ÇANKAYA UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

Project Report

Version 1

CENG 408

Innovative System Design and Development II

202105
Game Recommendation System using Machine Learning Algorithms

Tolga Bolat
201711008
Kutay Kabadaş
201711039
Berfin Varlı
201711068

Dr. Serdar ARSLAN

İçindekiler

Abstract

The game industry evolved so much that there are 10.000 games being released every year. With this, game users can’t even decide what’s in their taste. It gives customers a very overwhelming and lost feeling in these large, detailed choices of products. A solution to this relies on building such systems that search desired but not yet discovered games. Thus, the hunger in the market led to these kinds of programs. Especially in recent years, we are using recommendation systems without our knowledge in many places that we don't even realize anymore. These systems are in a very important place, without them we would be lost. The aim of this project is to develop a system that can give game recommendations to people who are looking for games they may like based on the games they have liked before or the users that have similar history with the current user. The game is recommended to the current user based on the games that other users like. Within the scope of this project, we are conducting extensive research in machine learning, which is the application of artificial intelligence, in order to work more efficiently.

Öz

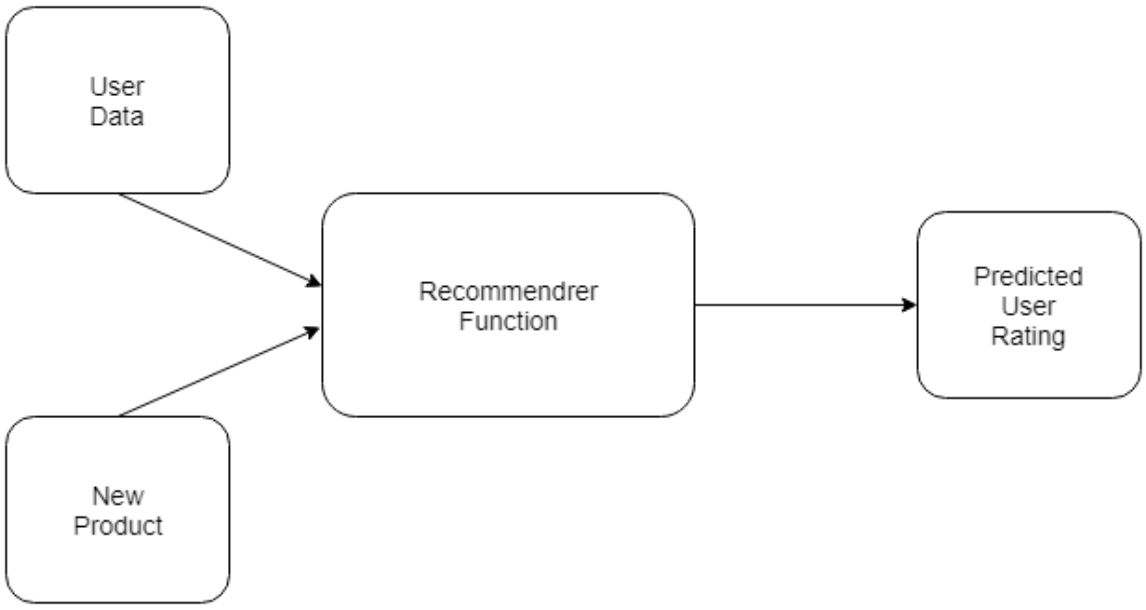
Oyun endüstrisi o kadar çok gelişti ki yılda 10.000’den fazla oyun çıkıyor. Bununla beraber kullanıcılar artık kendi zevklerini bile karar veremiyor. Bu geniş, ayrıntılı ürün seçeneklerinde müşteriler adeta kayboluyor. Buna bir çözüm olarak, istenen ancak henüz keşfedilmemiş oyunları arayan sistemleri oluşturmaya dayanır. Bu yüzden piyasadaki açlık bu tarz programlara sebep olmuştur. Günümüzde önerici sistemleri farkında olmadan hayatımızın her yerinde kullanıyoruz. Bu sistemler o kadar önemli bir noktadaki, onlar olmadan kayboluruz. Bu projenin amacı, daha önce beğendiği oyunlardan veya mevcut kullanıcıyla benzer bir geçmişi olan kullanıcılardan yola çıkarak beğenebileceği oyunları arayan kişilere oyun tavsiyeleri verebilecek bir sistem geliştirmektir. Kullanıcıya, diğer kullanıcıların ilgi duyduğu ve sevdiği oyunlara göre bir oyun önerilir. Bu proje kapsamında daha verimli çalışabilmek için yapay zeka uygulaması olan makine öğrenmesi konusunda kapsamlı araştırmalar yapıyoruz.

1. Literature Review

1.1 Introduction

Recommendation systems are algorithms that aim to suggest relevant items to users (games, movies, text to read, products to buy, etc.).[1] Recommender systems are a crucial feature in our world, as users are often overwhelmed by choice and need help finding what they're looking for. Recommender systems use related items user chose and other users history to give a good recommendation. In this case, it is going to

find many game users might like, by checking users with similar interests and tastes. Our goal is to make it as perfect as it can be. Recommender systems are one of the most used applications of machine learning technologies. Machine learning uses both user data and item data to build a sample dataset. By using this dataset, we train the methods we use to make a prediction. Machine learning algorithms in recommender systems are classified into three categories - content-based and collaborative filtering methods, but modern recommenders combine both approaches and this is called hybrid.



[2]

Content-based Filtering

Content-based approaches use additional information about users such as age, gender, location, etc. When a game is chosen by the user, with information such as age and gender, we can recommend another user with similar information, the same game. Another model is using keywords of the game, such as its title, tags, description, etc. We can make many models using the informations we have about users. With these models, we can make new suggestions for a user. In content-based filtering, recommendations are specific to each user. It is therefore scalable to a large user base. Such recommendations are often found under the "Similar products" or "Recommended items" tags. [3] As a result many websites ask you to indicate your date of birth, gender etc when registering, this is because you need to provide more data for their system to make better predictions. Limitation: The recommendation will be limited to what users liked, watched, interacted with before. It doesn't give users a chance to explore a new area they've never been to before. Also, all users who like item X will receive the same recommendation set. [4]

Collaborative Filtering

Collaborative filtering is a technique used by recommendation systems. Collaborative filtering is a method of making predictions about the user's interests using the preferences made. For example, if users A and B have played a game and liked it, the system will also recommend the game that user A has played to user B.[5] Thus, the input to a collaborative filtering system will be all historical data of user interactions with target items. This data is typically stored in a matrix where the rows are the users, and the columns are the items.[6] There are 2 types of Collaborative Filtering Systems :

User-Based Collaborative Filtering:

This technique is personalized, special to every user. User-Based Collaborative Filtering is a technique used to predict games a user might like based on whether they are liked or disliked by other users of similar taste. [7]

Item Based Collaborative Filtering:

Item-item collaborative filtering is a type of recommendation method that suggests a similar product that the user previously liked. It was developed by Amazon in 1998 and plays a huge role in Amazon's success. [8]




[6]

Since we have to make user-item interaction, we can describe it in 2 way. Explicit and implicit way.

- With explicit rating, we ask user to rate items based on their likings. We can understand users satisfaction directly for a specific item. For example we can ask user to rate a game, ranging from 1-5 scale, user's score would give us the data we can use on recommending other items to the same user as well as other users.
- The implicit rating is gathering the data from the user indirectly from users behaviour. For example a user can play a game for 10 minutes or 100 hours. We can see how much the user liked the product indirectly.

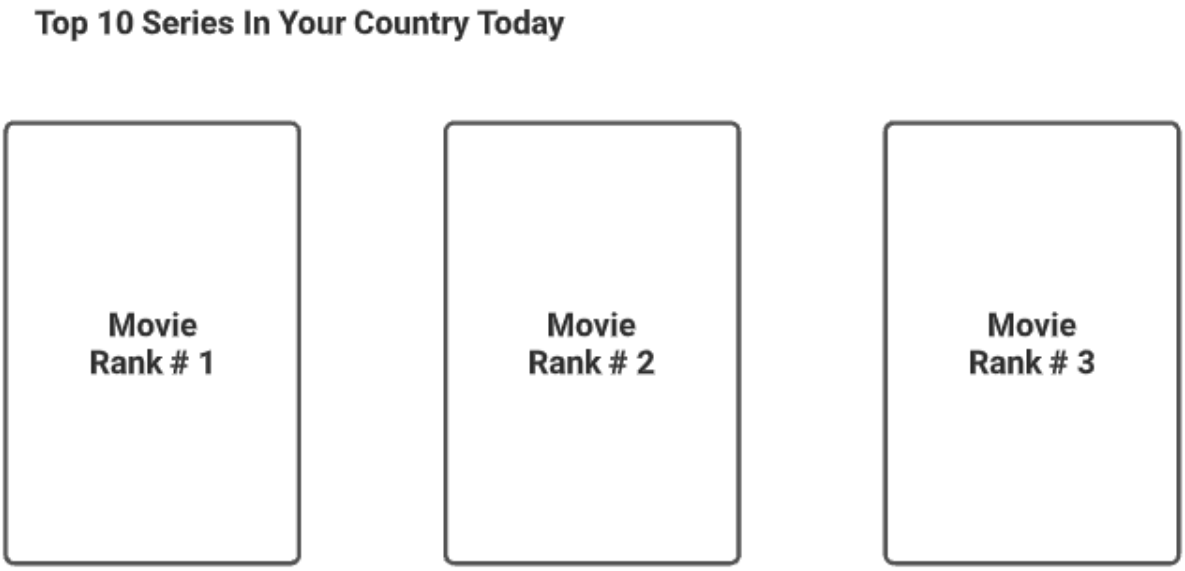
Hybrid Recommendation Systems

Hybrid approaches can be enforced in several ways, for illustration by making content-based filtering and collaborative filtering independently and combining

them, or by combining approaches into a single model, etc.[9] Almost all modern recommendation system implementations are hybrid.

Popularity Based Recommendation Systems

This type of recommendation system is using popularity as ranking what to recommend to users. It helps users to find what other combined users liked. It is a popular and trending type of working recommendation system. These systems recommend the most popular product, game, music, movie among users.



[4]

Merits of Popularity Based Recommendation System

The user's historical data is not required. If the user is newly registered, Trends can help new users to recommend a product without any user activity occurred.

Demerits of Popularity Based Recommendation System

It is not personalized. The system recommends the same type of products to each user based only on popularity.

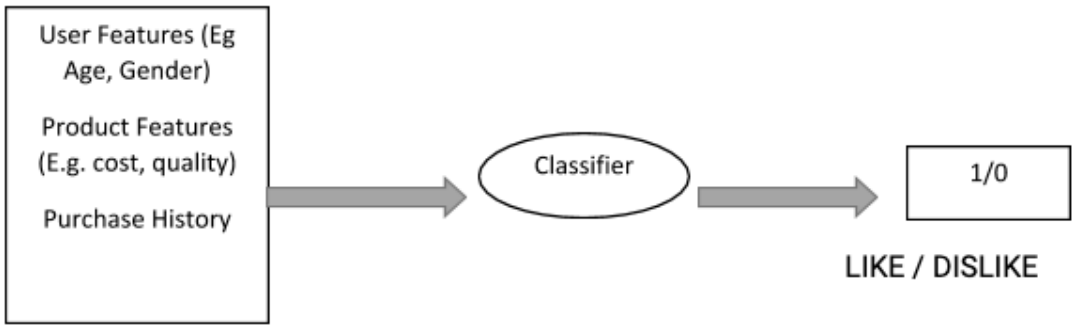
Example:

Google News: News filtered by trending and most popular news.

YouTube: Trending videos.

Classification Model

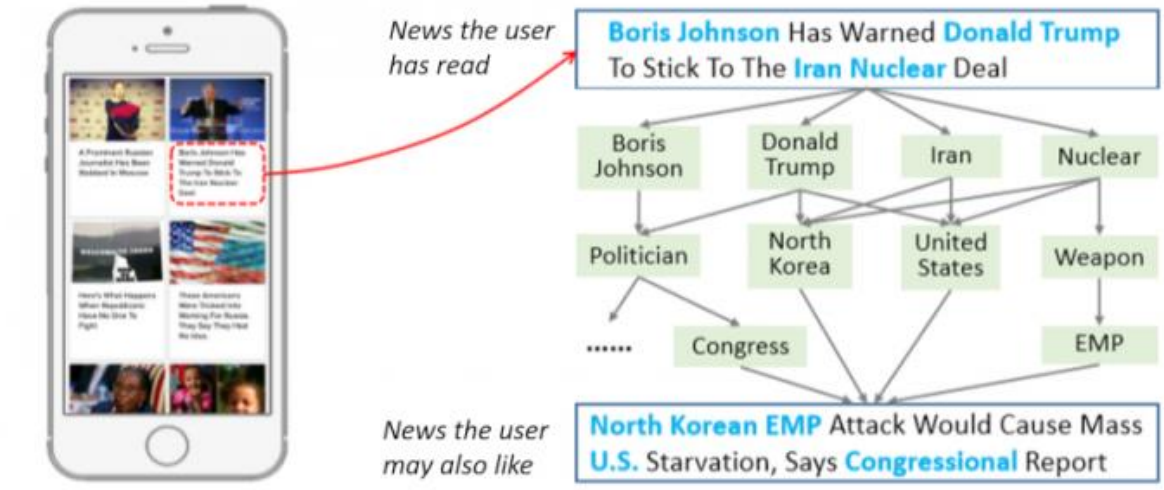
It predicts whether the user will like the product or not. To do this, it uses both product attributes and the user's information. If the user likes the product, the output will be 1 and if not, the output will be 0. [10]



[10]

Knowledge-based Recommender Systems

This kind of recommendation system with machine learning extracts a company's domain knowledge that is governed by 'if-this-then-that' rules. The USP of a knowledge-based recommendation system is that it can be constantly improved not by the user's history but its interaction with the system. This can happen through the underlying 'critique method' that enables users to assign feedback to recommendations for improving search results.



[3]

1.2 Similar Works

Netflix’s Recommendation System

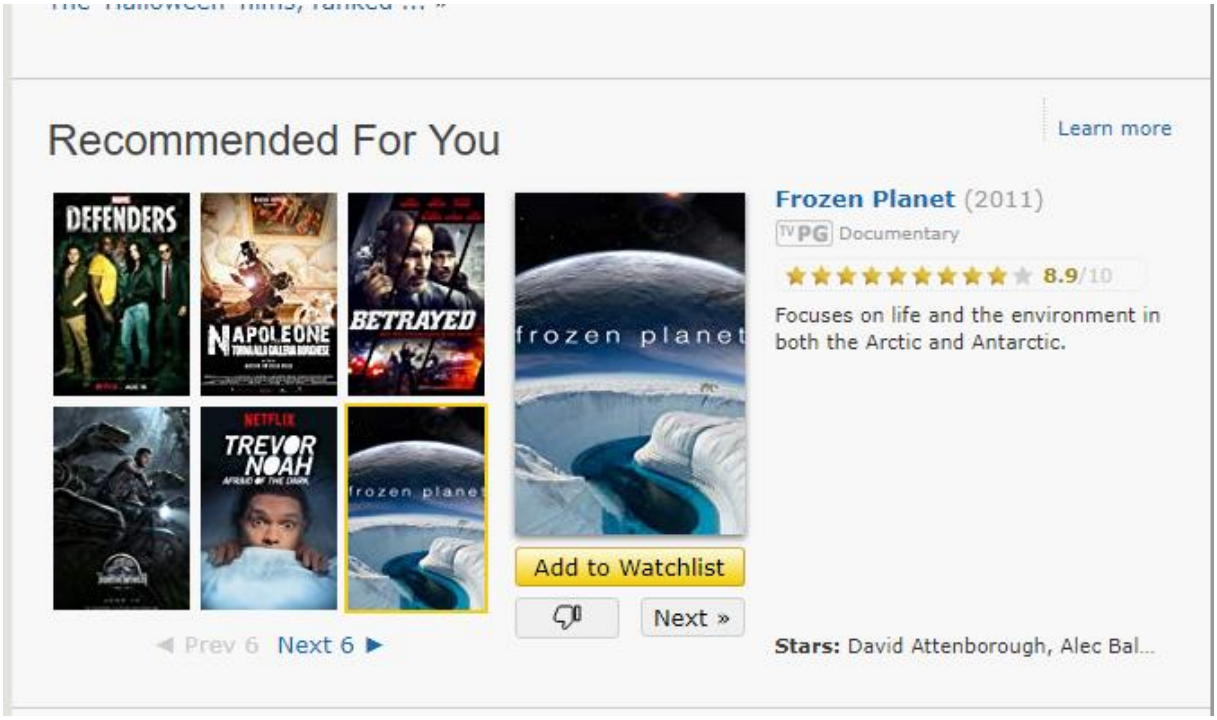
Netflix’s Recommendation System is one of the biggest successes in the history of recommender systems. It has 80% stream time. Netflix uses hybrid recommendation system. Also, one of Netflix's biggest feature is recommending by the thumbnail of a show. Which means that, if a person is into action or violent movie this person will be recommended more violent frames from the show.[11]

Steam’s Recommendation System

Steam’s recommendation system is based on the user's history of chosen games. The idea of choosing this method is more like giving equal opportunities to the companies with less popularity. Steam is now the most used game store in the whole world. Because non-popular companies will have more chances to advertise their games.[12]

IMDb’s Recommendation System

When we rate a TV show or movie on IMDb it recommends other shows or movies based on important details like cast, genre, sub-genre, plot and summary.



[2]

As we can see above, I was recommended to rate Frozen Planet because I've watched David Attenborough's wildlife documentary series. In this case, IMDb suggested this to me based on the cast of the series.

Amazon's Recommendation System

Item-item collaborative filtering, or item-based, or item-to-item, is a form of collaborative filtering for recommender system based on the similarity between items calculated using people's ratings of those items. Item-item collaborative filtering was invented and used by in 1998.[13]

[14]

Python Libraries

A number of Python libraries are available that are specifically created for recommendation purposes. Here are the most popular ones:

- Surprise: A Python scikit building and analyzing recommender systems.
- Implicit: Fast Python Collaborative Filtering for Implicit Datasets.
- LightFM: Python implementation of a number of popular recommendation algorithms for both implicit and explicit feedback.
- pyspark.mlib.recommendation: Apache Spark's Machine Learning API.[15]

1.3 Conclusion

Recommendation systems have a strong presence in our world. What is expected is to use them in a way that we can give users the best experience. Since we are dealing with user data, we need to use machine learning. By feeding sample data to model

based methods we get the best predictions. Using the methods content-based, collaborative and hybrid, we suggest users the best suggestions, thus increasing their enjoyment to maximum.

2. Software Requirements Specification

2.1.INTRODUCTION

2.1.1 Purpose

The purpose of this project is to design a recommender system that recommends video games to help users to see more similar games to their taste. After the user registers, the system shows the user a few game types and lists the games they may like according to the user's preferences. The user rates the games they played and liked before. In this way, the system can find games that are closest to the user's style. Our main purpose in this document is to explain the requirements and functions of our project. This system is going to be designed as a mobile application. Our mobile application is called 'RecoSystem'. Our goal is to find users that have drowned to find games. This topic is not just related to game industries, it is a general problem of many industries such as, movies, music, videos, etc.. The system will have core statistics about users and start recommending games. Each user will have a profile of their own where they can rate the recommended games, add a profile picture, etc.

2.1.2 Scope of the Project

The purpose of this project is to suggest games using certain algorithms based on games that people play and like. This system aims to help improve user interactions with the similar games that they have played before. With this system, the time it takes to get personalized advice will be reduced. With this system we aim to bring the feature to the Android platform in which users can access the application more easily. The project will contain specific Machine Learning algorithms to match the users history of played games. First of all, the system will ask the user to choose game genres the user likes. Then, the system will start suggesting games using machine learning algorithms based on the data, which is based on the preferences of other users, and is similar from the database that we will create for the application to the information with the current user. In addition, users can rate the games the user has played and liked before, so the system will be able to make better suggestions. The

application will be released to Android platforms, then other platforms in future versions.

2.1.3 Glossary

Term	Definition
App	Application
User	The person using and interacting with the system.
AI	Artificial Intelligence
ML	Machine Learning
Android	Android is a mobile operating system which is based on a derived Linux kernel. [16]
Python	Python is an interpreted high-level general-purpose programming language.[17]
Firebase Database	The Firebase Realtime Database is a cloud-hosted NoSQL database that lets you store and sync data between your users in real time.
React Native	React Native is an open-source UI software framework created by Meta Platforms, Inc.[18]
GUI	Graphical User Interface
API	API is an interface that utilizes communication between two clients [19]

2.1.4 Overview of the Document

This document is prepared in accordance with the IEEE Std 1016-2009 [5], IEEE Recommended Practice for Software Requirements Specifications [6]. This document has three main headings. The first main topic is Introduction. The introduction simply explains the main purpose of this project in general. The second main title is the Overall Description. This section considers the determining factors that influence requirements. The third part is Requirements Specification, it contains all the requirements of the system, design constraints etc.

2.2. OVERALL DESCRIPTION

2.2.1 Product Perspective

RecoSystem is developed with machine learning algorithms to recommend games which will be implemented using Python. It will be designed as a mobile application. The development of the RecoSystem application will be implemented with React Native. For the database Firebase Database will be used.

- Users register to RecoSystem.
- RecoSystem will ask users to rate the games on a scale of 1-5 to see if they like the games. 5 means a like, 1 means a dislike at all.
- RecoSystem will check its data about the games user rates, checks related games accordingly and, based on the algorithm gets the best results.
- RecoSystem will use the Python Pandas Data Analysis Library to check its datasets, and display recommendation algorithm results.
- RecoSystem redirects the user to the recommendation page which is the outcome of the algorithm.
- RecoSystem will also show users the TOP10 most liked games to help users what the community is rating best.
- RecoSystem shows the user all the games the user can like, decreasing accordingly. Also shows users the Top 10 games most liked.
- If the user views game profiles they can see the games description and photos.

2.2.2 User Characteristic

2.2.2.1 User

- Users must be familiar with mobile phones.
- Users will have to use an Android phone and download the application in order to access the system.
- Users must have an internet connection.
- When users register, they can choose the games they like.
- Users can rate the recommended games.
- Users can navigate through sections like homepage, game profile.

2.2.2.2 System

- System provides model with algorithm.
- System checks clock.
-

2.2.3 General Constraints and Assumptions

RecoSystem will be available to everyone with a mobile phone and an internet connection. The system will be released for android phones. Users must have an android phone. It will be assumed that users will have an internet connection. Internet connection is a restriction for the system.

2.3. REQUIREMENTS SPECIFICATION

2.3.1 External Interface Requirements

2.3.1.1. User Interfaces

The user interface of our application will be in English. However, in later versions there will be multiple language selection options. The system will be quite understandable and easy to use. RecoSystem will be necessary for users to register and login to the application. There will be a home screen, recommendation page, and a game profile section.

2.3.1.2. Hardware Interfaces

Mobile phones support 5.1 lollipop and higher and IOS version is to be decided.

2.3.1.3. Software Interfaces

The application will run on Android, Python ML libraries will be used.

2.3.1.4. Communications Interfaces

The user must have an internet connection and an android phone.

2.3.2 Functional Requirements

Use Case Number	1
Use Case Name	Login
Summary	User logs in the system.
Actor	User
Trigger	Login button clicked.
Precondition	Actors must be connected to the internet. Actors must have an android phone. Actors must download the application. Actors must be registered.
Scenario	User enters already existing username and password. Click Login button.
Exceptional Situations & Alternative Flow	The actor's phone may not be connected to the Internet. Server could be under maintenance. The actor entered an incorrect username or password. There is no account with the entered username.
Postcondition	If it is the first login of the actor, it is directed to the first time login screen. If it's not the actor's first login, redirects to the home page.

Use Case Number	2
Use Case Name	Register
Summary	Actor registers to the system
Actor	User
Trigger	Register button clicked
Precondition	Actors must be connected to the internet. Actors must have an android phone. Actors must download the application.
Scenario	Actors must enter information such as username, age, gender, email, password, and location. The actor must re-enter the password for verification purposes. Email verification link will be sent to the actor. Actor registers to the system after clicking on the link.
Exceptional Situations & Alternative Flow	The actor's phone may not be connected to the Internet. Server could be under maintenance. The email or username entered by the user has been used before.
Postcondition	Actors redirect to the login page.

Use Case Number	3
Use Case Name	Choose Liked Games
Summary	Actors choose the games they have played and liked at the first login after registration
Actor	User
Trigger	After first time login
Precondition	Actors must be connected to the internet. Actors must have an android phone. Actors must download the application. Actors must be registered. Actors must login to the system for the first time.
Scenario	The first time screen appears when the actor logs into the system for the first time. Actor sees the names and pictures of many games on the screen. Actor chooses the games he has played before and likes. Clicks the Next button.
Exceptional Situations & Alternative Flow	The actor's phone may not be connected to the Internet. Actor must check the internet connection. Server could be under maintenance. Actor must restart the application.

Use Case Number	4
Use Case Name	Search for Games
Summary	Actors write the keywords for the game they are looking for.
Actor	User
Trigger	Actors click on search bar.
Precondition	Any keyword must be entered before pressing Search button.
Scenario	Actors get into the home page. Actors click on the search bar and write keywords for the game they are looking for and click the Search button. Any games matched with keywords are shown.
Exceptional Situations & Alternative Flow	If there is nothing entered, and clicked search, nothing happens.
Postcondition	System finds and shows Actors the games they searched.

Use Case Number	5
Use Case Name	Rate the Games
Summary	Actors rate the games scaling from 1-5.
Actor	User
Trigger	Actors click on stars on the game's profile.
Precondition	-
Scenario	Actors find the game they want to rate. Actors click on the game and enter the game's profile. Actors choose how many stars they want to give to the game ranging from 1 to 5. Actors click on the star they want to give.
Exceptional Situations & Alternative Flow	If actors are already rated but want to change the rating, they can click on stars again with a different star chosen.
Postcondition	Rating to a game has given ranging from 1 to 5 by Actors.

Use Case Number	6
Use Case Name	Check Recommended Games
Summary	Actors check the recommendation page to see recommended games.
Actor	User
Trigger	Actors click on the Recommended Games page.
Precondition	Actors chose liked games when they first login.
Scenario	Actors click on the Recommended Games page button from the home page. Actors can see recommended games that algorithm has provided.
Exceptional Stiuations & Alternative Flow	-
Postcondition	Actors can see recommended games.

A

Use Case Number	7
Use Case Name	Create Model
Summary	System recommendation algorithm creates model
Actor	User, System
Trigger	After Actors's first time login process ends.
Precondition	First time login.
Scenario	Actors first time log in. After Actors chose liked games, algorithm creates model.
Exceptional Stiuations & Alternative Flow	If no user has logged in yet, Algorithm uses raw dataset to create model.
Postcondition	Model created for recommending games.

Use Case Number	8
Use Case Name	Update Model
Summary	Every 24 hours System updates model.
Actor	System
Trigger	Every 24 hours at 12am
Precondition	A model has been created before.
Scenario	System checks if the clock is 12am. System considers last 24 hours datas to update the model.
Exceptional Stiuations & Alternative Flow	Even though no new data processed last 24
Postcondition	New model created based on the old model.

Use Case Number	9
Use Case Name	Get Recommendations
Summary	User views the recommendations.
Actor	User
Trigger	View Recommendation Page
Precondition	Actor clicks the recommendation page by button of the home page.
Scenario	Actor logs in to the system. Actor clicks on the recommendation page from the button of the home page.
Exceptional Stiuations & Alternative Flow	-
Postcondition	Actor can view recommended games

2.3.3 Software System attributes

2.3.3.1. Portability

RecoSystem is designed for the devices Android version 5.1(Lollipop) and above, so RecoSystem can work for new versions of the Android platform. With the upcoming updates, we are going to release IOS versions.

2.3.3.2. Usability

- Software should be able to let users see images clearly.
- Software should be able to let users rate

2.3.3.3. Adaptability

There are 2 user types, both can use the application differently.

2.3.4 Safety Requirement

For security reasons, users can not access the database or add new games to the app.The users must choose at least 8 characters long, must include alphanumeric and special characters

3. Software Design Document

3.1. Introduction

3.1.1 Purpose

The purpose of this project is to design a recommender system that recommends video games. Building a system that suggests video games to any user with any taste is the main purpose. This documentation includes detailed information about the design of the project.

3.1.2 Scope of Project

The project aims to recommend video games using user data and applying certain algorithms. This document will show how system functions, data transferred and give details of implementation.

3.1.3 Glossary

Term	Definition
App	Application
User	The person using and interacting with the system.
ML	Machine Learning
Android	Android is a mobile operating system which is based on a derived Linux kernel. [22]
Python	Python is an interpreted high-level general-purpose programming language. [23]
Firebase Database	The Firebase Realtime Database is a cloud-hosted NoSQL database that lets you store and sync data between your users in real time.
HTML	The HyperText Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as CSS and JavaScript.
CSS	Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML.
JavaScript	JavaScript is a dynamic programming language that is widely used in web browsers.
React Native	React Native is an open-source UI software framework created by Meta Platforms, Inc. [24]

3.2. ARCHITECTURE DESIGN

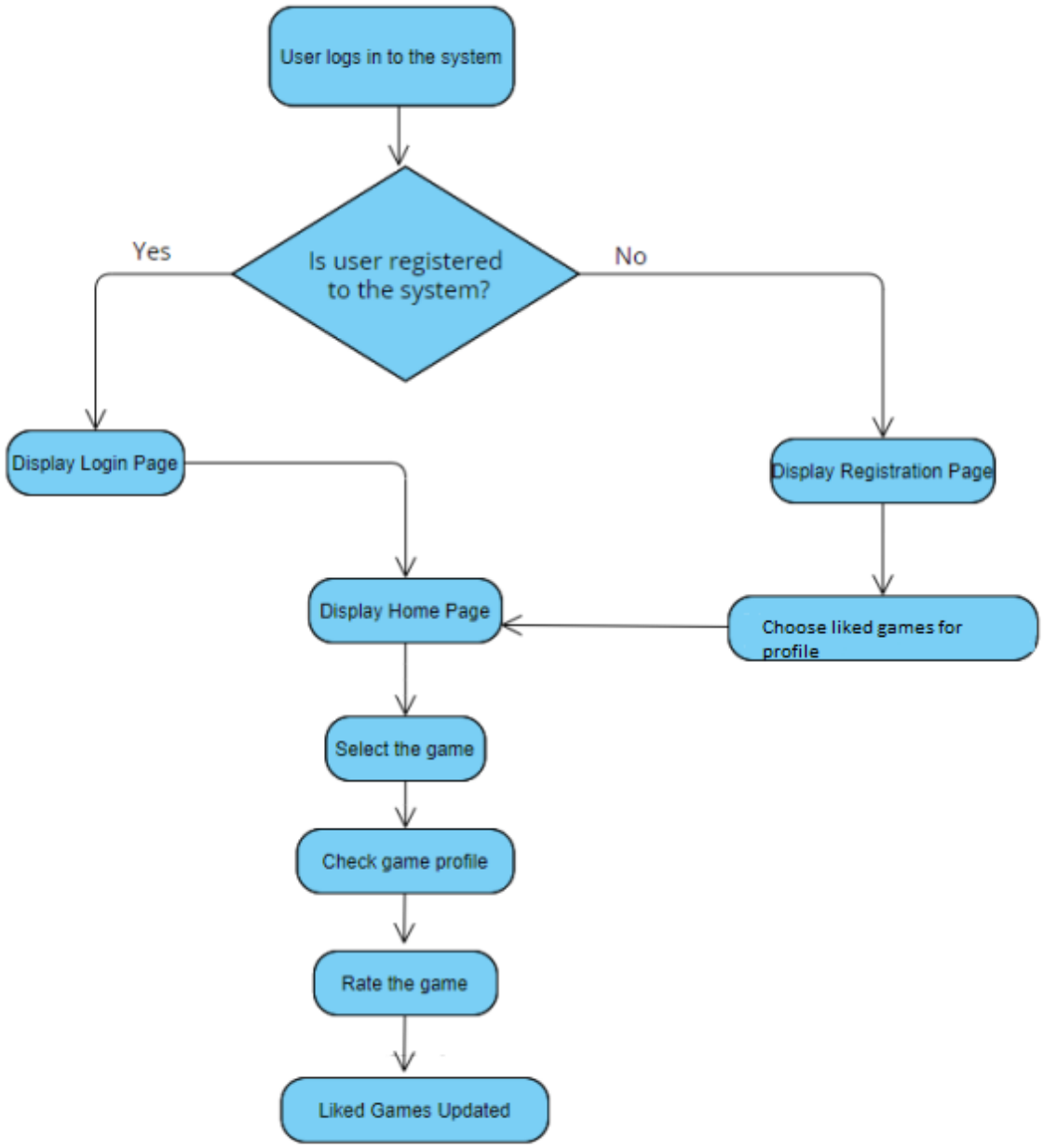
3.2.1 Technologies Used

RecoSystem uses many technologies and algorithms and combines all in a way that their best parts are used. RecoSystem uses Python Pandas Data Analysis Library to manage datasets and user data. RecoSystem uses Firebase database to manage databases and users. RecoSystem is developed with React Native and includes HTML, CSS, Javascript. Mobile phones support 5.1 lollipop and higher required.

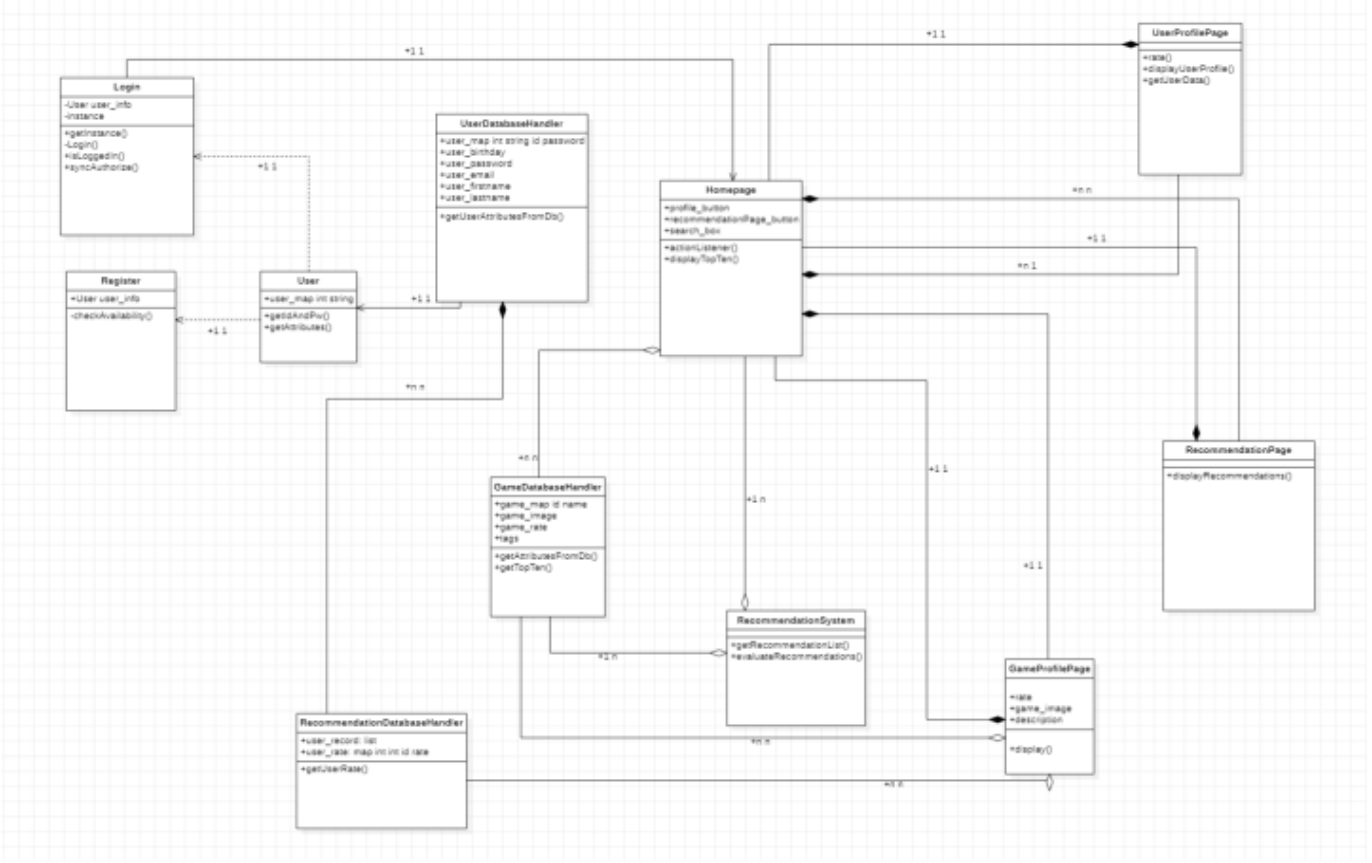
3.2.2 Use Case Diagram

3.2.3 Activity Diagram

3.2.3.1 User Activity Diagram



3.2.4 Class Diagram



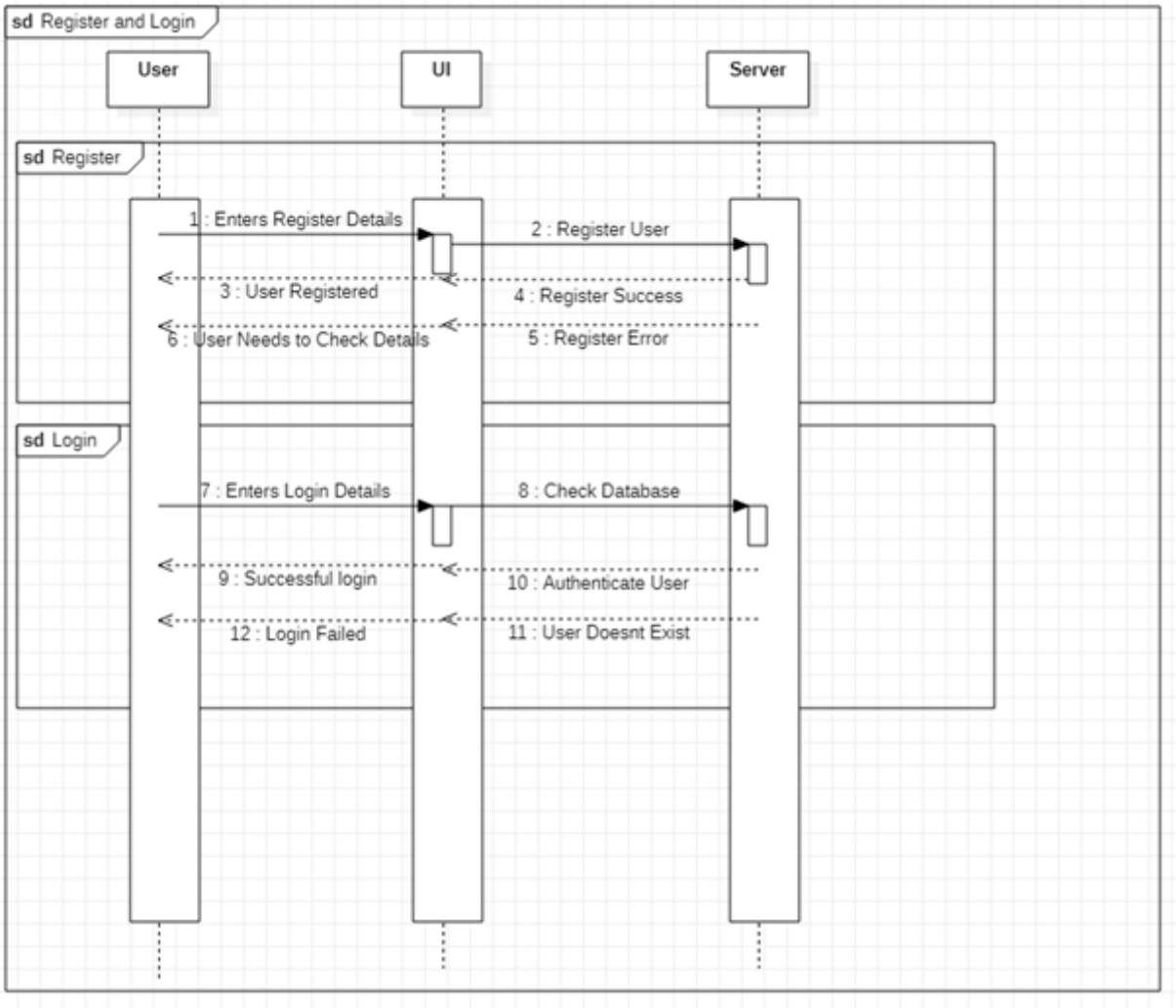
3.2.5 Architectural Design

Application Server: An application server is used to keep the selections made for the recommendation, to keep the application up-to-date, such as the initial registration information.

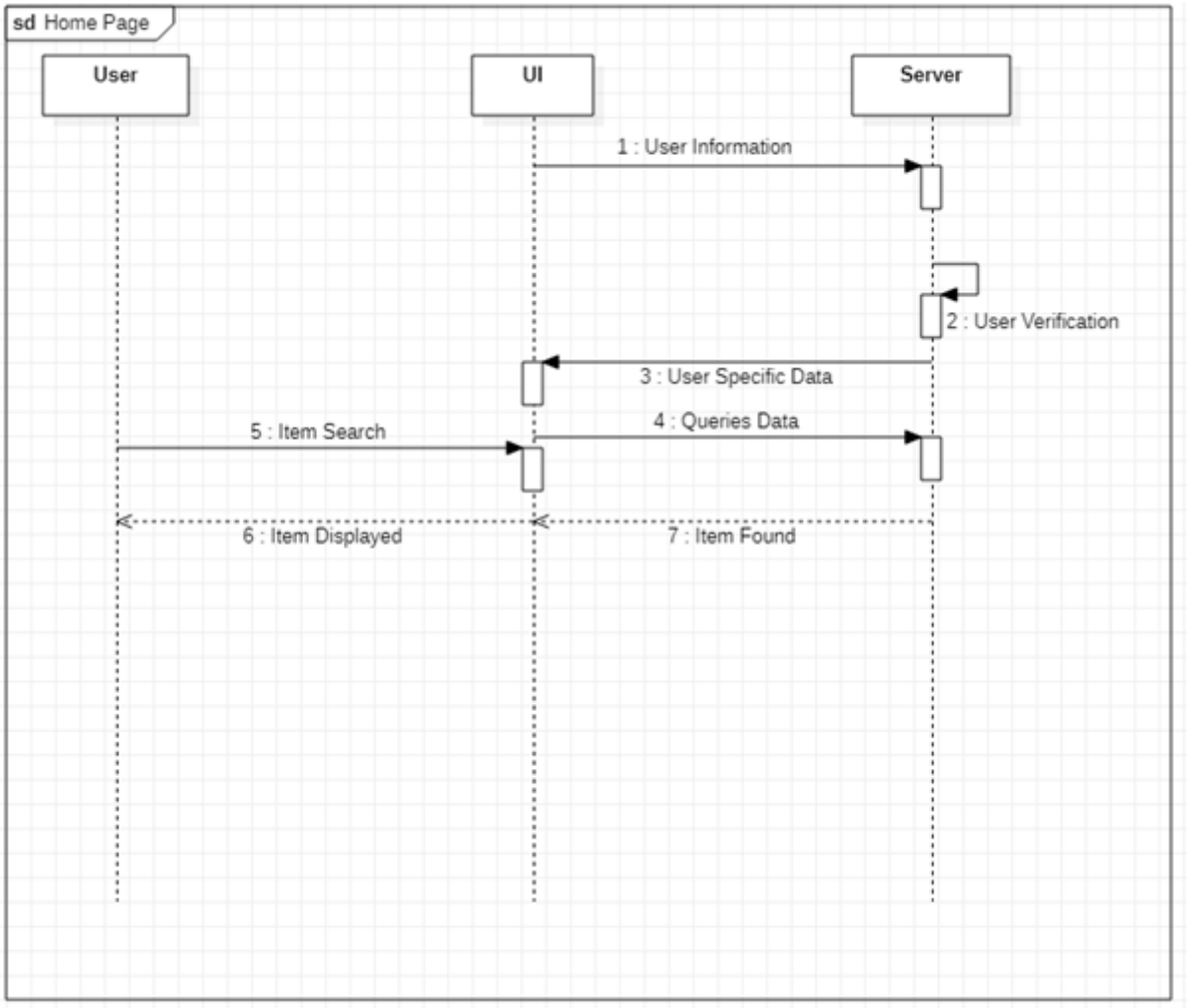
Database Server: The data or backend layer of a web application is the database server, which handles system data and transports data to and from the system database.

3.2.6 Sequence Diagram

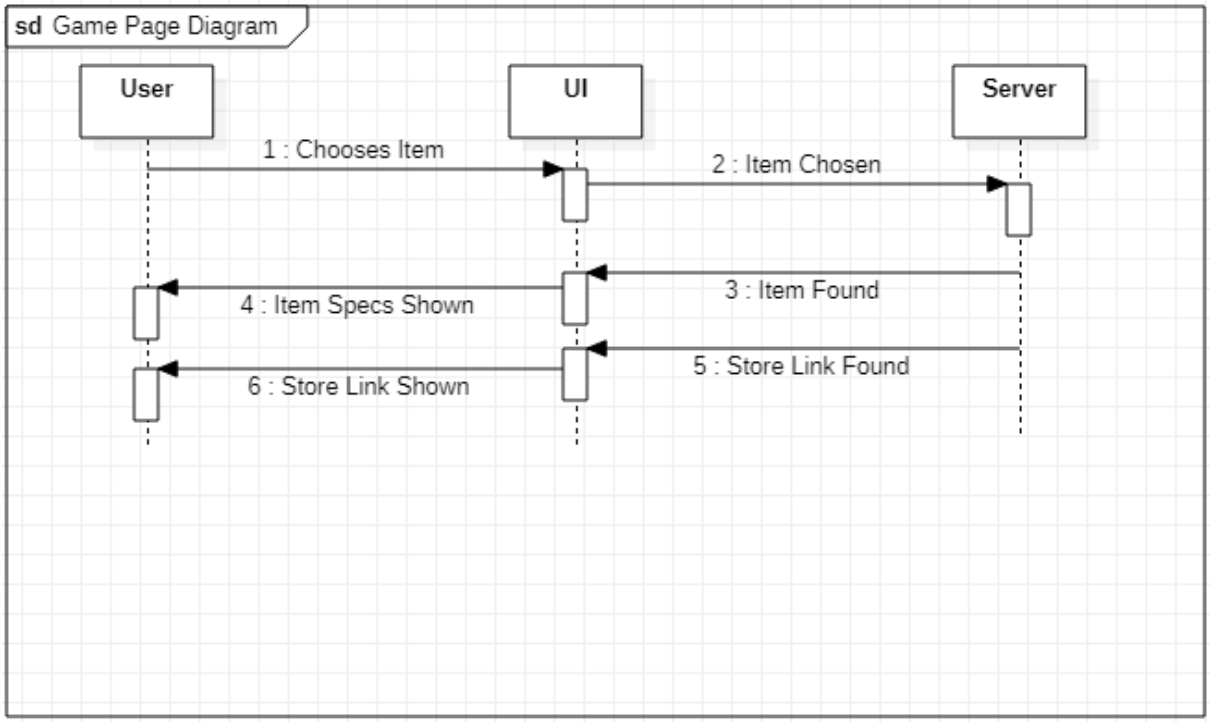
2.6.1 Register and Login Diagram



2.6.2 Home Page Diagram

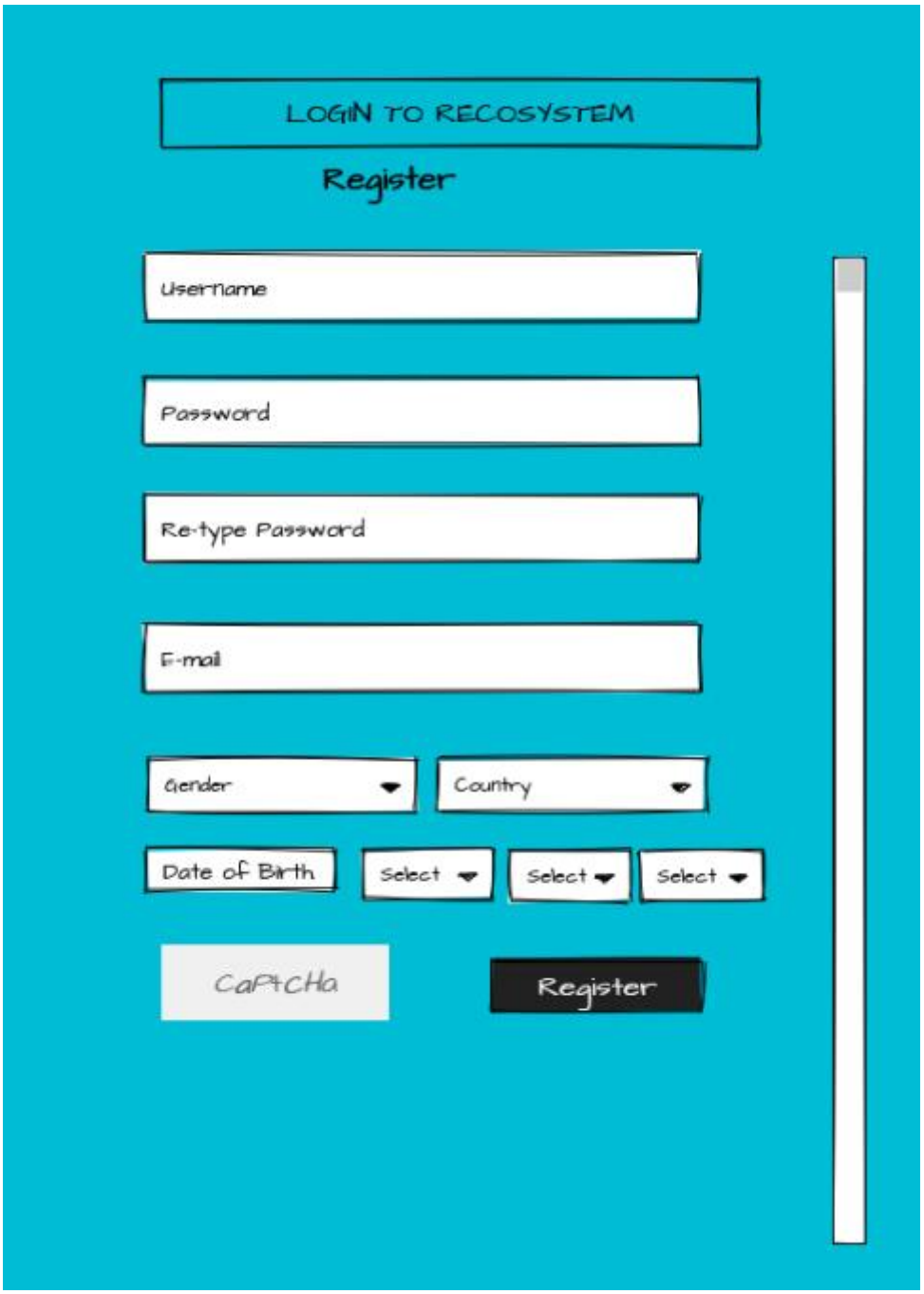


2.6.3 Game Page Diagram



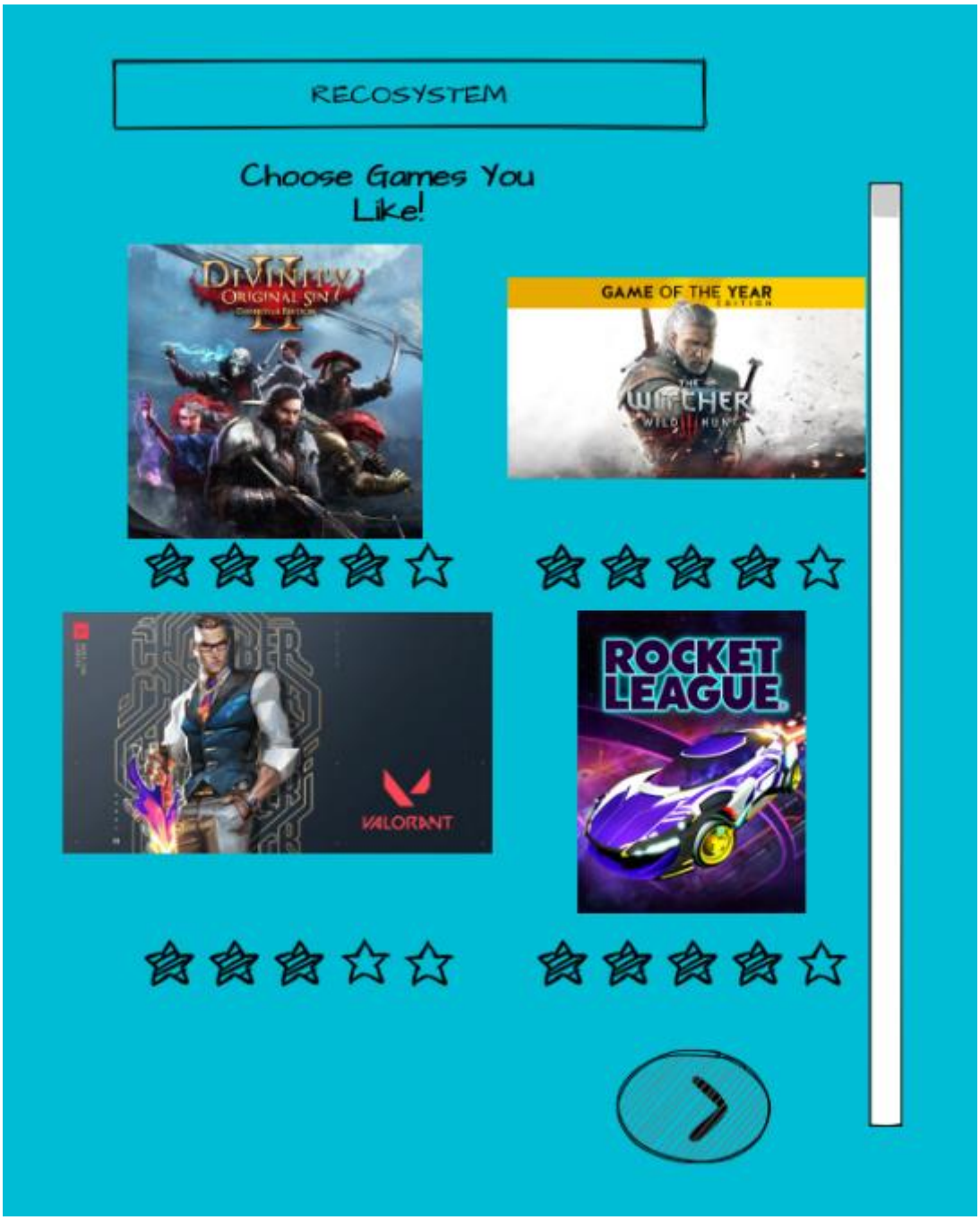
3.3. User Interface

3.3.1 Register Page

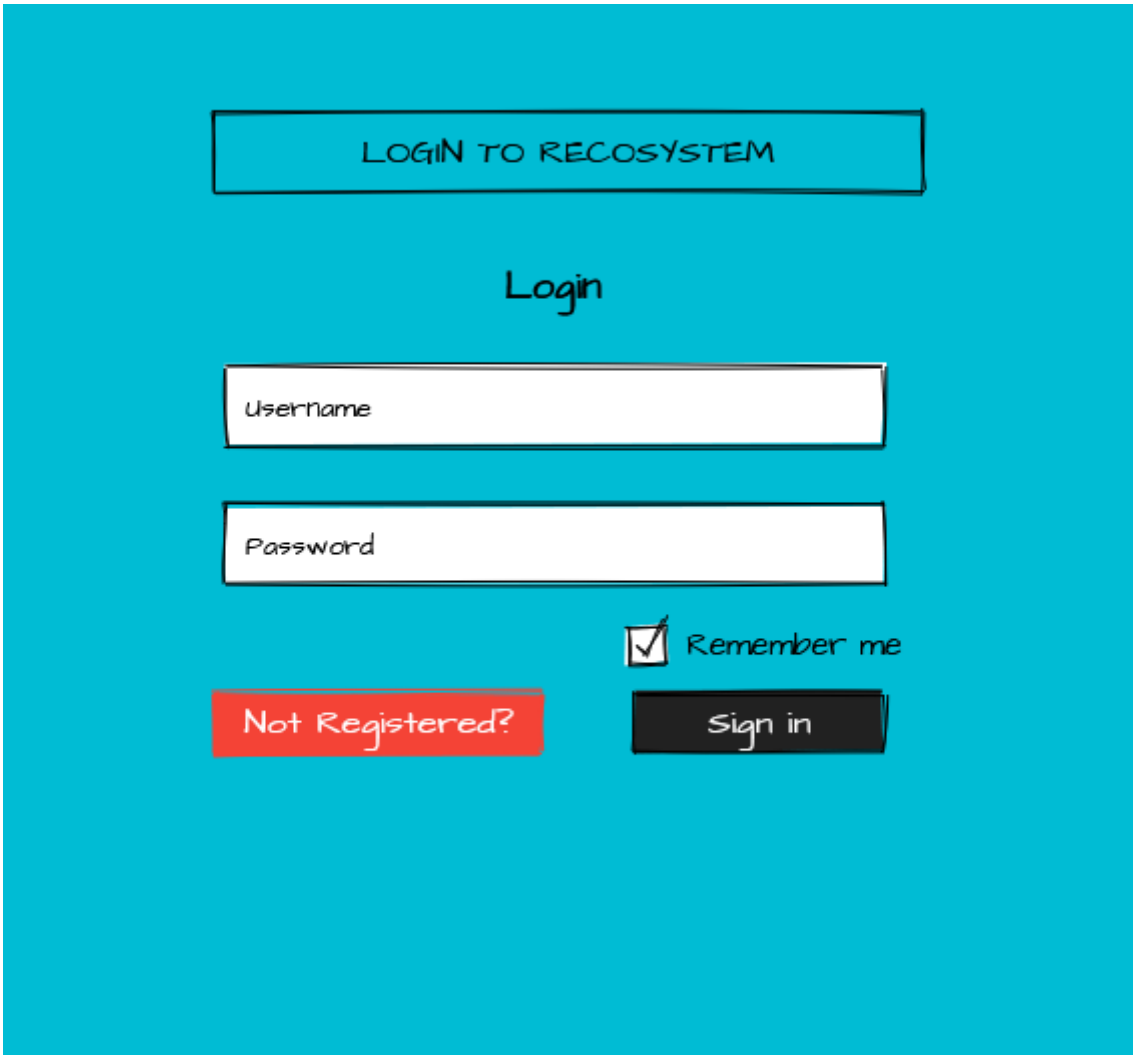


3.3.2 User Liked Games Page

After registration, a page appears where you can rate the games you have played and liked before.

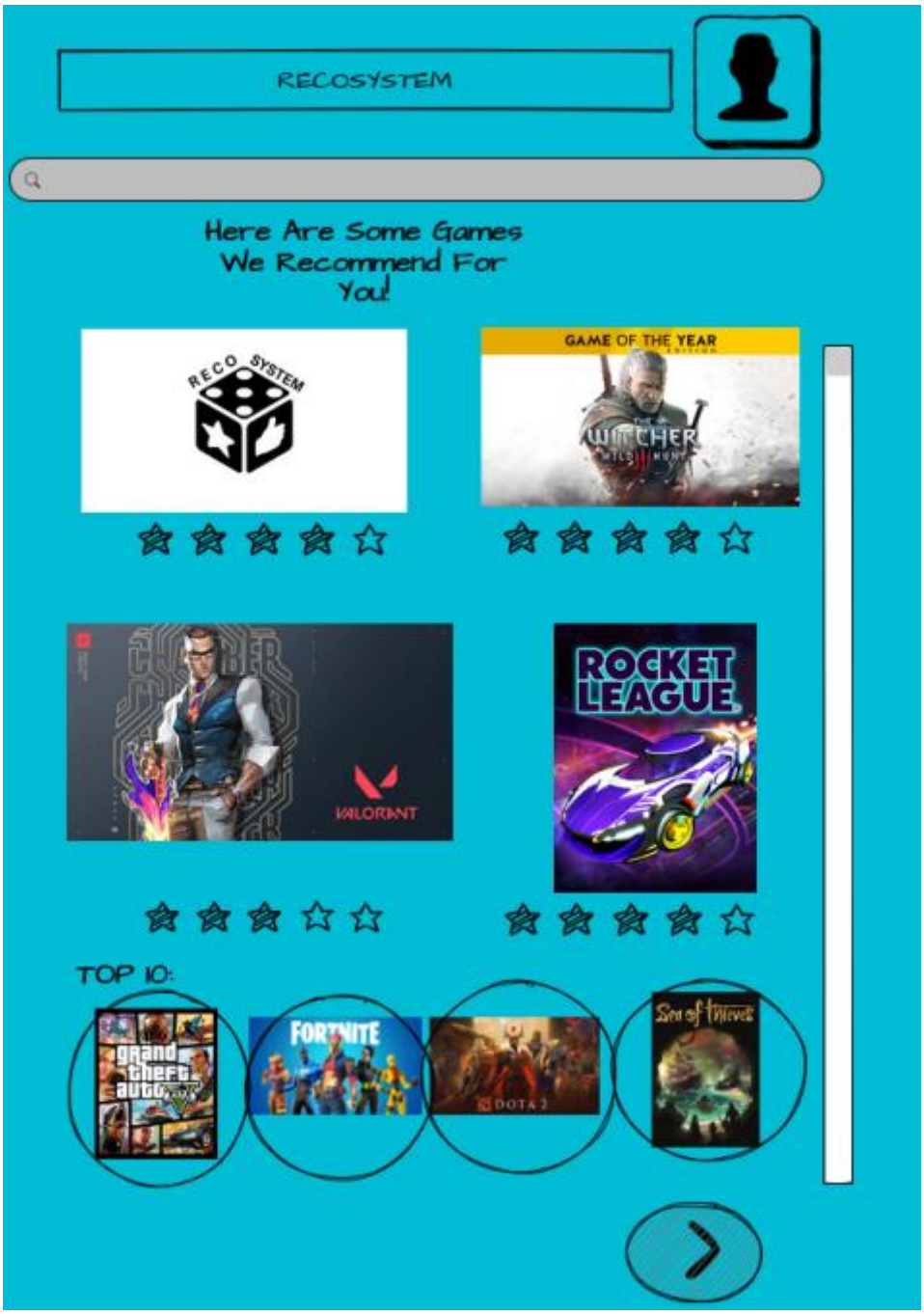


3.3.3 Login Page



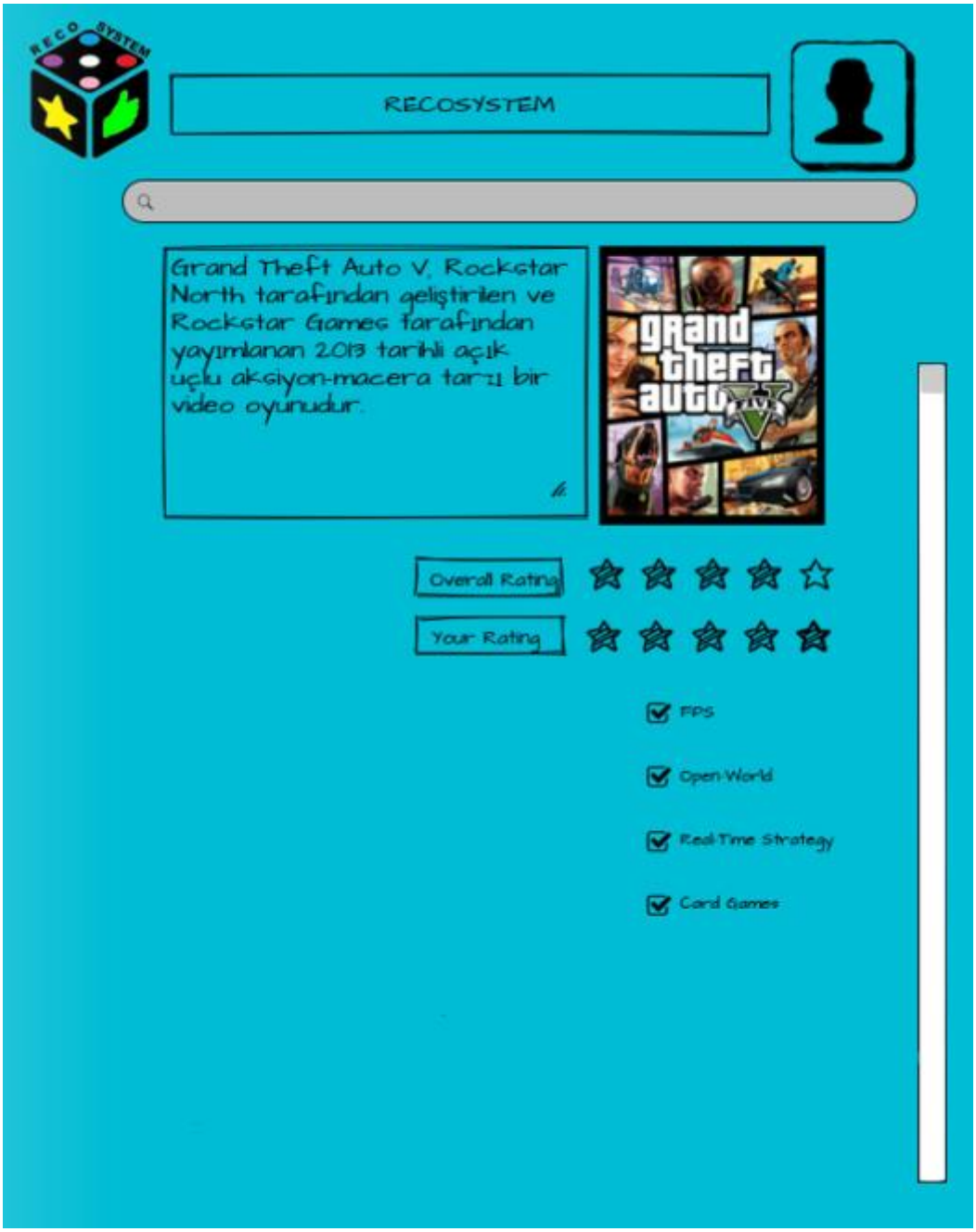
3.3.4 Home Page

After logging in, you will be directed to the homepage. On this page, there are recommendations given by the system based on your previous selections. You can also see the 10 most liked games on this page.



3.3.5 Game Profile Page

Each game will have its own page. Information about the game and used rates can be seen here.



4. Test Plan & Results

4.1. Introduction

4.1.1 Version Control

Version No	Description of Changes	Date
1.0	First Version	April 3, 2022

4.1.2 Overview

In this test plan, game recommendation system called RecoSystem will be tested. With this test plan, the functional and performance requirements of the system will be tested. Requirements, functions and more details were previously specified in the SRS and SDD documents.

4.1.3 Scope

This document contains the test plan of the use cases. In this document, we have provided information on how our test criteria will be and how we will apply them.

4.1.4 Terminology

Term	Definition
RS	Recommendation System
TP	Test Plan
SRS	Software Requirements Specification
SDD	Software Design Document

4.2. FEATURES TO BE TESTED

In this section, we will provide general information about the features to be tested. For each feature mentioned here, there will be a Test Design Specification.

4.2.1 Register Button (REG.BTN)

It is used to register so that new users can use the features of the system.

4.2.2 Login Button (LOG.BTN)

It is used for users to log in to the system so that they can use the features of the system.

4.2.3 Search Button (SRCH.BTN)

Users use it to find the game they are looking for.

4.2.4 Get Recommendation Button (REC.BTN)

After users do the necessary steps, they use it to see the games that suit their taste.

4.2.5 Rate the Games (RATE.GMS)

Users can rate their favorite games. This way, the system can offer better recommendations.

4.2.6 Game Profile Button (PRFL.BTN)

Users enter here to view game profiles.

4.3. FEATURES NOT TO BE TESTED

Performance will not be tested. With this test plan, no loading tests will be applied to the system.

4.4. ITEM PASS/FAIL CRITERIA

4.4.1 Exit Criteria

The system will be considered successful when it fulfills the functions we have determined beforehand. These functions are explained in detail in our SRS and SDD documentation.

4.5. REFERENCES

[1] Group5_SRS, April 2, 2022. Available: https://github.com/CankayaUniversity/ceng-407-408-2021-2022-Game-Recommendation-System-using-Machine-Learning-Algorithms/wiki/Software-Requirements-Specification-(SRS)
[2] Group5_SDD, April 2, 2022. Available: https://github.com/CankayaUniversity/ceng-407-408-2021-2022-Game-Recommendation-System-using-Machine-Learning-Algorithms/wiki/Software-Design-Document-(SDD)

4.6. TEST DESIGN SPECIFICATIONS

4.6.1 Register Button (REG.BTN)

TC ID	Requirements	Priority	Scenario
REG.BTN.01	3.2.2	H	Select 'Register' button and there will open register page
REG.BTN.02	3.2.2	H	Enter non-existing username and non-existing email with valid password.
REG.BTN.03	3.2.2	H	Enter non-existing username and non-existing email with invalid password.
REG.BTN.04	3.2.2	H	Enter existing username and non-existing email with valid password.
REG.BTN.05	3.2.2	H	Enter existing username and non-existing email with invalid password.
REG.BTN.06	3.2.2	H	Enter non-existing username and existing email with valid password.
REG.BTN.07	3.2.2	H	Enter non-existing username and existing email with invalid password.
REG.BTN.08	3.2.2	H	Enter existing username and existing email with valid password.
REG.BTN.09	3.2.2	H	Enter existing username and existing email with invalid password.
REG.BTN.10	3.2.2	H	Leave the username or email or password fields blank.

4.6.2 Login Button (LOG.BTN)

TC ID	Requirements	Priority	Scenario
LOG.BTN.01	3.2.1	H	Select 'Login' button and there will open login page.
LOG.BTN.02	3.2.1	H	Enter existing username with invalid password.
LOG.BTN.03	3.2.1	H	Enter non-existing username with invalid password.
LOG.BTN.04	3.2.1	H	Enter existing username with valid password.
LOG.BTN.05	3.2.1	H	Enter non-existing username with valid password.

4.6.3 Search Button (SRCH.BTN)

TC ID	Requirements	Priority	Scenario
SRCH.BTN.01	3.2.4	M	Search for existing game name.
SRCH.BTN.02	3.2.4	M	Search for non-existing game name.

4.6.4 Get Recommendation Button (REC.BTN)

TC ID	Requirements	Priority	Scenario
REC.BTN.01	3.2.6	H	Select 'Get Recommendation' button and there will open this page.

4.6.5 Rate the Games (RATE.GMS)

TC ID	Requirements	Priority	Scenario
RATE.GMS.01	3.2.5	H	Rate the games.
RATE.GMS.02	3.2.5	H	Leave the rating area empty.

4.6.6 Game Profile Button (PRFL.BTN)

TC ID	Requirements	Priority	Scenario
PRFL.BTN.01	3.2.4	M	Select 'Game Profile' button and there will open this page.

4.7. Detailed Test Cases

TC ID	REG.BTN.01
Purpose	'Register' button and there will open register page
Requirements	3.2.2
Priority	High
Estimated Time Needed	Less than 1 minute
Dependency	Register page should be displayed
Setup	Register page should be accessible.
Procedure	Click on the Register Page.
Cleanup	Logout

TC ID	REG.BTN.02
Purpose	Enter non-existing username and non-existing email with valid password.
Requirements	3.2.2
Priority	High
Estimated Time Needed	Less than 1 minute
Dependency	Register page should be displayed
Setup	Register page should be accessible.
Procedure	Click on the Register Page. Enter a non-existing username Enter a non-existing email Enter a valid password Press Register button.
Cleanup	Refresh

TC ID	REG.BTN.03
Purpose	Enter non-existing username and non-existing email with invalid password.
Requirements	3.2.2
Priority	High
Estimated Time Needed	Less than 1 minute
Dependency	Register page should be displayed
Setup	Register page should be accessible.
Procedure	Click on the Register Page. Enter a non-existing username Enter a non-existing email Enter a invalid password Press Register button.
Cleanup	Refresh

TC ID	REG.BTN.04
Purpose	Enter existing username and non-existing email with valid password.
Requirements	3.2.2
Priority	High
Estimated Time Needed	Less than 1 minute
Dependency	Register page should be displayed
Setup	Register page should be accessible.
Procedure	Click on the Register Page. Enter an existing username Enter a non-existing email Enter a valid password Press Register button.
Cleanup	Refresh

TC ID	REG.BTN.05
Purpose	Enter existing username and non-existing email with invalid password.
Requirements	3.2.2
Priority	High
Estimated Time Needed	Less than 1 minute
Dependency	Register page should be displayed
Setup	Register page should be accessible.
Procedure	Click on the Register Page. Enter an existing username Enter a non-existing email Enter a invalid password Press Register button.
Cleanup	Refresh

TC ID	REG.BTN.06
Purpose	Enter non-existing username and existing email with valid password.
Requirements	3.2.2
Priority	High
Estimated Time Needed	Less than 1 minute
Dependency	Register page should be displayed
Setup	Register page should be accessible.
Procedure	Click on the Register Page. Enter a non-existing username Enter a existing email Enter a valid password Press Register button.
Cleanup	Refresh

TC ID	REG.BTN.07
Purpose	Enter non-existing username and existing email with invalid password.
Requirements	3.2.2
Priority	High
Estimated Time Needed	Less than 1 minute
Dependency	Register page should be displayed
Setup	Register page should be accessible.
Procedure	Click on the Register Page. Enter a non-existing username Enter a existing email Enter a invalid password Press Register button.
Cleanup	Refresh

TC ID	REG.BTN.08
Purpose	Enter existing username and existing email with valid password.
Requirements	3.2.2
Priority	High
Estimated Time Needed	Less than 1 minute
Dependency	Register page should be displayed
Setup	Register page should be accessible.
Procedure	Click on the Register Page. Enter an existing username Enter a existing email Enter a valid password Press Register button.
Cleanup	Refresh

TC ID	REG.BTN.09
Purpose	Enter existing username and existing email with invalid password.
Requirements	3.2.2
Priority	High
Estimated Time Needed	Less than 1 minute
Dependency	Register page should be displayed
Setup	Register page should be accessible.
Procedure	Click on the Register Page. Enter an existing username Enter a existing email Enter a invalid password Press Register button.
Cleanup	Refresh

TC ID	REG.BTN.10
Purpose	Leave the username or email or password fields blank.
Requirements	3.2.2
Priority	High
Estimated Time Needed	Less than 1 minute
Dependency	Register page should be displayed
Setup	Register page should be accessible.
Procedure	Click on the Register Page. Leave the blank all the fields Press Register button.
Cleanup	Refresh

TC ID	LOG.BTN.01
Purpose	Select 'Login' button and there will open login page.
Requirements	3.2.1
Priority	High
Estimated Time Needed	Less than 1 minute
Dependency	Login page should be displayed.
Setup	Login page should be accessible.
Procedure	Click on the Login Page.
Cleanup	Logout

TC ID	LOG.BTN.02
Purpose	Enter existing username with invalid password.
Requirements	3.2.1
Priority	High
Estimated Time Needed	Less than 1 minute
Dependency	Login page should be displayed.
Setup	Login page should be accessible.
Procedure	Click on the Login Page. Enter an existing username Enter a invalid password Press Login button.
Cleanup	Refresh

TC ID	LOG.BTN.03
Purpose	Enter non-existing username with invalid password.
Requirements	3.2.1
Priority	High
Estimated Time Needed	Less than 1 minute
Dependency	Login page should be displayed.
Setup	Login page should be accessible.
Procedure	Click on the Login Page. Enter a non-existing username Enter a invalid password Press Login button.
Cleanup	Refresh

TC ID	LOG.BTN.04
Purpose	Enter existing username with valid password.
Requirements	3.2.1
Priority	High
Estimated Time Needed	Less than 1 minute
Dependency	Login page should be displayed.
Setup	Login page should be accessible.
Procedure	Click on the Login Page. Enter an existing username Enter a valid password Press Login button.
Cleanup	Refresh

TC ID	LOG.BTN.05
Purpose	Enter non-existing username with valid password.
Requirements	3.2.1
Priority	High
Estimated Time Needed	Less than 1 minute
Dependency	Login page should be displayed.
Setup	Login page should be accessible.
Procedure	Click on the Login Page. Enter a non-existing username Enter a valid password Press Login button.
Cleanup	Refresh

TC ID	SRCH.BTN.01
Purpose	Search for existing game name.
Requirements	3.2.4
Priority	Medium
Estimated Time Needed	Less than 1 minute
Dependency	Search button should be displayed.
Setup	Search button should be accessible.
Procedure	Click on the Search Button. Enter an existing game name Press Search button.
Cleanup	Logout

TC ID	SRCH.BTN.02
Purpose	Search for non-existing game name.
Requirements	3.2.4
Priority	Medium
Estimated Time Needed	Less than 1 minute
Dependency	Search button should be displayed.
Setup	Search button should be accessible.
Procedure	Click on the Search Button. Enter a non-existing game name Press Search button.
Cleanup	Logout

TC ID	REC.BTN.01
Purpose	Select 'Get Recommendation' button and there will open this page.
Requirements	3.2.6
Priority	High
Estimated Time Needed	Less than 1 minute
Dependency	Get Recommendation button should be displayed.
Setup	Get Recommendation button should be accessible.
Procedure	Click on the Get Recommendation Button.
Cleanup	Logout

TC ID	RATE.GMS.01
Purpose	Rate the games.
Requirements	3.2.5
Priority	High
Estimated Time Needed	Less than 1 minute
Dependency	Rate Games button should be displayed.
Setup	Rate Games button should be accessible.
Procedure	Click on the Rate Games. Rate the games. Click save button.
Cleanup	Logout

TC ID	RATE.GMS.02
Purpose	Leave the rating area empty.
Requirements	3.2.5
Priority	High
Estimated Time Needed	Less than 1 minute
Dependency	Rate Games button should be displayed.
Setup	Rate Games button should be accessible.
Procedure	Click on the Rate Games. Leave the rate area empty. Click save button.
Cleanup	Logout

TC ID	PRFL.BTN.01
Purpose	Select 'Game Profile' button and there will open this page
Requirements	3.2.4
Priority	Medium
Estimated Time Needed	Less than 1 minute
Dependency	Game Profile button should be displayed.
Setup	Game Profile button should be accessible.
Procedure	Click on the Game Profile button.
Cleanup	Logout

4.8. Individual Test Results

TC ID	Requirements	Priority	Result	Run By
REG.BTN.01	3.2.2	H	Passed	Kutay Kabadaş
REG.BTN.02	3.2.2	H	Passed	Kutay Kabadaş
REG.BTN.03	3.2.2	H	Passed	Kutay Kabadaş
REG.BTN.04	3.2.2	H	Passed	Kutay Kabadaş
REG.BTN.05	3.2.2	H	Passed	Kutay Kabadaş
REG.BTN.06	3.2.2	H	Passed	Kutay Kabadaş
REG.BTN.07	3.2.2	H	Passed	Kutay Kabadaş
REG.BTN.08	3.2.2	H	Passed	Kutay Kabadaş
REG.BTN.09	3.2.2	H	Passed	Kutay Kabadaş
REG.BTN.10	3.2.2	H	Passed	Kutay Kabadaş
LOG.BTN.01	3.2.1	H	Passed	Berfin Varlı
LOG.BTN.02	3.2.1	H	Passed	Berfin Varlı
LOG.BTN.03	3.2.1	H	Passed	Berfin Varlı
LOG.BTN.04	3.2.1	H	Passed	Berfin Varlı
LOG.BTN.05	3.2.1	H	Passed	Berfin Varlı
SRCH.BTN.01	3.2.4	M	Passed	Berfin Varlı
SRCH.BTN.02	3.2.4	M	Passed	Berfin Varlı
REC.BTN.01	3.2.6	H	Passed	Tolga Bolat
RATE.GMS.01	3.2.5	H	Passed	Tolga Bolat
RATE.GMS.02	3.2.5	H	Passed	Tolga Bolat
PRFL.BTN.01	3.2.4	M	Passed	Tolga Bolat

4.9. Summary of Test Results

Priority	Number of TCs	Executed	Passed
High	18	18	18
Medium	3	3	3
Total	21	21	21

4.10 Exit Criteria

We executed test cases which are given in the above table. All of test cases are passed. Therefore, exit criteria is provided.

5.References

[1] Rocca, (2019, June 3). “Introduction to recommender systems”, https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada
[2] Badreesh Shetty, (2019, July 24). “An In-Depth Guide to How Recommender Systems Work”, https://builtin.com/data-science/recommender-systems
[3] Sanam Malhotra, (2020, August 25). “5 Unique Recommendation Systems with Machine Learning” https://artificialintelligence.oodles.io/blogs/recommendation-systems-with-machine-learning/
[4] Pathairush Seeda, (2021, Oct 13). A Complete Guide To Recommender Systems — Tutorial with Sklearn, Surprise, Keras, Recommenders https://towardsdatascience.com/a-complete-guide-to-recommender-system-tutorial-with-sklearn-surprise-keras-recommender-5e52e8ceace1
[5] Wikipedia: Collaborative filtering, (2021, Oct 29). https://en.wikipedia.org/wiki/Collaborative_filtering
[6] George Seif, (2019, Sep). ”An Easy Introduction to Machine Learning Recommender Systems”, https://www.kdnuggets.com/2019/09/machine-learning-recommender-systems.html
[7] Rachit Gupta, (2020, Jul 16). “User-Based Collaborative Filtering”, https://www.geeksforgeeks.org/user-based-collaborative-filtering/
[8] Gregory D. LindenJennifer A. JacobiEric A. Benson, "Collaborative recommendations using item-to-item similarity mappings", 1998
[9] Wikipedia: Recommender system, (2021, Sep 23). https://en.wikipedia.org/wiki/Recommender_system#cite_ref-65
[10] Rohit Dwivedi, (2020, Apr 16). What Are Recommendation Systems in Machine Learning? https://www.analyticssteps.com/blogs/what-are-recommendation-systems-machine-learning
[11] David Chong, (2020, Apr 30). “Deep Dive into Netflix’s Recommender System”, https://towardsdatascience.com/deep-dive-into-netflixs-recommender-system-341806ae3b48
[12] Adi Robertson, (2019, July 11). “Steam’s new Interactive Recommender is built for finding ‘hidden gems’.” https://www.theverge.com/2019/7/11/20690231/valve-steam-labs-interactive-recommender-game-recommendation-machine-learning-tool
[13] Wikipedia: “Item-item collaborative filtering”, (2020, Dec 9). https://en.wikipedia.org/wiki/Item-item_collaborative_filtering

[14] Sanam Malhotra,(2020,Aug 25). “5 Unique Recommendation Systems with Machine Learning” https://artificialintelligence.oodles.io/blogs/recommendation-systems-with-machine-learning/
[15] Parul Pandey, (2019, May 17). “Recommendation Systems in the Real world”, https://towardsdatascience.com/recommendation-systems-in-the-real-world-51e3948772f3
[16] Wikipedia: Android (operating system),(2021, Dec 6). https://en.wikipedia.org/wiki/Android_(operating_system)
[17] Wikipedia: Python (programming language),(2021, Dec 7). https://en.wikipedia.org/wiki/Python_(programming_language)
[18] Wikipedia: React Native,(2021, Nov 15). https://en.wikipedia.org/wiki/React_Native
[19] MuleSoft: What is an API? (Application Programming Interface) (2021). https://www.mulesoft.com/resources/api/what-is-an-api
[20] "IEEE 1016-2009 - IEEE Standard for Information Technology--Systems Design--Software Design Descriptions", Standards.ieee.org, 2020. [Online]. Available: https://standards.ieee.org/standard/1016-2009.html. [Accessed: 08- Dec- 2021].
[21] "IEEE 830-1998 - IEEE Recommended Practice for Software Requirements Specifications", Standards.ieee.org, 2020. [Online]. Available: https://standards.ieee.org/standard/830-1998.html. [Accessed: 08- Dec- 2021].
[22] Wikipedia: Android (operating system),(2021, Dec 6). https://en.wikipedia.org/wiki/Android_(operating_system)
[23] Wikipedia: Python (programming language),(2021, Dec 7). https://en.wikipedia.org/wiki/Python_(programming_language)
[24] Wikipedia: React Native,(2021, Nov 15). https://en.wikipedia.org/wiki/React_Native