T.C.

MARMARA UNIVERSITY

FACULTY of ENGINEERING

COMPUTER ENGINEERING DEPARTMENT

30.11.2022

CSE4197 Project Specification Document

Title of the Project

“SUTOGA- A Social Network Platform for Gamers“

Group Members

Muhammet Eren ATALA 150119904

Hüseyin Kerem MİCAN 150119629

Emrullah SEVMİŞ 150119016

Supervised by

Asst. Prof. Sanem ARSLAN YILMAZ

Table of Contents

1. Problem Statement

With the development of the internet and computer parts, the game world, which used to consist of Atari, started to grow rapidly. With this growth, the number of factors that could bring people together at a common point increased, but these interactions are limited to the games they play together. Other gaming platforms are ineffective in terms of interaction.

Today, people have trouble coming together in terms of games. Finding a teammate of your own, having fun with similar things and sharing this fun with people, making your voice heard is almost impossible among millions of people and with the resources we have. With Sutoga, a social network application, we plan to bring together many functions that can solve the problems we mentioned before, and to offer new experiences to users that they will enjoy and meet their needs using the application.

2. Problem Description and Motivation

We define social network sites as web-based services that allow individuals to construct a public or semi-public profile within a bounded system, articulate a list of other users with whom they share a connection, and view and traverse their list of connections and those made by others within the system. The nature and nomenclature of these connections may vary from site to site. [1]

When we look outside the game world, there are many social sharing platforms used in the world. One of them is Instagram, which is known around the world, we see a successful and revolutionary application that brings everyone together and connects lives. It's the same with Facebook. When we look at it from a different perspective, we are exposed to things we don't want to see for hours every day. Although these platforms have their own algorithms, the results are not that successful at some parts. Platforms like Instagram and Facebook struggle to offer us what we want while trying their best to be "everything".

Another platform we all know about: LinkedIn. LinkedIn is a more targeted, specific platform than Instagram and Facebook. When you open the application, you mostly see recruitment news or updates about the business world and the companies you follow. Not the teenagers who shoot TikTok, football videos or TV series; you only see posts related to your needs and this is one of the reasons you use the app. Another is to connect with people who are suitable for you, expand your circle and open the door to opportunities that you never expected in your life. That's one of the things Sutoga tries to do.

When we say “game”, most people approach with a big prejudice because games are just arcades they went to in the 90s for them. But nowadays this is not so. In fact, it is not surprising that the game culture is developing at a great speed and reaching millions, together with the electronic goods we use every day. We cannot ignore the fact that people make money, make their voices heard and add color to the gray lives. They will have seen that, big companies in the world have been entering the game industry for years (Amazon, Netflix, Microsoft etc..).

A game-specific social media platform is one of the most suitable solutions to these problems. The profile system is one of these solutions so that people can interact easily and get to know each other without speaking. The personal information, favorite games, interactions and preferences in his/her profile show the personality of the user and what he/she is looking for.

With the machine learning technology developing with today's possibilities, it has been one of the most functional ways to determine the wishes of the users in such platforms and to organize the flow of the platform accordingly.

In the market where user experience is important, a desktop application is one of the most suitable and handy method for the segment we are addressing. With these features and functions, we can ensure that users enjoy and benefit from the application they use.

We propose a social media system that will show users more specific content, improve their socialization experience in the game world, expand their horizons, test their game culture and improve themselves, with the game recommendation system included.

3. Main Goal and Objectives

On the platform we propose, the main goal is to bring users together, enable them to socialize and to expand communication options, as well as to expand their gaming culture with a recommendation system using machine learning and design our own unique systems.

Objectives of the project are listed below.

- **Building a social network (Database design):** Social networking and social media are two different but related concepts. Social networking works on social platforms and what it does is connect users, build relationships between them. On the social media platform we have built, users will have a lot of relationships with each other and we will have to engage in many techniques and relational issues to establish these relationships. Creating a properly and error-free database can be mentioned under this title. Creating this database and establishing the relationships between them properly is one of the key points of our project.

- **Backend development:** For the functions that appear in the interface and that require processing to work, we will create the sections that can be requested from the backend and form the basis of the program. The backend part, which we will write in a layered way, will also be object-oriented and will treat users, shares, etc. as objects.

- **Frontend development:** The interface is what consumers care about and use the most. For this reason, we will find a sample template in the interface and modify and develop it over this example. In addition, we will use most modules and frameworks that should be on a social media platform.

- **Developing dynamic chat system, voice chat and video chat:** Private chat is one of the key features in a social media app. Video and voice chat are features that have been added to most popular applications. This feature found in most apps still mostly doesn't work properly, except for Discord (The most used audio and video communication application in the world.). Since we think that the video chat feature on the desktop application can work properly, we will add it to the platform and increase people's socialization options. Private chat groups are another feature of our application.

- **Design and implement the recommendation system:** Finding more than one dataset where we can have an idea about the algorithm we will write and see what we can use and forming an opinion on them. According to the results we will reach through the datasets we find; we will test what we can use and what can be obtained from the data we use and write the correct and accurate algorithm. We will combine the platform-independent recommendation system, written in a different software language, with the main code, and make it usable by users on the platform.

- **Deploying the whole system:** In order to test the platform we have created and see it fully operational, access over the internet must be open to more than one user. For this reason, we need to deploy the codes to a server where they can work together.

4. Related Work

GameTree

GameTree is an application for finding gamer friends, getting personal game recommendations, and coordinating gameplay sessions with people. There is a friend feature in the application, and friends can be searched according to the specified filters (age, gender, etc.). It has text messaging feature. The application can run on android, iOS and web platforms. [3]

Differences:

- Our application will only run on the desktop.
- In our application, users will be able to video and voice chat with each other.
- In our application, users will be able to create channels and have group chats.

Steam

The Steam client includes a digital storefront called the Steam Store through which users can purchase computer games. Users can add each other as friends, message each other. Thanks to the forum feature, topics can be discussed. Information about the games played by users (such as how many hours played) is recorded and displayed on the user's profile. In addition, the user is recommended games that he/she may like. The application can run on android, iOS and web platforms. [4]

Differences:

- Our application will only run on the desktop.
- There will be no game purchase feature in our app. There will also be no game launch feature, so we will not keep data such as how many hours the game has been played.
- In our application, users will be able to video and voice chat with each other.

Discord

Discord is an audio, video and text chat application that people use to hang out with their community and friends. Users can open channels with multiple people and chat there. Users can add each other as friends and directly message each other. Application can run on android, iOS and web platforms. [5]

Differences:

• Our application will only run on the desktop.

• In our application, there will be features such as recommending games to the user, discussion parts about games.

5. Scope

Our main purpose in the project is to gather the features that the market is not aware of but that we, as users, think should be, that we spend time unconsciously or that cannot be reached at all, and to offer users a clearly defined product-function based on needs. In addition, instead of being a plain social media, there will be functions that will satisfy users and use today's technologies correctly and properly.

5.1. Constraints

- The application only runs on desktop, it can’t be used on web or mobile.
- User must have Steam account to get a proper recommendation.
- Our system could have Cold Start problem. Cold start is a potential problem in computer-based information systems which involves a degree of automated data modelling. Specifically, it concerns the issue that the system cannot draw any inferences for users or items about which it has not yet gathered sufficient information [2].
- Server that we will deploy might have its own problems. The server may crash or an extreme event may happen out of control.
- Because it is a desktop application, computer compatibility issues may arise.
- The running speed of the application depends on the computer.
- There may be some difficulties in connecting the backend, frontend and machine learning codes. Some requests may cost a lot, it may take a while for the return value to come from the machine learning section. They should response in milliseconds and

can’t be more than a few seconds.

5.2 Assumptions

- It is assumed that users will have an internet connection (to use the “social” platform).
- It is assumed that users will have a compatible computer (macOS or Windows).
- It is assumed that users will have gaming data from other platforms (Steam, League of Legends etc..).
- It is assumed that server will have good performance (waiting time) in a way that can handle simultaneous actions. It should response in milliseconds.

6. Methodology and Technical Approach

Flow chart of this project shown below.

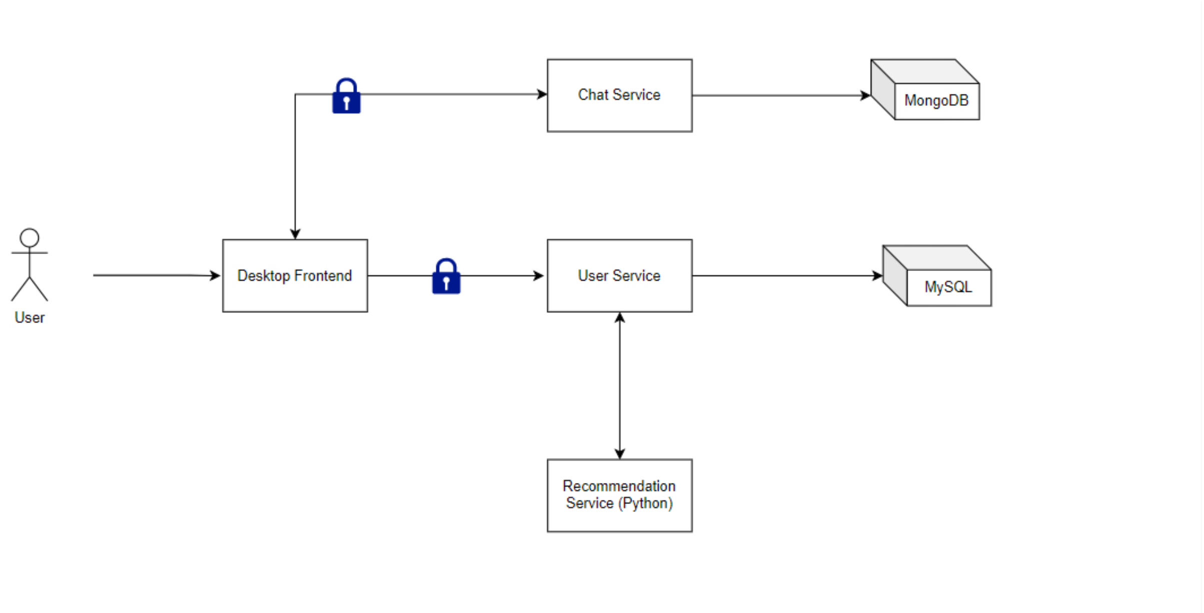


Figure 1. Flow chart.

Our project consists of three parts. These are a web application developed by writing backend and frontend, a written, voice and video chat system that enables dynamic communication, and a machine learning algorithm that provides game suggestions.

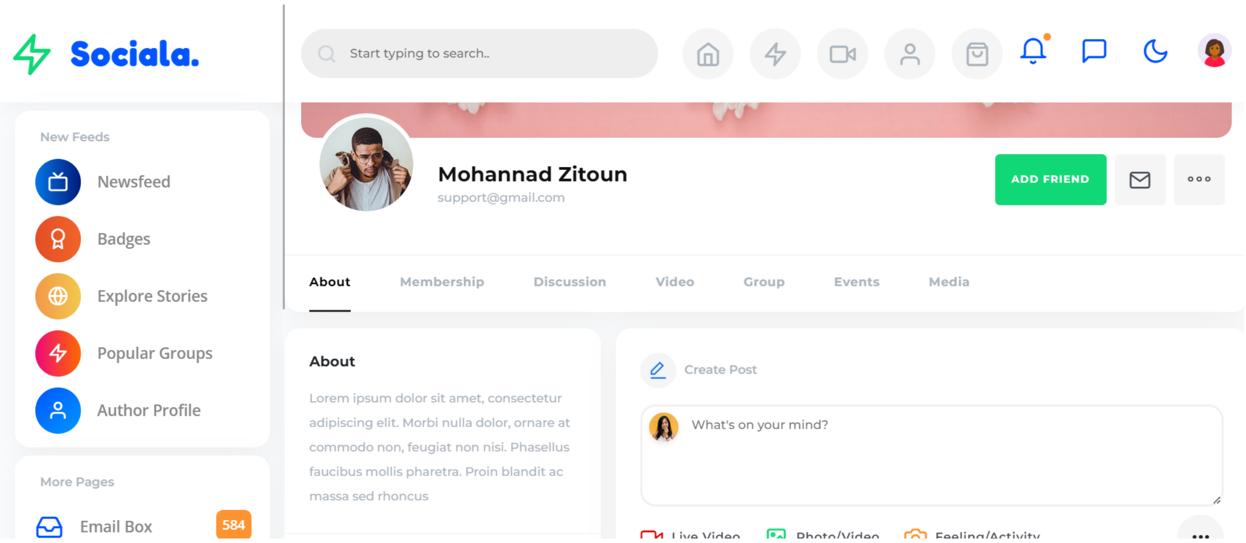
Frontend

In this section, we will design a useful and simple interface that is suitable for a social media, where we can send requests to the end points we created in the backend, and use HTML, CSS, JavaScript and ReactJs. We will add and remove many things according to our own needs through a paid or free template that we will find on the internet. We will choose this template suitable for React and thus we will get a more useful and faster interface.

After designing the pages and creating the components, we will send a request to the currently running Backend using Axios, and the processes returned in the backend will send us a response after the payload is gone.

We will try to develop the interface, which is the only part that will be visible to the user, in a user-friendly, simple, and fast working way.

This is an example of what we aim to use in our project as user interface (frontend):



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi nulla dolor, ornare at commodo non, feugiat non nisi. Phasellus faucibus mollis pharetra. Proin blandit ac massa sed rhoncus

Figure 2. An example template for frontend. [8]

User Service

This service will contain the user's id, personal information, friend information and Steam account data and perform CRUD(create,read,update,delete) operations on them. It will also provide the link with the Recommendation Service. This service will also perform user registration and login processes. In addition, CRUD(create,read,update,delete) operations for the post and comments to the posts will be made in this service for the part where the users can discuss the games. Users can add or delete each other as friends; transactions will take place here.

Technologies to be used:

Spring Boot : Spring Boot is an open source Java-based framework used to create a micro Service. [13]

MySQL : MySQL is one of the most recognizable technologies in the modern big data ecosystem. Often called the most popular database and currently enjoying widespread, effective use regardless of industry, it’s clear that anyone involved with enterprise data or general IT should at least aim for a basic familiarity of MySQL. [14]

Chat Service

This service will enable users to make text, voice and video calls with each other. In addition, channels can be established and chat can be made in the form of groups.

Technologies to be used:

Node.js: As an asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications.[9]

Express.js: Express is a minimalist web framework for Node.js — Express makes it very easy to create and run a web server with Node.[10]

Socket.io: Socket.IO is a library that enables low-latency, bidirectional and event-based communication between a client and a server.[11]

WebRTC: With WebRTC, you can add real-time communication capabilities to your application that works on top of an open standard. It supports video, voice, and generic data to be sent between peers, allowing developers to build powerful voice- and video-communication solutions.[12]

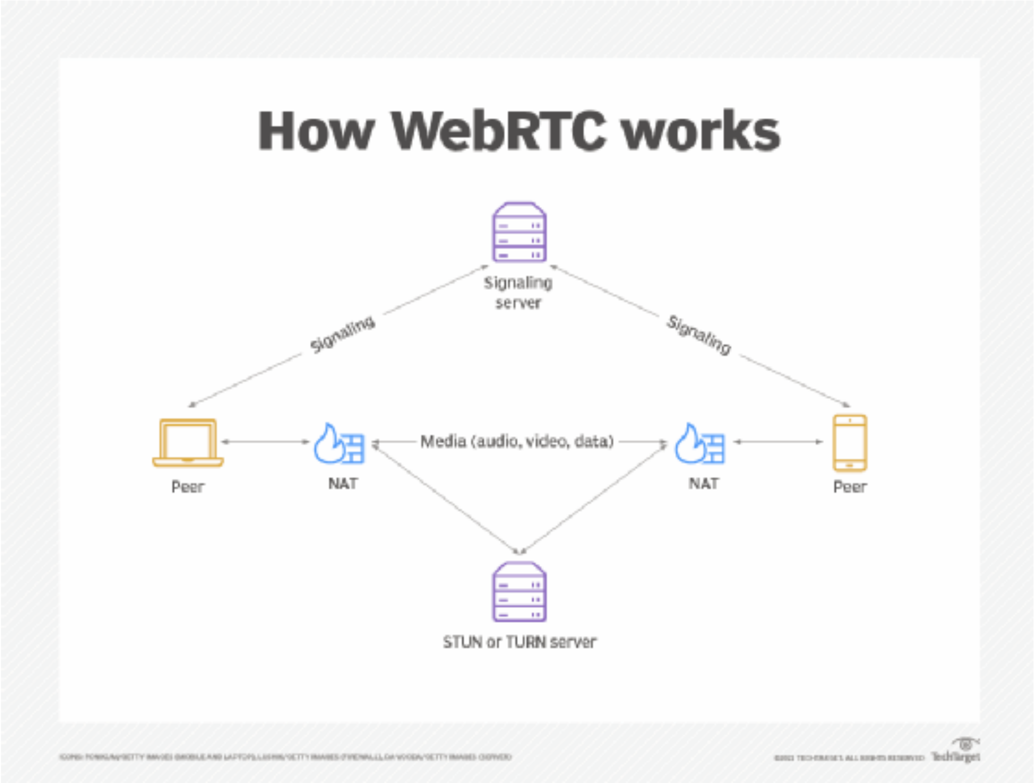


Figure 3. How WebRTC works. [15]

Peer.js: PeerJS allow us to implement WebRTC. [10]

Hybrid Recommendation System

For the recommendation algorithm, we will use the "hybrid recommendation system" as used by large companies like Netflix. This hybrid system consists of a combination of content-based filtering and collaborative filtering as show below.

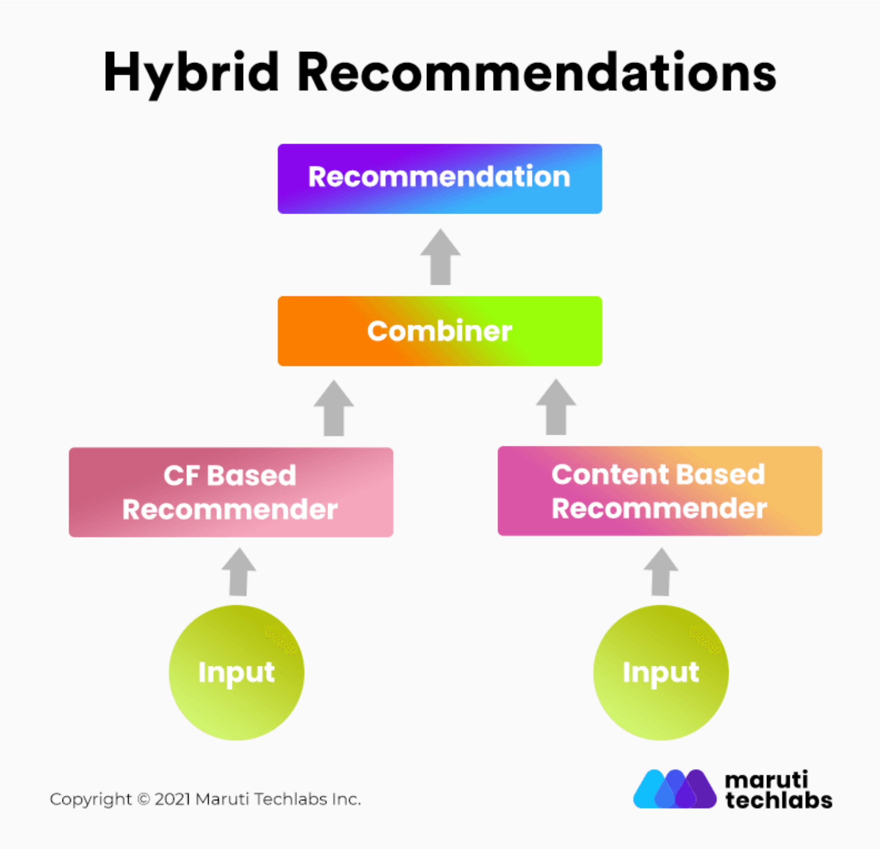


Figure 4. A schema of hybrid recommendation. [6]

Compared to pure collaborative and content-based methods, hybrid methods can provide more accurate recommendations. They can also overcome the common issues in recommendation systems such as cold start and the data paucity troubles. [6]

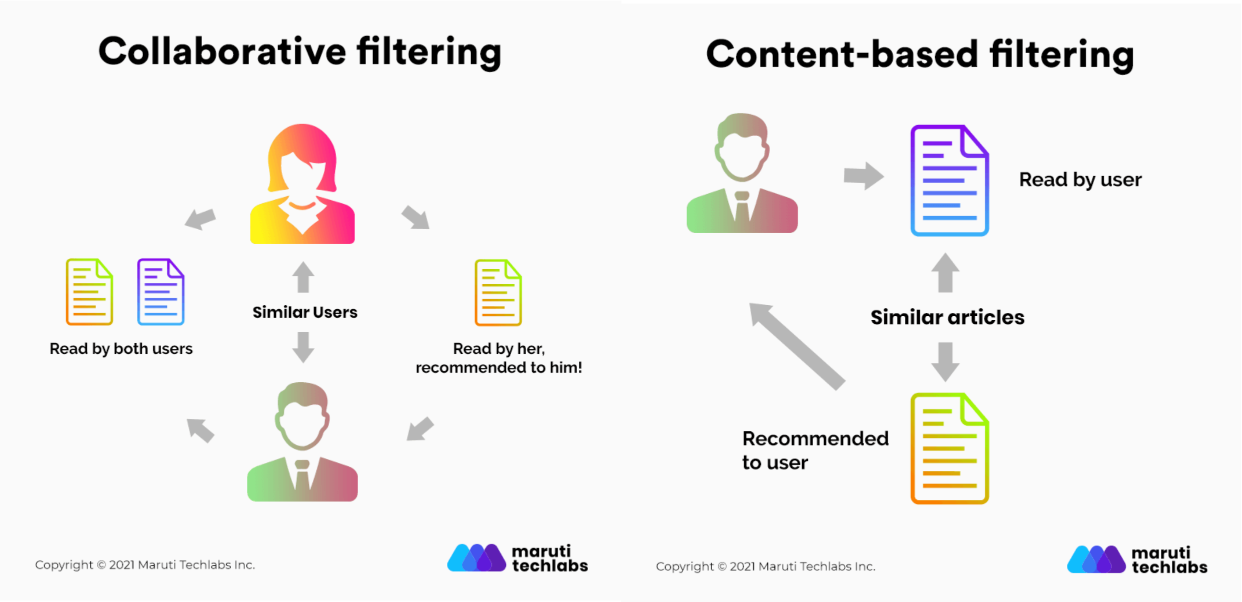


Figure 5. Collaborative and content-based filtering as schema. [6]

The collaborative filtering method is based on collecting and analyzing information based on behaviors, activities, or user preferences and predicting what they will like based on the similarity with other users [6].

We will use the test and training data separately from the dataset. We will also use as much information as we can from the user logging into the platform. At this point, we will ask the user to log in with Steam and authorize the platform. From the data we reach in this way, we will see the games played by the user, how much he played these games.

In addition, cleaning unnecessary columns or rows in the data we will access is one of the things we need to do. One of the most important points is to use only the data we need in order to reach the desired output in the fastest and clearest way.

This technique will customize a general user-game table into user-some features and game-some features and make it more useful and simple. By arriving at a result in this way and combining these results with content-based filtering, we can suggest games that the user does not own and may like.

We have three types of data for collaborative filtering:

- **Explicit data** – users specify how much they liked the game.
- **Implicit data** – if a user plays a game, the system infers that the user is interested.
- **Laten (hidden) Data -** The data we learn from our output data.

In the dataset part, there will be user and game datasets. Things we need in User datasets: user id, game name and time the game was played. This data can show us the level of whether the user likes the game or not. In the game dataset, we can pay attention to data such as how much the game is liked and the number of positive and negative comments. We will reserve some of this data for testing and some for training. There are such datasets on the Internet. Although we have accessed a few of them from sites that provide this data free and legally. In our system, we will use the Matrix Factorization method, as the implicit data will be more intense.

In this part, Alternating Least squares (ALS) will use matrix factorization. ALS is an iterative optimization process where we for every iteration try to arrive closer and closer to a factorized representation of our original data [7]. With these techniques we will make each user and item a vector and this will give us ability to find any similarity and connection between users or items.

We will learn and compare the relationship between games and users. This relationship can be explicit data like the vote given to the game. But since we are unlikely to access this data, we will look at other user interactions with games.

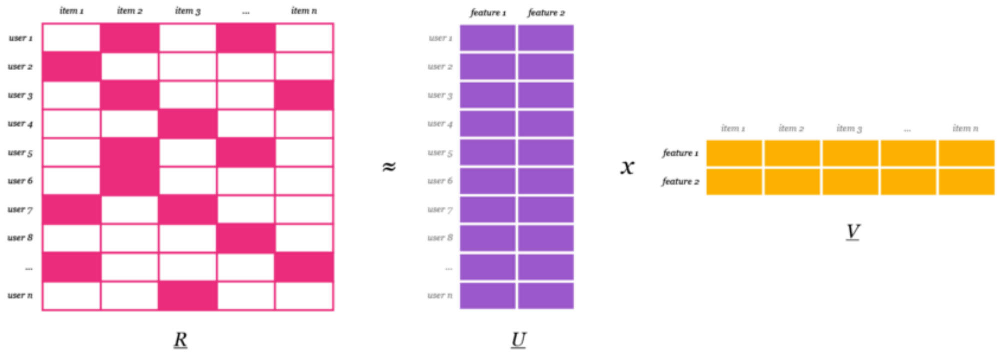


Figure 6. Matrix factorization. [7]

In order for the suggestion system we will use to be hybrid, we need to use content-based filtering as well as collaborative filtering. Content-based filtering methods are mainly based on the description of an item and a profile of the user's preferred choices. In content-based filtering, keywords are used to describe the items, whereas a user profile is built to state the type of item this user likes [6].

In this section, we will try to find a similarity between the games. We'll look at the game's genre, developer, and tags for this similarity. This information may change later in the project. After specifying what we are looking for, we will find similar games for the games the user is playing.

As a result, we will use two systems, collaborative and content-based filtering, and improve the algorithm until we get better results. By doing all of these, we will try to make the most appropriate suggestion for the user.

We will use libraries such as Pandas, Numpy, Implicit to implement these algorithms and methods. These libraries may increase as the project progresses.

Tools that are going to be used:

Tools	**Purposes**
	Implementation languages

Java, Python, Javascript, HTML, CSS Languages	
Github	Version and source control
Spring, Node.js , Flask	Implementation of API’s
Socket.io , Peer.js, Express.js	Implementation of chat service
Pandas, Numpy, Pathlib, Implicit	Implementation of recommendation service
Electron.js , React.js	Implementation of frontend
Zoom	Communication between the supervisor and team members
draw.io	Preparing of diagrams
Trello	Task tracing

7. Professional Considerations

- ***Methodological considerations/engineering standards:***

  - Git is a speed-oriented, distributed version control and source code management system used in software development processes. We will use Git for version control of the source code.
  - JavaScript is a language that enables applications to be developed on both the frontend and backend. We will use JavaScript’s Electron.js, Node.js, React.js frameworks.
  - Python has libraries that make it easy to develop on machine learning and artificial intelligence. Python language will be used for the suggestion system we will prepare. Libraries such as Pandas, Pathlib, Numpy will be used.
  - Java is a suitable language for developing fast and reliable services. We will use Java and Spring Boot framework in our project.
  - A flowchart diagram and other necessary diagrams will be prepared to visualize the progress of the application. For this, draw.io website will be used.
  - Zoom will be used for communication between the supervisor and team members.

- ***Realistic Constraints:***

  Economic:

- As of November 2022, Twitter has a market cap of $41 Billion. According to the most recent data, Instagram’s net worth s estimated to be around $102 billion. These are similar apps to our project.
- There may be server and database costs.

Environmental:

- It is not considered to cause any harm to the environment.

Ethical:

- We can observe users' interactions and show advertisements etc. in the application accordingly.
- We will get users’ Steam profile information.

Health and Safety:

- Since the application we want to do is to show users what they want more clearly and quickly than other platforms, the time they will spend on the platform may be less than other applications. However, mental, and physical diagnosed and known problems may occur.

Sustainability:

- Today, application development and publishing has become easy for all types of people. It is difficult, perhaps impossible, to implement applications against monopolized giants and to ensure that they can continue to live, both in terms of time and economy.
- In our opinion, this idea, which has "similar" but not "same" in the market, has a more solid basis than the ideas that are presented to the market by "copying" applications with millions of users.

Social:

- The platform we intend to build will not exist to separate people, but to gather people with the same interests together.

8. Management Plan

8.1 Phases

- **Phase 1**

Literature survey for related works. Which technologies the platform will be built using, the most suitable machine learning algorithms that can be used etc...

- **Phase 2**

Determining the database structure, creating services and planning the flow. Creation of the overall design.

- **Phase 3**

Building the platform. Completion and linking of backend and frontend codes and converting them to desktop application.

- **Phase 4**

Improvement of the chat section, adding voice and video chat features.

- **Phase 5**

Writing and editing machine learning algorithm.

- **Phase 6**

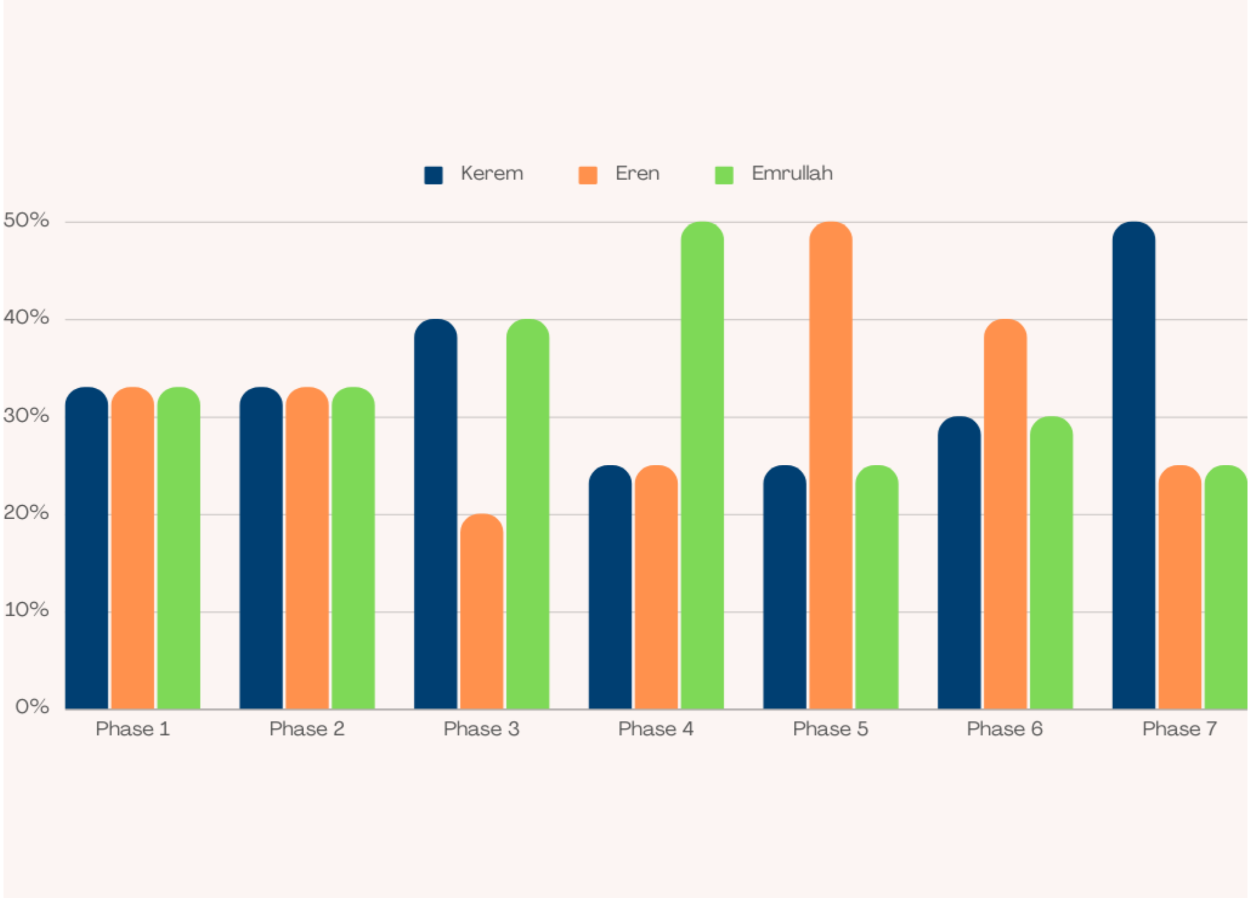
Integration of Machine Learning and social platform parts of the project.

- **Phase 7**

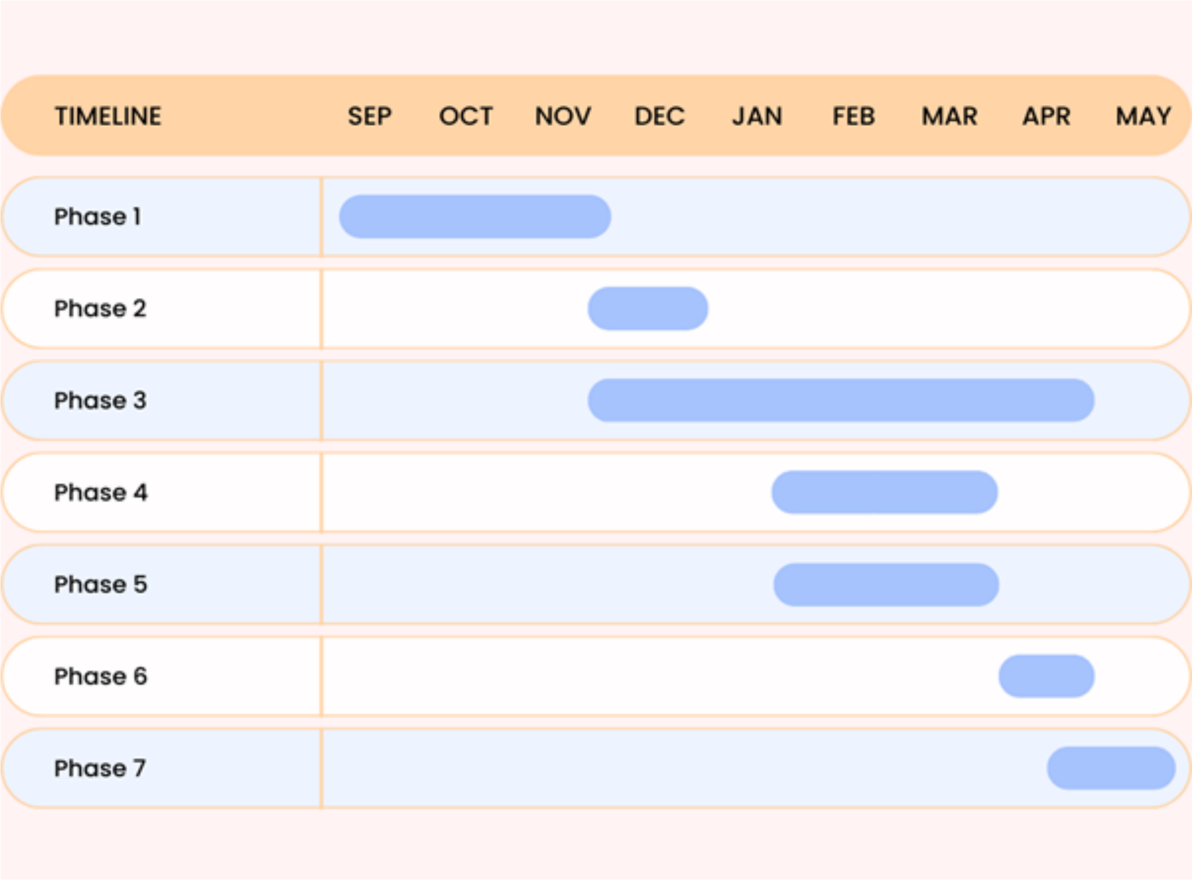
Reviewing and testing deployment options and deploying it so that the entire application can be properly tested. Testing and troubleshooting.

8.2 Division of Responsibilities

In this section, you can see the responsibility sharing within the team. In most places, the division of duties is equal, and in some places, certain people stand out. Kerem undertakes most of the task in the DevOps part, Emrullah in the video chat, and Eren in the machine learning part.



8.3 Timeline



9. Success Factors and Risk Management

A. Success Factors:

(i) Success Factor for Objective 1 : Databases that work properly and are error-free should be prepared. Connections between databases must be established successfully. Private parts (such as password, message) in databases should be encrypted to increase security. Databases should be backed up periodically.

(ii) Success Factor for Objective 2 : Endpoints should work successfully. The concept of Object-Oriented programming must be followed. Token implementation should be done for security. Connections with databases must be successfully established.

(iii) Success Factor for Objective 3 : A user-friendly interface should be prepared. The connection between the front end and the backend must be successfully established. User login, logout, text messaging, audio and video messaging, game recommendation and other features must be present in the interface.

(iv) Success Factor for Objective 4 : In the chat system; users should be able to successfully provide messaging, voice conversation, video conversation functions with each other. Users should not be able to access messages from different users or channels they do not belong to. There shouldn't be a delay bigger than 2 seconds in these functions.

(v) Success Factor for Objective 5 : Datasets about the user, the game and the time spent by the users in the games they play will be used. Datasets must contain at least 5000 data. In addition, in order to test the predictions made, a dataset containing the user and the scores given to the games played by the user will be used. The algorithm we will use should rate games that the user has not rated before, with at least 80% accuracy level, according to the user rating dataset we have. The connection between the main service and the recommendation service must be properly established.

(vi) Success Factor for Objective 6 : The services and interface of the application must be successfully deployed. Then the application should be tried on several different devices and its accuracy should be tested.

B. Risk Management:

- We intend to get the data that we will use to suggest games to the user from the user's Steam account. If we can't get it from Steam, we're considering getting player data from an alternative gaming platform like Epic Games as a plan B.
- Data security: We will take measures to increase security. JWT token will be used for login and this token will be checked in transactions. In addition, passwords and chat messages will be encrypted to increase security.
- Database: The databases will be backed up at regular intervals in case of any negativities in the database.

10. Benefits and Impact of the Project

People who play computer games more will benefit from the application. Thanks to the application, users will be able to find playmates and chat with other users. Users will be able to get game recommendations from the application. There will also be a section where users can discuss games.

- Scientific Impact: We do not aim to write scientific papers as it will be a commercial product.

- Economic/Commercial/Social Impact: As outcome, we expect a commercial product. The platform we intend to build will not exist to separate people, but to gather people with the same interests together.

- Potential Impact on New Projects: The game and therefore the player industry is developing rapidly. Many social network projects can be developed by being inspired by our application.

- Impact on National Security: We do not expect any impact on this issue.

References

[1] "Social Network Sites: Definition, History, and Scholarship" [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1111/j.1083-6101.2007.00393.x . [Accessed: 26-Oct-2022]

[2] "Cold start (recommender systems)" [Online]. Available: https://en.wikipedia.org/wiki/Cold_start_(recommender_systems) . [Accessed: 13-Nov-2022]

[3] "GameTree" [Online]. Available:

https://gametree.me . [Accessed: 25-Nov-2022]

[4]] "Steam" [Online]. Available:

https://store.steampowered.com/?l=turkish/ . [Accessed: 25-Nov-2022]

[5] "Discord" [Online]. Available

https://discord.com . [Accessed: 25-Nov-2022]

[6] "How do recommendation engines work? What are the benefits?" https://marutitech.com/recommendation-engine-benefits/ . [Accessed: 29-Nov-2022]

[7] "ALS Implicit Collaborative Filtering" https://medium.com/radon-dev/als-implicit-collaborative-filtering-5ed653ba39fe . [Accessed: 29-Nov-2022]

[8] "Sociala – Social Network App React Template" https://themeforest.net/item/sociala-social-network-app-react-template/32118267 . [Accessed: 29-Nov-2022]

[9] "About Node.js" [Online]. Available: https://nodejs.org/en/about . [Accessed: 26-Nov-2022]

[10] "Building a Video Chat App with Node.js + Socket.io + WebRTC" [Online]. Available: https://levelup.gitconnected.com/building-a-video-chat-app-with-node-js-socket-io-webrtc-26f46b213017 . [Accessed: 26-Nov-2022]

[11] “What Socket.IO is” [Online]. Available: https://socket.io/docs/v4/ . [Accessed: 26-Nov-2022]

[12] “Real-time communication for the web” [Online]. Available: https://webrtc.org/ . [Accessed: 26-Nov-2022]

[13] “Spring Boot - Introduction” [Online]. Available: https://www.tutorialspoint.com/spring_boot/spring_boot_introduction.htm#:~:text=Spring%20Boot%20is%20an%20open,you%20with%20its%20basic%20concepts . [Accessed: 26-Nov-2022]

[14] “What is MySQL? Everything You Need to Know” [Online]. Available: https://www.talend.com/resources/what-is-mysql . [Accessed: 26-Nov-2022]

[15] “What is WebRTC (Web Real-Time Communications)?” [Online]. Available: https://www.techtarget.com/searchunifiedcommunications/definition/WebRTC-Web-Real-Time-Communications . [Accessed: 26-Nov-2022]