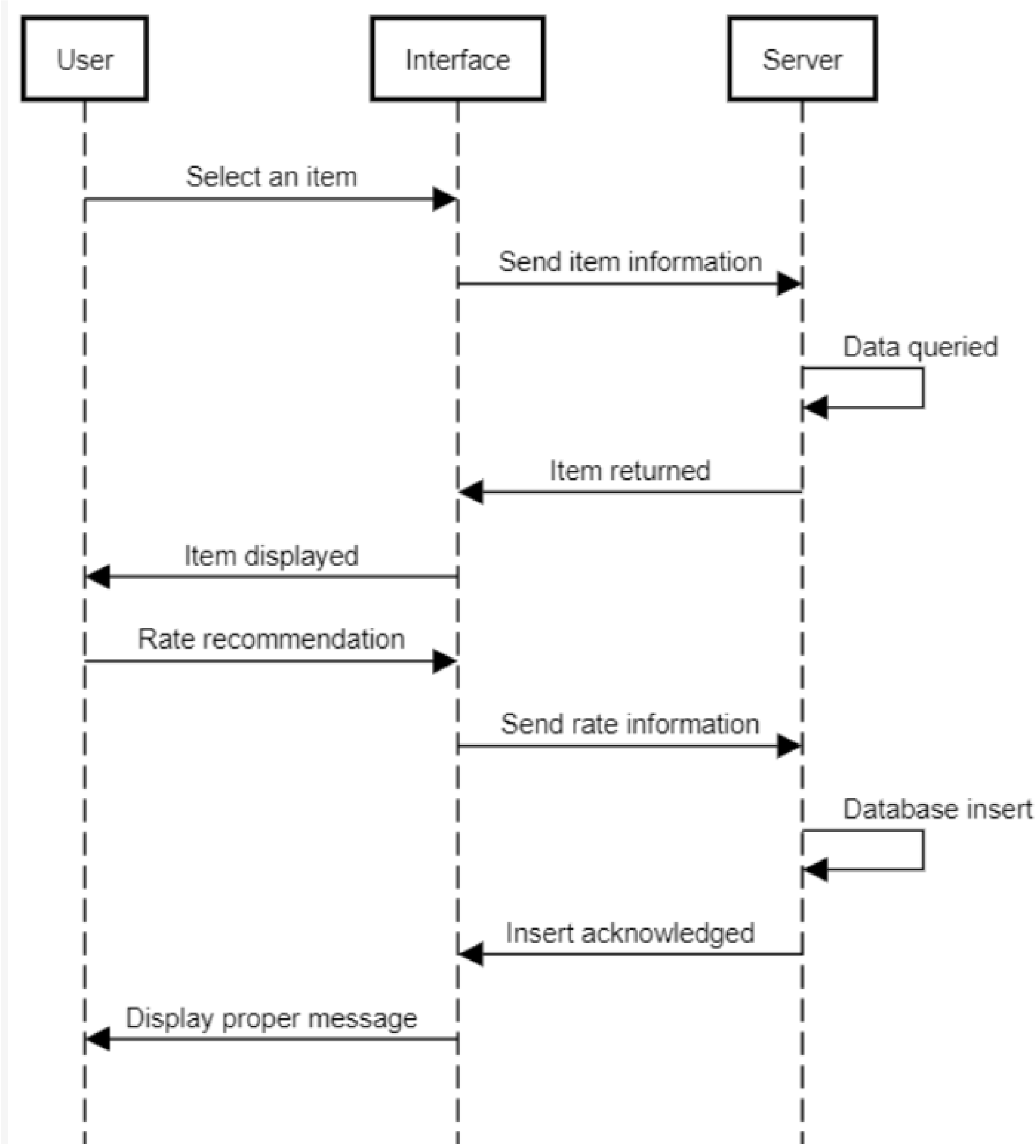


Figure 24 Sequence Diagram of Details

From the questionnaire page, the user will select their preferences according to what they would like to receive recommendations about. And if required conditions for insert operation are held, such as user access token verification, the server inserts the selections to the database, and returns an OK message in the insert acknowledgement. Then the preferences of the user are saved to the system. Else if the required conditions are not held, the server does not save the preferences to the database and returns an ERROR message in the insert acknowledgement. Then the preferences of the user are not saved to the system. With OK return messages, the user is directed to their recommendations page, and with ERROR messages, proper messages will be displayed on the screen so that the user understands that the recommendation preferences are not saved.

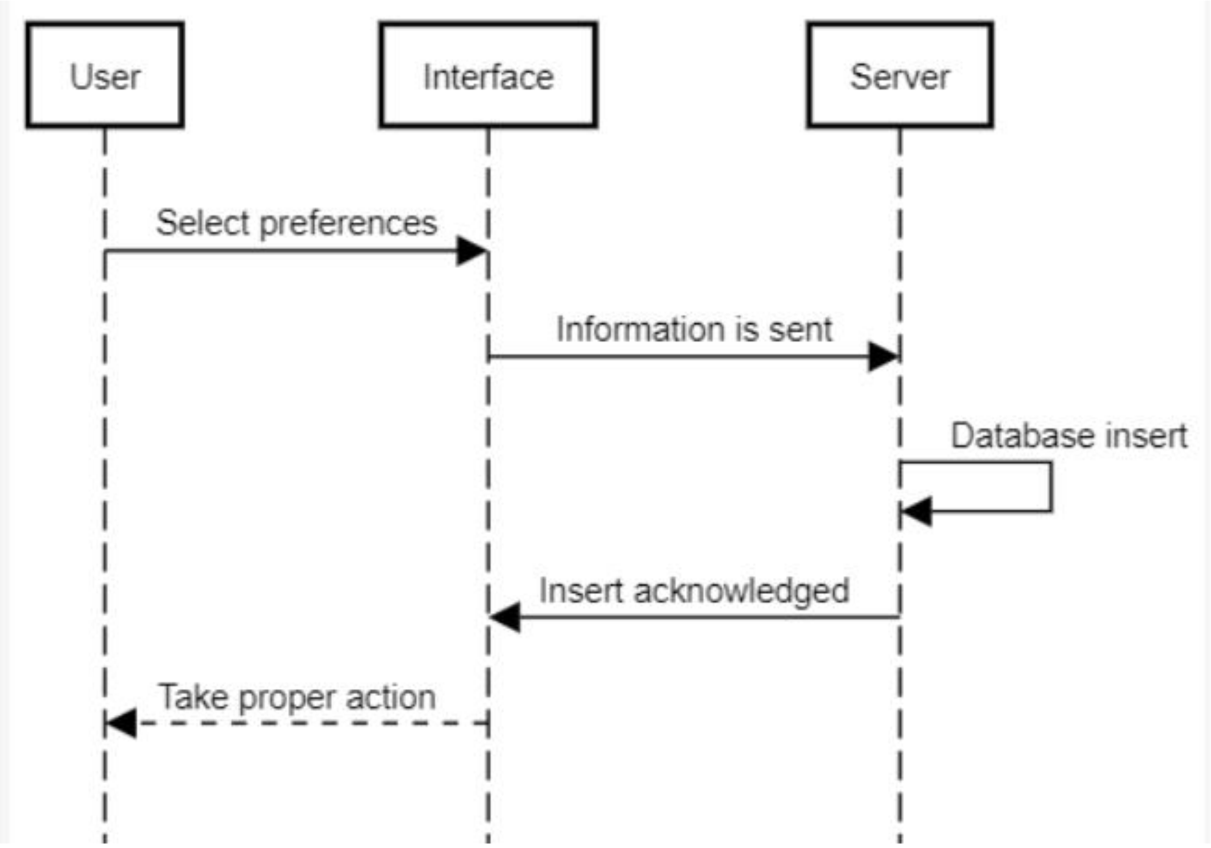


Figure 25 Sequence Diagram of The Questionnaire

From the browser page, the user will see all the movie, book and music records available in the database. To achieve this, the system will proceed as if the keywords are null, and the server will query the database to find all records in the database and return all items to the interface. Then all items will be shown to the user. Also, from this page, the user can search through the records available in the database. To achieve this, the user needs to enter keywords on the search bar according to what they want to search, and the interface transfers those keywords to the server, and the server will query the database to find a matching record in the database. Then, the result of the query will be passed to the interface. If any matching is achieved, the user will see these items on the screen, but if no matching is achieved, a proper message for the situation will be displayed.

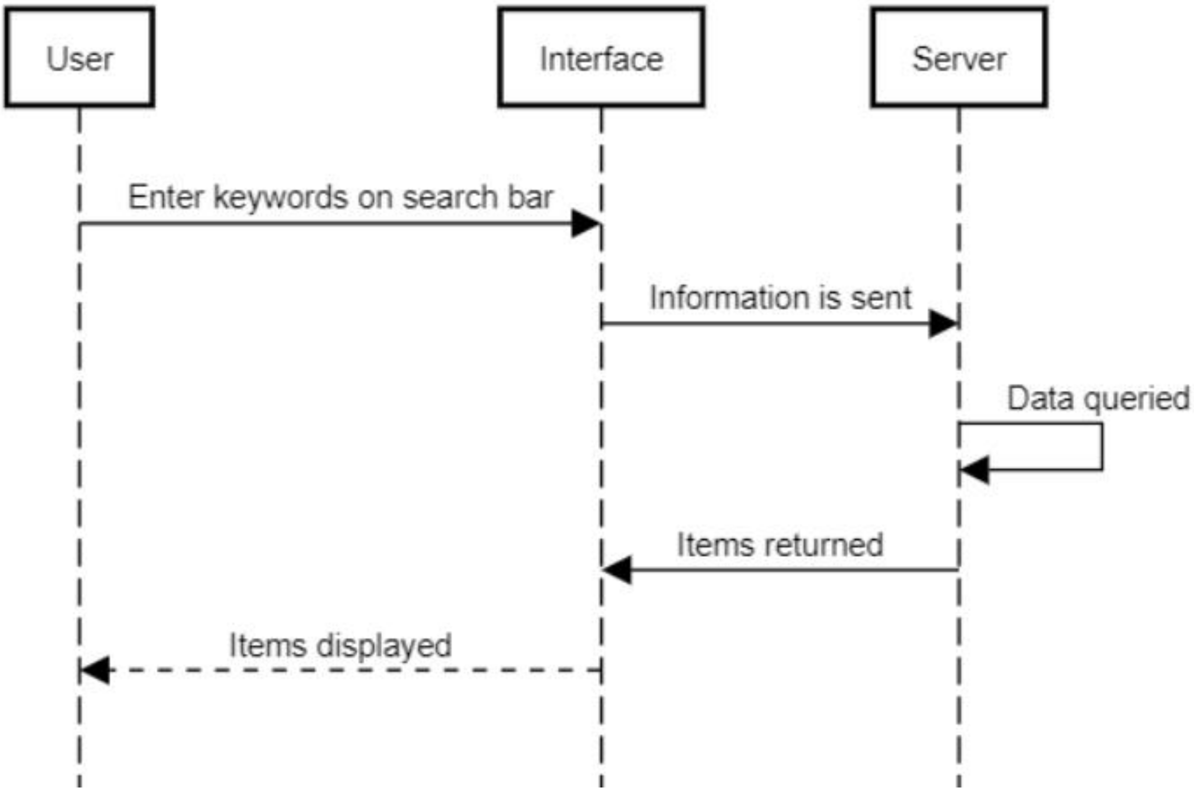


Figure 26 Sequence Diagram of Browser

When the admin opens the administration page, the interface sends the user information to the server. The server makes access token verification, to control if the user is the admin. Result of the verification will be sent to the interface by the system in the verification acknowledgement. If an error is returned from the server, the information sending step will be skipped by the interface and a proper error message will be displayed. If no error is returned from the server, the interface will send the user information again to the server. After this step, the server will query the database to find rating information according to categories. Then, the result of the query, the server either sends rating information to the interface, or returns an empty response. If the interface receives an empty response, it will display a proper message on the screen. Else if the interface receives rating information, the user will see this information on the screen.

When the admin selects the database update option on the screen, the interface sends the user information to the server. Since database update affects the system as a whole, this operation is restricted with the admin. So, another access token verification will be performed by the server to control if the user is admin. Result of the verification will be sent to the interface by the system in the verification acknowledgement. If an error is returned from the server, the database update processes will be skipped by the interface and a proper error message will be displayed. If no error is returned from the server, the server will continue with database update processes. Result of the update operation will be sent to the interface by the system in the update acknowledgement. With return messages, proper messages will be displayed on the screen so that the user understands the response of their requests.

When the admin selects the retrain model option on the screen, the interface sends the user information to the server. Since model retraining affects the system as a whole, this operation is restricted with the admin. So, another access token verification will be performed by the server to control if the user is admin. Result of the verification will be sent to the interface by the system in the verification acknowledgement. If an error is returned from the server, the model retraining processes will be skipped by the interface and a proper error message will be displayed. If no error is returned from the server, the server will continue with model retraining processes. Result of the retraining operation will be sent to the interface by the system in the retraining acknowledgement. With return messages, proper messages will be displayed on the screen so that the user understands the response of their requests.

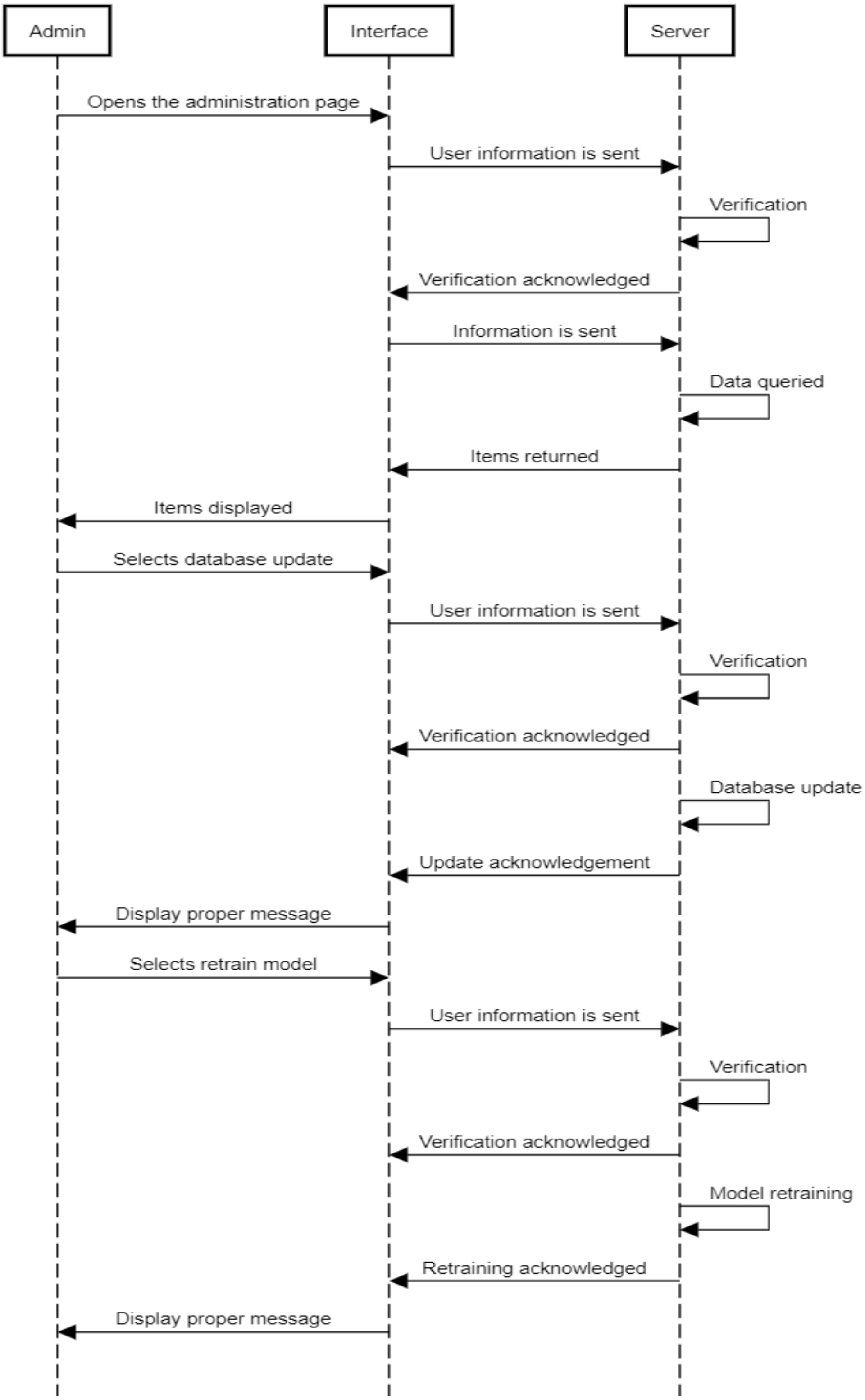


Figure 27 Sequence Diagram of Admin

5.CONCLUSION

To conclude it can be said that the literature review for the system is conducted and according to that research the methods to be used are decided with the exception of the deep learning algorithm. After the literature review is conducted the software requirements specifications document prepared. With this document the information about the systems functionality and performance criteria is explained. There is also an information upon the security of the systems. The last document is the software design description document with includes the information about the architecture of the system as well as the UI design information. This report includes every theoretical aspect of the recommender systems and the next step in the project is to implement such system that covers the theoretical information prepared in this document.

REFERENCES

Ali, A. (2019, May 26). *Restricted Boltzmann Machine (RBM) with Practical Implementation.* Retrieved from Medium: https://medium.com/machine-learning-researcher/boltzmann-machine-c2ce76d94da5

Amazon. (n.d.). *What is Mobile Application Developmeny*. Retrieved from AWS: https://aws.amazon.com/tr/mobile/mobile-application-development/

Aparna. (2020, October 2020). *Which Approach Is Best For Your Project: Waterfall Vs Agile Methodology?* (Mobile App Daily) Retrieved from Mobile App Daily: https://www.mobileappdaily.com/waterfall-and-agile-methodology-comparison

Balaji, s., & Murugaiyan, S. (2012). WATEERFALLVs V-MODEL Vs AGILE: A COMPARATIVE STUDY ON SDLC . *International Journal of Information Technology and Business Management*, 26-30.

Banga, S. (2020, June 16). *What is Web Application Architecture? Components, Models and Types*. Retrieved from Hackr.io: https://hackr.io/blog/web-application-architecture-definition-models-types-and-more

Bean, J. (2009). *SOA and Web Services Interface Design: Principles, Techniques and Standarts.* Morgan Kaufmann.

Hidasi, B., Baltrunas, L., Karatzoglu, A., & Tikk, D. (2015). SESSION-BASED RECOMMENDATIONS WITH RECURRENT NEURAL NETWORKS. *ICLR 2016.*

IBM. (2020, July 16). *SOAP*. Retrieved from IBM: https://www.ibm.com/support/knowledgecenter/SSGMCP_5.2.0/com.ibm.cics.ts.webservices.doc/concepts/soap/dfhws_overview.html

IBM. (2020, July 16). *What is a Web Service?* Retrieved from IBM: https://www.ibm.com/support/knowledgecenter/SSGMCP_5.2.0/com.ibm.cics.ts.webservices.doc/concepts/dfhws_definition.html

IBM. (n.d.). *SOAP Endpoint*. Retrieved from IBM: https://www.ibm.com/support/knowledgecenter/SSSHYH_7.1.0/com.ibm.netcoolimpact.doc/dsa/imdsa_web_netcool_impact_soap_endpoint_c.html

Kidecha, S. (2020, May 27). *Native vs. Hybrid vs. Cross-Platform: How and What to Choose?* Retrieved from DZone: https://dzone.com/articles/native-vs-hybrid-vs-cross-platform-how-and-what-to#:~:text=Hybrid%20has%20limited%20access%20to,match%20the%20Native%20app%20development

Latif, M., Lakhrissi, Y., Nfaoui, E., & Es-Sbai, N. (2016). Cross platform approach for mobile application development: A survey. *2016 International Conference on Information Technology for Organizations Development (IT4OD)* (pp. 1-5). Fez: IEEE.

Le, J. (2019, October 31). *Recommendation System Series Part 2: The 10 Categories of Deep Recommendation Systems That Academic Researchers Should Pay Attention To.* Retrieved from Towards Data Science:

https://towardsdatascience.com/recommendation-system-series-part-2-the-10-categories-of-deep-recommendation-systems-that-189d60287b58

Lops, P., de Gemmis, M., & Semeraro, G. (2010). Content-based Recommender Systems: State of the Art and Trends. In *Recommender Systems Handbook* (pp. 73-105). Boston: Springer.

Mahmoud, Q. H. (2005, April). *Service-Oriented Architecture (SOA) and Web Services: The Road to Enterprise Application Integration (EAI)*. Retrieved from Oracle: https://www.oracle.com/technical-resources/articles/javase/soa.html

O'Hear, S. (2013, July 9). *Soundwave, The Music Discovery App Backed By Mark Cuban, Is Placing A Big Bet On Big Data.* Retrieved from Techcrunch: https://techcrunch.com/2013/07/09/soundwave/

Red Hat. (n.d.). *What is an API*. Retrieved from Red Hat: https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces

Rocca, B. (2019, june 3). *Introduction to Recommender Systems.* Retrieved from Towards Data Science: https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada

Strickland, J. (2009, May 19). *How Goodreads Works.* Retrieved from How Stuff Works: https://computer.howstuffworks.com/internet/social-networking/networks/goodreads.htm

Summerfield, J. (-). *Mobile Website vs. Mobile App: Which is Best for Your Organization?* Retrieved from Human Service Solutions: https://www.hswsolutions.com/services/mobile-web-development/mobile-website-vs-apps/

Varolgüneş, E., Sarı, E. E., İpek, Y., & Serimer, P. (2020, October). *Literature Review for a Platform for Music, Book and Movie Recommendation.* Github: https://github.com/CankayaUniversity/ceng-407-408-2020-2021-A-Platform-for-Music-Book-and-Movie-Recommendation/blob/main/LiteratureReview_Group07.pdf adresinden alındı

Wang, H., Wang, N., & Yeung, D.-Y. (2015). Collaborativa Deep Learning for Recommender Systems. *KDD '15: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1235-1244). New York,NY: Association for Computing Machinery.

Ying , R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., & Laskovec, J. (2018). Graph Convolutional Neural Networks for Web-Scale Recommender Systems. *KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 974-983). London: Association for Computing Machinery.