ÇANKAYA UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

CENG 407

PROJECT REPORT

Anıl Mithatcan AKBAŞ	201811001
Başak ÇAVUŞDAĞ	201711014
Berkan Taha ÖZDEMİR	201811402
Ceren AKKUŞ	201711401

Group No: 08

Supervisor: Assist. Prof. Dr. Uğur SOPAOĞLU

CONTENTS

ABSTRACT

Recommender systems are now being used in many areas around the world. Recommender systems appears in every field, from movie, music recommendation to paper recommendation. The number of papers published is increasing day by day and it is increasingly difficult for researchers to find reliable sources that match what they are looking for. Researchers waste far too much time seeking for the right article. The goal of an academic research recommendation system is to save time and introduce them to articles that are related to what they are reading. It is a software tool and approaches that filter tailored information based on the user's preferences from a big volume of data to give suggestions based on the customer's taste in order to discover new relevant things for them. This literature review aims to find the best methodologies to create well-functioning recommendation platform for academic research.

Keywords: recommender system, paper recommendation, recommendation algorithms

ÖZET

Öneri sistemleri artık dünya çapında birçok alanda kullanılmaktadır. Tavsiye sistemleri filmden müzik tavsiyesine, kâğıt tavsiyesine kadar her alanda karşımıza çıkıyor. Yayınlanan makalelerin sayısı her geçen gün artmakta ve araştırmacıların aradıkları şeye uygun güvenilir kaynaklar bulması giderek zorlaşmaktadır. Araştırmacılar doğru makaleyi aramak için çok fazla zaman harcıyorlar. Akademik araştırma öneri sisteminin amacı, zaman kazanmak ve okudukları ile ilgili makaleleri onlara tanıtmaktır. Kullanıcının tercihlerine göre uyarlanmış bilgileri büyük hacimli verilerden filtreleyerek, müşterinin zevkine göre önerilerde bulunmak ve onlar için yeni alakalı şeyler keşfetmek için kullanılan bir yazılım aracı ve yaklaşımlardır. Bu literatür taraması, akademik araştırmalar için iyi işleyen bir öneri platformu oluşturmak için en iyi metodolojileri bulmayı amaçlamaktadır.

Anahtar Kelimeler: öneri sistemi, makale önerisi, öneri algoritmaları

1. Introduction

1.1 Problem Statement

Nowadays, recommender systems are commonly utilized in ecommerce for the purpose of targeted advertising. They propose items that each user is likely to enjoy based on their profile, prior purchase history, and online activities. Amazon.com, for example, suggests comparable goods such as books, while Netflix suggests movies based on a user's preferences. The rapid development of information technology has led to a rapid increase within the volume of digital information.

1.2 Solution Statement

Researchers sharing their research with the aim of exchanging information on a digital platform can lead to information overload. Thus, information pollution also arises, making it difficult for users to be sure of the accuracy of the information. In academic research, recommender systems can provide researchers with accurate and relevant papers in no time.

2. Literature Search

2.1 Content-Based Filtering (CBF)

Content-based filtering (CBF) attempts to recommend items to active users based on the similarity count of the users' past positive reviews. For instance, if users like web pages with the words "cast list", "trailer" and "scene", Content-based filtering will recommend pages related to the movies. Content-based filtering relies heavily on item descriptions and user orientation profiles. There are a lot of ways to building profile. First of all, a researcher's papers collected. For instance, using keywords that the user has already searched for, the user profile can be predictable. Content-based filtering algorithms try to recommend items based on similarity count. Recommend the most matching item by comparing various candidate items with items previously rated by the user. Also, the academic research recommendation systems take keywords from paper's title and parts of paper like abstract and content. It recommends an article from the database by matching the keyword of the user's profile and the keyword from the papers.

Item Representation, Profile Learning, and Recommendation Generation are the three important processes of CBF.

Item Representation: To comprehend how items differ from one another, special attributes are required. These attributes are examined in two categories: Structured attribute and Unstructured attribute. Structured and Unstructured attribute difference is the value of attribute. For the structured attribute, the value of attribute is limited and specific but for the unstructured attribute, it is the opposite. As a result, unstructured attribute can't be utilized for analysis. There is an item representation method which name is TF-IDF model. TF-IDF model (term frequency inverse document frequency) is widely used method in the fields of information retrieval and text mining. DF returns a statistical result based on how often a keyword occurs in the text. This method is a useful way to present similar papers to the user. Also, there is a method which is for create a description of the content of the papers. This method is Key phrase (typically constituted by one to three words) Extraction Model. If the paper has not the keyword session, analysis system finds the most appropriate words from the paper. The key phrase list is a list that helps to understand the main idea, subject of the paper. In contrast to CBF, keyword-based search only focuses on the searched word and doesn't care about the user's other interests. Also, some researchers cannot choose their keywords correctly. Therefore, they cannot get efficient results. That's why it's important to personalize recommendation results. For the CBF, if researcher's interests change, recommendations are formed accordingly. Consequently, the importance of a profile cannot be overstated.

Profile Learning: CBF recommender system builds a profile according to the interests and tastes of the researcher. So, it can determine whether this profile likes the new recommendation or not. Building user profiles may be done in a variety of ways. For example, the LDA method is used to create a profile based on the researcher's previous publications. The researcher profile is organized according to users newly publish papers. In order to make recommendation systems and user profiles more personal, researchers are divided into senior and junior researchers according to the number of papers they publish. All the profile learning methods mentioned are based on researchers' previous publications and activities. For example, the user's interests are discovered from the title or references section of the papers. The system can divide the abstract into two parts to suggest papers from two aspects and to make more specific suggestions to the user: problem description and solution description. In addition, there is another form to represent user profile. Docear arranges users' data in a tree before creating a user model from the user's mind map collection.

Recommendation Generation: To display the most relevant papers to researchers, user profiles and representations of candidate papers have been constructed. A user profile is also builds for these random results. There are two recommendation lists: related papers and unrelated papers. The result list is sorted by the similarity of the user profile and the candidate paper. Sometimes, in order to have a broader perspective and gain new information, researchers can be presented with paper recommendations from more distant topics. There are some disadvantages in the CBF system. For example, it may not detect insufficient

information in junior researchers' paper because it relies on the word analysis technique. Therefore, it cannot provide fully guaranteed information.

2.2 Collaborative Filtering (CF)

CF, just like CBF, makes recommendations on the user's interests. So, it needs to know these interests. CF considers that if two different users are rating on the same common items, those users have similar interests. So, it can be recommended if one user saves something the other user does not. User opinions can be obtained through a survey. CF finds similar user by looking at rating history and uses neighborhood. Compared to the Content-Based Filtering method, CF has some advantages: evaluated based on user ratings, regardless of recommended paper content. Also, since only cross-user similarity is considered, recommendations may be independent of the user's current research. CF mainly contains the two categories of methods: **User-based approach:** Users are the main subject of the system. Similar users are found, and recommendations are made according to their common interests. **Item-based approach:** Relationships between papers are the main subject of the system. Recommendations are made according to the ratings made by the user, assuming that the interests of the user will not be variable. In the user-based filtering, candidate papers are found based on the past preferences of the neighboring user. In the item-based filtering, papers are found based on the user's past preferences.

Although CF is a common method of recommendation, it does have certain drawbacks. The most evident flaw is the cold start problem. Papers that haven't been rated yet aren't recommended till they have Because new users with few ratings on any papers have an empty history, the algorithm will not be able to discover a neighborhood until they have enough ratings.

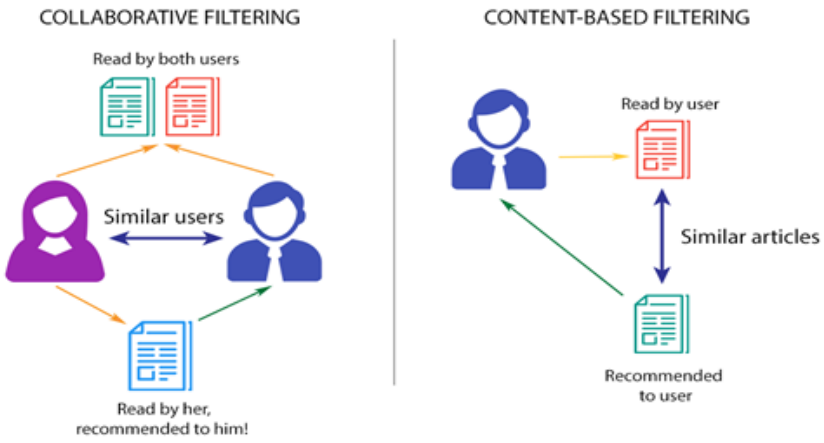

Figure 1: Collaborative Filtering and Content-Based Filtering

2.3 Graph-Based Method (GB)

Graphs are made of nodes and edges; they can be made of several categories of data such as researchers and their social relationships and their papers which involves their citations and list goes on. Recommendation systems use algorithms to look over the graph to find useful data. Graph-Based Method uses more than one source of information to better its accuracy when other methods stick with one or two sources only.

Every graph starts showing its collected data in a mixed form consists of researchers and papers. For example, if certain researchers publish about the same topic and if every researcher and their papers are shown in a graph-based model, we can see a pattern that would be helpful for an algorithm.

The progress of recommending based of GB recommender systems follow two processes which are Graph Construction and Recommendation Generation.

Graph Constructing uses digital information mostly nowadays because most researchers choose to publish and read papers for sharing precious data, with that data graphs can be made. Matrices can be made with knowledge of a researcher and his interest in a paper, with that confirmation papers' coauthors if there are any might publish similar papers with that probability accuracy of recommendation can be increased.

Bi-Relational Graph (BG) is an alternative graph for RSs. BG includes subgraphs of paper similarities, researcher similarities and graph of researchers and papers conjoined together.

Citation Graph (Network) consists of papers and their citation correlations among each other. If papers have mutual references or they use citation to point one another, they are considered alike and that's how RSs use structure of the network. Co-authors connections can be used as an asset to a Citation Graph, and it's called as citation-collaborative network which contains citations, collaborative and author, paper relationships.

Graph based methods doesn't use the content of the paper and the profiles of researchers because their data is not appropriate for using as nodes.

Basic random walk algorithm can't be said as the best method because it jumps to another different vertex for traversing a graph and jumping randomly. Random walk with restart method walks among the neighbor knowing where its base of movement is and that helps with ranking articles such as bipartite network.

Random walk model can be used for cross-domain recommender systems. Such as recommending a certain user their friend candidates with the data of their social relations between each other. Random walk is used on the network of similar candidates. Cross-domain systems use all of these find relations between source domain and a certain domain.

PaperRank is used for computing relations between papers in citation network, it's also used for evaluating papers which don't have anything in common.
In the end Graph-Based method mostly makes use of correlation among nodes.

2.4 Hybrid Method (HM)

Using a few methods is proven to be better choice in couple of areas such as accuracy and performance. Different methods provide more analysis techniques.

Combining Content-Based and Collaborative-Filtering resulted in a mixed bag of advantages and disadvantages for first rater and sparsity issues. CB methods can be used to setting up the researchers' profiles then CF is used to finding out the potential citation papers. Building up a profile requires TF-IDF scheme and finding papers with the highest cosine similarities. Then CF comes into play, paper-citation matrix is used for calculating similarities and the most similar ones are called neighbors in the end.

Other algorithms have risen from combining methods such as CBF Separated which is an algorithm based on CBF difference is it additionally recommends similar lists for its references, and it combines the previous list with the new one and presents only the combined one; CF-CBF Separated algorithm requires CF method to run first then CBF; CBF-CF Parallel algorithm makes use of both CF and CBF in parallel and it combines lists from both in a right order. These HM's are verified to be better than a single method running and other special methods exist and they are better as well, they are CF with latent factor, probabilistic topic, spreading activation etc.

CF with latent factor works with other users' past interests then calculating similarity with targets' interests and this is mostly used for known papers. Spread activation method is used in CB and user-based CF to find targets similarities. EIHI algorithm is suitable for dynamic datasets such as modern world digital papers, this one can be personalized to guarantee quality of the content.

CB and GB combination performs better than usual methods, CB digs the profiles of its users' interest, and the GB uses citations to find similar papers form the graph. CB with citation network can be used to show most related papers from digital datasets.

2.5 Others

Long Short-Term Memory finds out a semantic representation of the papers after CBF is used.

Comparisons of Common Techniques

CB Filtering has advantages such as: Every paper is individually handled for similarity; results are based on users' personal preferences.
Disadvantages: Word relevance quality is uncertain, new users cause problems.
CF has advantages such as: Results might be from by coincidence, quality is assured.
Disadvantages: Cold start problem, sparsity problem.
GB has advantages such as: Uses different sources.
Disadvantages: Doesn't use papers' content or users' interests.

Open Issues and Challenges

Cold Start
When a newly introduced paper or a user pops up in the algorithm, it can cause issues. New user means no back story to compliment the accuracy of algorithm. New papers disappear among popular ones. CB filtering analyses the papers' content so that solves the issue.

Sparsity
It's a common misconception to assume users and the papers are in the same amount or the users are more than papers. The truth in the situation is that papers are a lot more than users and not every paper is rated for users by users. CF method takes a toll with sparsity, not enough sparsity, and similarity among users.

Scalability
Scalability is for a system to work efficiently in an environment. CBF and CF uses static datasets, but datasets grow each moment so it's an issue. EIHI is a newly developed algorithm and its efficient with dynamic datasets.

Privacy
Datasets require users' data and that involves personal information as well. That causes privacy concerned issues, so algorithms need to be developed without invading someone's privacy.

Serendipity
This helps users to find papers with their interests such as younger researchers need to know more about their profession so meanwhile senior researchers need newer data. CF offers serendipity because it doesn't consider the contents but the neighbors while offering papers.

Unified Scholarly Data Standard
There are different sources of academic data. Most of them have their own characters and causes issues.

All these issues with static datasets also have an impact on ever-expanding datasets. This situation is called incremental dataset problem. There are studies developed for this problem, such as the EIHI algorithm and the EIHI-based technique.

INCREMENTAL DATASET

The Efficient Incremental High-Utility Itemset Mining method (EIHI), which is designed to operate with dynamic datasets, was utilized. The EIHI algorithm is used in this proposed solution because it is compatible with dynamic datasets. The tree-based algorithm makes it easier to update the dataset than other algorithms. Academic literature is incremental. When it comes to the continuous update process, it becomes difficult to scan the entire article, so the EIHI algorithm does not rescan the previous dataset.

Related Work

One of the bases of the research paper is makes recommendations based on the citation frequency of an article or the relevance of its content to the user. The method of using semantic data to improve the quality of the recommendation and using the concepts of co-authors and different users to encourage diversity was used. To boost the quality of suggestions even further, a research paper recommender system supporting diversity has been built, which makes recommendations based on the notions of co-authors and dissimilar users. To better respond to the user's personal searches, a recommendation system was created by considering the user's recent searches or interests. To create this system, the user's inputs, or the published article that the user needs can be used.

Proposed Approach

The datasets are based on a two-stage algorithm that limits the efficient use of the incremental nature of the research report pool. To overcome this limitation, the dynamically assisted EIHI-based recommendation approach is preferred. EIHI is used to suggest the most appropriate result for user searches regardless of the increase in articles. The technique performs the overall recommendation process in two steps. 1) To select the publications that are suitable for the researcher's interest. 2) Suggest to researchers with extremely useful articles. They used EIHI to extract HURs to keep it running efficiently even as the amount of research publications in the repository increased (as EIHI can handle dynamic datasets).

Working

The proposed approach works as a two-stage method:

Stage I. Finding papers that are relevant to the user based on their contents:

The dataset is partitioned into 'k' clusters using the PLSA algorithm based on the word distributions of publications and their chances of fitting into a specific cluster in this phase. Each cluster is assigned a distinct subject implicitly and then based on their word distribution.

Following the formation of the clusters, the closeness of each cluster to the user's topic of interest is assessed using similarity metrics.

The PLSA algorithm determines which cluster of papers has the most affinity for the user's topic of interest. When a new paper is added to the repository, the algorithm is rerun to allocate the newly added papers to their corresponding clusters. Only the cluster with the highest relevance to the users' issue is given to the next phase.

Stage II. Identifying Reference-sets of high benefit to the user based on the user's customized requirements:

We may use the date of publication, the publishing authority, and other factors to determine the value of a reference-set. EIHI is a HUIM technique that extracts the most suitable reference sets for user requirements.

The internal utility(i) of a reference can take values of 1 or 0, indicating whether the reference is referenced by a work. The researcher provides the external utility(e) depending on his preferences (which in the presented approach is publishing
date). The utility of a reference (r) in any paper (P) is $u(r, P) = i*e$, where I and e are the internal and external utilities, and the utility of a reference-set (R) is the sum of the utilities of all the references in that set. Usually, all reference sets with minimal benefit are collected for inclusion in the recommendation list. The first 10 reference sets are selected and recommended to the user. In case more articles are added to the repository, the EIHI is applied again to the newly added articles. EIHI is capable to deal with dynamic datasets as one can discover new HURs only by judging newly added articles together pre-found HURs.

Dataset

Network ACL Anthology (a real-world dataset) collects research articles from many places and displays them in the form of a citation network.[41] Because of the consistency of the dataset's focus on computational linguistics, they had to execute the second step of mining HURs from the research paper repository directly. There is no need for clustering to discover research papers that are relevant to the user. Each paper Pi in the citation network is considered a transaction, with its references signifying the objects in the transaction. They must have a network of the "transaction-itemset" type to utilize HUIM techniques. IDs (unique) must be assigned to the papers, and the output only represents the papers with these IDs. Pre-processing of raw data is performed.

The Result of an Exemplary Study

In the case of the Two-Phase approach, if the dataset is increased, the entire operation for generating HURs must be restarted. When updates (increments) are applied to the dataset, the graph that they created shows that the EIHI-based technique is always faster than the

Two-phase based approach. As a result, their suggested method can work efficiently even with dynamic datasets.

2.6 Conclusion

Today, recommendation systems have an important place in obtaining information over the internet. There are four groups for paper recommender systems: content-based filtering, collaborative filtering, graph-based method and hybrid method. If the hybrid recommendation system is compared with a collaborative or content-based system, the recommendation accuracy of the hybrid system is generally higher. This happens because of the absence of understanding of domain dependencies in collaborative filtering and people's preferences in content-based systems. The combination of the two leads to an increase in common knowledge, which contributes to better recommendations. The increase in knowledge makes it especially promising to explore new methods of using content data to extend the underlying collaborative filtering algorithm and using user behavior data to extend content-based algorithms.
In this review, initially, there is an explanation of the logic, advantages, and disadvantages of each technique. To assess the performance of paper recommender systems, the following measures are introduced: Precision, Recall, F-measure, NDCG, MAP, MRR, MAE, and UCOV. This review summarizes the open issues. And then, we explain compatibility of the EIHI method with dynamic datasets. After that, we talked about the feature and convenience of the tree-based algorithm, why they chose the ACL dataset. Also, we talked about the result obtained after the study on these.

The site papers.labml.ai shares some goals and techniques with our project. Both pull and process data from other sources to make suggestions for content appropriate to their users. While reviewing a paper, showing what is similar to that paper makes anyone’s workflow faster because you can simply choose another paper which shares the same areas or crosses interest of the user. Design wise site offers minimalistic user-friendly approach as ours will as well.

Also, Mendeley is important social network for paper recommendation. Mendeley is a reference manager and academic social network for finding, reviewing, and sharing research materials. It is a desktop and online application. Mendeley allows users to collect references from the Web and the UCI database, automatically produce citations and bibliographies, and the system recommends articles depending on what they read. Users can share their findings with other researchers.

3. Software Requirements Specification

3.1 Introduction

The following subsections are an overview of the entire Software Requirements Specification (SRS) document.

3.1.1 Purpose of This Document

The purpose of this document is to offer a detailed description of the system requirements as well as an overview of the project definition. In the form of diagrams, it depicts the system's restrictions as well as its characteristics. There are both user and machine interfaces. This document serves as a starting point for developing the Hypercorrects initial edition.

3.1.2 Scope of This Document

The purpose of the “Hypercorrect” is to provide a website that helping people recommend academic paper based on user’s interests. It is able to search papers according to user’s filter. Users will be able to view current and interesting papers in line with their filters and receive e-mails. Also, users may favorite papers and vote for papers. In addition, the software requires an Internet connection to get and display data. All system data is kept in a database that may be accessed over the internet.

3.2 General Description

3.2.1 Glossary

Abbreviation/Acronym	Definition
PR	Paper recommendation aims to recommend new articles or classical articles that match researchers' interests.
API	Application Programming Interface (API) is a set of definitions created for a software to use its functions defined in another software. API; It can be used for web application, operating system, database, hardware or software library.
BERT	Bidirectional Encoder Representations from Transformers (BERT) is a transformer-based machine learning technique for natural language processing (NLP) pre-training developed by Google.
RS	Recommender system or a recommendation system is a subclass of information filtering system that seeks to predict the “rating” or “preference” a user would give to an item.
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.
SQL	Structured Query Language (SQL) is a standardized programming language that’s used to manage relational databases and perform various operations on the data in them
UC	Use Case (UC) is a software and system engineering term that describes how a user uses a system to accomplish a particular goal. A use case acts as a software modeling technique that defines the features to be implemented and the resolution of any errors that may be encountered.
UML	The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.

Table 1: Acronyms and Abbreviations

3.2.2 User Characteristics

There are three types of users that interact with the system: admin, non-member, member. Each of these three types of users has different use of the system so each of them has their own requirements.

The admin may edit site settings.

The non-members may sign up to the system. Also, the non-members may browse trending papers on the homepage and view papers by filtering them. The non-members can view the paper's PDF and GitHub links if exist.

The member, in addition the simple filters on the homepage, s/he can see the papers that may be of interest by doing a special filtering. These filters include how many papers you want to view and interest rate similarity. The member also receives e-mail about newly published related article that may be of interest to the user. The members can vote on the papers and add them to their favorites. The members can view the paper's PDF and GitHub links if exist.

3.2.3 Product Perspective

Hardware

This website is compatible with all smartphones, tablets, and computers. To connect to the Hypercorrect, these devices must have an internet connection.

3.2.4 Overview of Functional Requirements

Sign Up

Req001

The system shall check the necessary field whether user enter or not.

Req002

The system shall check whether the entered information is correct.

Req003

The system shall send a confirmation e-mail when the user is registering.

Req004

The system shall allow the user's subscription when the mail is confirmed.

Req005

The system shall inform the user whether the user sign up successfully or not.

Req006

The system shall allocate space in the database for the user's data.

Login

Req007

The system shall control password and email properly.

Req008

The system shall inform the user whether the user log in successfully or not.

Logout

Req009

The system shall inform the user whether the user log out successfully or not.

Edit Profile

Req010

The system shall allow the user to change profile information and filters.

Req011

The system shall inform the user whether the saving is successfully or not.

Req012

The system shall update the database.

Receive Mail

Req013

The system shall allow the user take e-mail about papers that may be of interest to them.

Show the Related Papers

Req014

The system shall show the related papers.

Vote the Papers

Req015

The system shall allow the members to vote the papers.

Req016

The system shall find the average star.

Req017

The system shall keep the average star in the database.

Show the Papers You May Like

Req018

The system shall show the papers that recommended by e-mail to user.

Show the Trending Research

Req019

The system shall show the trending research by recent reading.

Filter the Papers

Req020

The system shall allow the users to filter the papers.

Req021

The system shall show the papers by using user's filters.

Add to Favorites

Req022

The system shall allow the members that add to favorites the papers.

Req023

The system shall show the papers in member's My Favorites Page.

Req024

The system shall keep the favorite papers in the database.

Show the Code

Req025

The system shall redirect the GitHub link.

View PDF

Req026

The system shall redirect the paper's PDF.

Set the Similarity Rate

Req027

The system shall allow the users to set the similarity rate for recommendation.

Req028

The system shall recommend papers by using similarity rate.

Req029

The system shall keep the user's similarity rate in the database.

Set the Maximum Number of Papers

Req030

The system shall allow the user to set the maximum number of papers for recommendation.

Req031

The system shall show the papers as many as the user wants.

Req032

The system shall keep the user's number of papers setting in the database.

3.2.5 General Constraints and Assumptions

Constraints

Python will be to gather and preprocess data. HTML, CSS, JavaScript and React will be used for the front-end part. Java and Spring Boot Framework will be used for the back end. Also, for recommendation system, Python will be used. The pretrained BERT model will be used to express the texts as vectors. Postgres will be used for data storage. Cosine similarity metric will be used to compare texts with each other.

Assumptions

The user must have at least one of the devices such as tablet, smartphone, or computer.

User's device must have an internet connection.

User's device must have pdf viewer software.

3.3 Overall Description

3.3.1 Interface Requirements

3.3.1.1 Hardware Interfaces

A server is required to host the website and store the data in the database. Additionally, both the devices and the server must have access to the internet.

3.3.1.2 Software Interfaces

Users cannot see the system's software product, which is the network software between the database and Hypercorrect.

3.3.1.3 Communication Interfaces

The server will be connected to the internet through the Wi-Fi or 4G.

3.3.2 Detailed Description of Functional Requirements

Sign Up

Inside the Sign-Up page, the system sends a form containing the information required for users to sign up. This form includes e-mail and password. There are separate labels for each of this information. The user enters her/his e-mail address. In the meantime, the system checks the e-mail address from the database. If this address is not registered in the database, it goes to homepage again but as a member. If this address is registered in the database, the system shows an error message to the user in the form of a message box. This message is as follows “This e-mail address has been used!”. The system checks after the user enters the password. When there is a missing requirement, it issues an error message to the user with a message box. The message reads, "Invalid password, missing requirement.". The system performs all control operations within 2 seconds at most. A possible systemic error etc. In case of malfunctions, an error message is broadcast by the system and the user is directed to the homepage within 2 seconds. If the reason for this failure is internet outage, the system sends a warning message to the user, "Check your internet connection".

Login

In the Login page, there is 2 different labels for email and password. The user enters her/his e-mail address and password. The system controls if the e-mail address and password match those in the database. If they are matching, the system shows a message box with “Login successful” in it and user can view her/his profile page. If they are not matching for the third time, the system shows a message box with “Did you forget your password?” in it and 2 separate buttons “Yes” and “No”. If the user chooses “No” button, the system turns back the page again. If the user choose “Yes” button, the system assigns new random password to user and sends this password to user’s email address. The system performs all control operations within 2 seconds at most. A possible systemic error etc. In case of malfunctions, an error message is broadcast by the system and the user is directed to the homepage within 2 seconds. If the reason for this failure is internet outage, the system sends a warning message to the user, "Check your internet connection".

Logout

When the user clicks the “Logout” button, the system shows a message box “Logout Successfully” and the system turn to the home page. The system performs all control operations within 2 seconds at most. A possible systemic error etc. In case of malfunctions, an error message is broadcast by the system and the user is directed to the homepage within 2 seconds. If the reason for this failure is internet outage, the system sends a warning message to the user, "Check your internet connection".

Edit Profile

In the Profile Page, there are four parts: Notification Preferences, General Details, Recommendation Preferences and Password. If user opens the sending emails, the system saves this choice and enables sending emails about papers that may be interest to the user. If the user clicks the “Change Name” button, the system changes name in the database and shows message box which says, “Name successfully changed.”. If the user clicks the “Change Email” button, the system changes email in the database and shows message box which says, “Email successfully changed.”. If the user clicks the “Change Password” button, the system changes password in the database and shows message box which says, “Password successfully changed.”. If the user changes the recommendation preferences, the system saves them in the database and changes the recommended papers in line with these options. The system performs all control operations within 2 seconds at most. A possible systemic error etc. In case of malfunctions, an error message is broadcast by the system and the user is directed to the homepage within 2 seconds. If the reason for this failure is internet outage, the system sends a warning message to the user, "Check your internet connection".

Receive E-mail

The system considers the similarities between the user's interests and the tasks of the papers and send e-mails about current papers. Also, it considers the number and similarity rate desired by the user and applies filtering. Repeats these events as new papers arrive.

Show the Related Papers

In the Paper Detail Page, the system shows to user similar papers with the same contents. The system performs all control operations within 2 seconds at most. A possible systemic error etc. In case of malfunctions, an error message is broadcast by the system and the user is directed to the homepage within 2 seconds. If the reason for this failure is internet outage, the system sends a warning message to the user, "Check your internet connection".

Vote the Papers

In each paper preview, the system shows the stars which representing the paper voting. If the user is not a member, the system will not allow voting. It shows a message box which says, "Become a member" and if user clicks this text, the system redirects to Sign Up Page. If the user is a member, the system calculates the average vote and rearranges the stars. The system saves new voting result to database. The system performs all control operations within 2 seconds at most. A possible systemic error etc. In case of malfunctions, an error message is broadcast by the system and the user is directed to the homepage within 2 seconds. If the reason for this failure is internet outage, the system sends a warning message to the user, "Check your internet connection".

Show the Papers You May Like

In the Papers You May Like Page, the system shows the papers that user may like. The similarities between the user's interests and the tasks of the papers are the main basis. Also, it considers the number and similarity rate desired by the user and applies filtering. It displays the number of papers set by the user. Besides, the system makes recommendations according to the similarity rate determined by the user. Repeats these events as user enters the Papers You May Like Page. The system performs all control operations within 2 seconds at most. A possible systemic error etc. In case of malfunctions, an error message is broadcast by the system and the user is directed to the homepage within 2 seconds. If the reason for this failure is internet outage, the system sends a warning message to the user, "Check your internet connection".

Show the Trending Research

In the Home Page, the system shows the most clicked papers recently. The system performs all control operations within 2 seconds at most. A possible systemic error etc. In case of malfunctions, an error message is broadcast by the system and the user is directed to the homepage within 2 seconds. If the reason for this failure is internet outage, the system sends a warning message to the user, "Check your internet connection".

Filter the Papers

In the Home Page, users can apply some filters such as best match or newest, filter by year or filter by GitHub. The system should display users with papers that match these filters and match the words they entered. And then, the system redirects them to the Searching Page. The system performs all control operations within 2 seconds at most. A possible systemic error etc. In case of malfunctions, an error message is broadcast by the system and the user is directed to the homepage within 2 seconds. If the reason for this failure is internet outage, the system sends a warning message to the user, "Check your internet connection".

Add to Favorites

In each paper preview, there is “Add to Favorites” button. If the user is not a member, the system will not allow to adding favorites. The system redirects non-members to Sign Up Page if they click the add to favorites button. If the user is a member, the system updates the database and add to user’s My Favorites Page. In the My Favorites Page, there is “Delete” button. If the user clicks this button, the system deletes this paper from user’s favorites and updates the database. The system performs all control operations within 2 seconds at most. A possible systemic error etc. In case of malfunctions, an error message is broadcast by the system and the user is directed to the homepage within 2 seconds. If the reason for this failure is internet outage, the system sends a warning message to the user, "Check your internet connection".

Show the Code

In each paper preview, there is GitHub button. If the user clicks this button, the system redirects user to GitHub page. The system performs all control operations within 2 seconds at most. A possible systemic error etc. In case of malfunctions, an error message is broadcast by the system and the user is directed to the homepage within 2 seconds. If the reason for this failure is internet outage, the system sends a warning message to the user, "Check your internet connection".

View PDF

In each paper preview, there is “View PDF” button. If the user clicks this button, the system redirects user to PDF. The system performs all control operations within 2 seconds at most. A possible systemic error etc. In case of malfunctions, an error message is broadcast by the system and the user is directed to the homepage within 2 seconds. If the reason for this failure is internet outage, the system sends a warning message to the user, "Check your internet connection".

3.3.3 Non-Functional Requirements

Usability

Hypercorrect will be evaluated in this part for its readability, learnability, operability and attractiveness.

Req033

The error messages shall include enough explanation.

Req034

There shall be relevance icons of buttons and function of buttons to facilitate to understand of users.

Req035

Hypercorrect should work in almost every web browser.

Reliability

One of the criteria used to assess quality is reliability. The system must be tested during the development phase to ensure stable software. It must be delivered on schedule and in accordance with the specifications mentioned in this document.

Req036

The system should be available always.

Req037

System should display informative messages when its components doesn’t work properly.

Performance

Req038

All the functions should be performed at most 2 seconds.

Req039

There will be large amount of information to be handled in database such as profile information and the server will be enough space to handle this occupation.

Supportability

Because change is unavoidable in today's environment, these innovations must be built to accommodate any future updates.

Req040

Design elements should be documented well.

Safety

Req041

There should be backup disks for all data, which must be synced on a regular basis to avoid data loss on the database in the event of a disaster.

Security

Req042

System requires email confirmation when registering.

3.4 Analysis - UML

3.4.1 Use Case Diagram

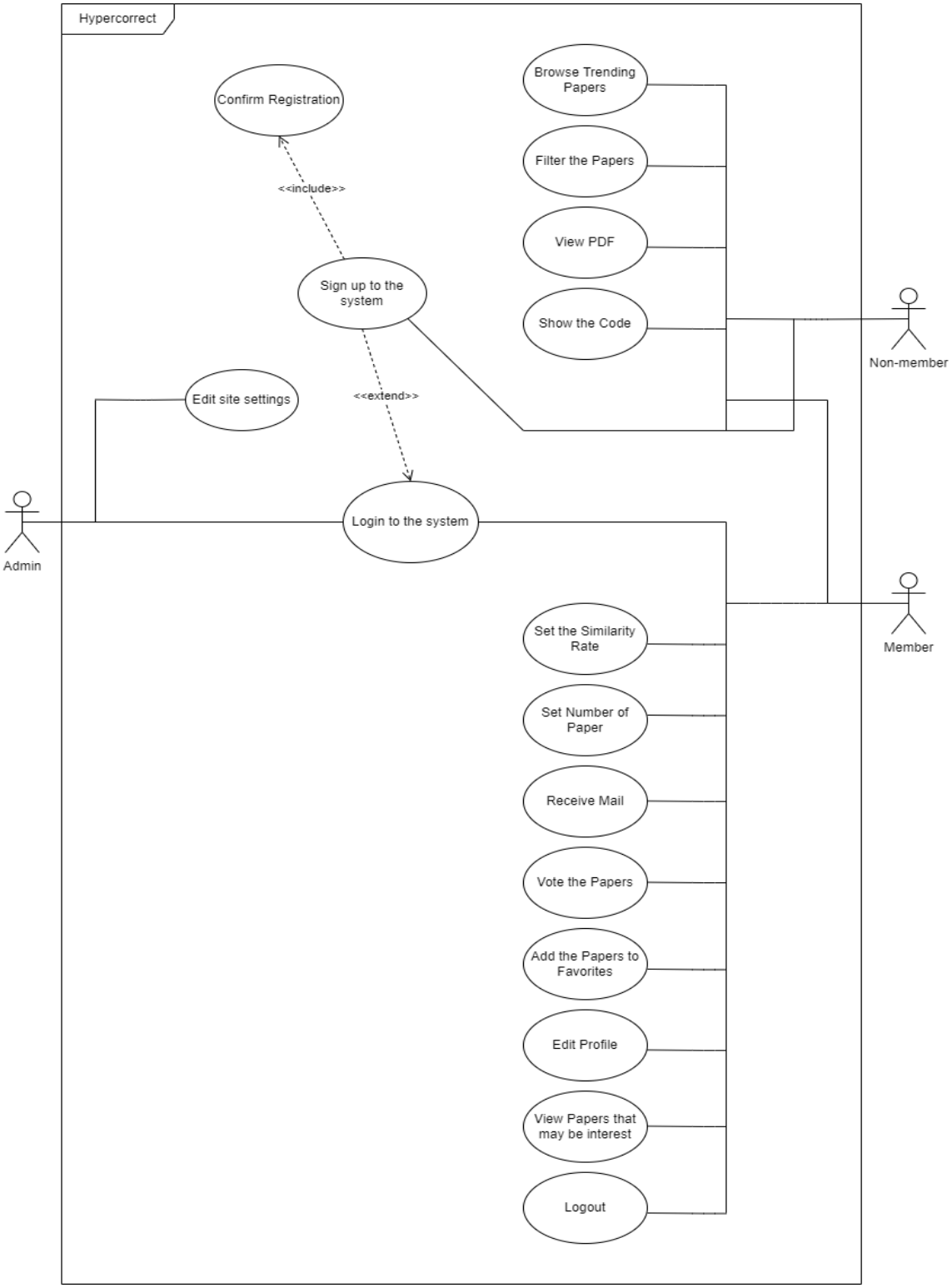

Figure 2: Use-case Diagram

3.4.2 Use Cases

Use-case ID:	UC-001
Use-case Name:	Sign Up to the System
Actors:	Non-Member
Description:	Non-members must sign up to the system for being a user.
Trigger:	Non-member clicks “Sign Up” button.
Preconditions:	User is not a member of system.
Post conditions:	Non-member becomes a member of the system.
Normal Flow:	1. Non-member clicks on “Sign Up” button. 2. Non-members enter her/his necessary information and password for signing up. 3. A confirmation email is sent to the user. 4. User confirms email. 5. A new account is created for the non-member.
Alternative Flow:	System checks the necessary information is valid. -If the information is valid, then go to step 3. -If the information is invalid, the system shows the error message and go to step 2 again. System checks the confirmation. -If the mail is confirmed, then go to step 5. -If the mail is not confirmed, the system shows the error message and go to step 2 again.
Exceptions:	If the non-member enters invalid or not completed information, the system redirects the non-member to the Sign Up page.
Priority:	High
Frequency of Use:	Every time a non-member wants to sign up.
Assumptions:	If the non-member doesn’t enter the necessary information, s/he cannot be a member.

Table 2: SignUpToTheSystem

Use-case ID:	UC-002
Use-case Name:	Login
Actors:	Member
Description:	User who completes the registration process, can login to the system.
Trigger:	This use case is activated when user enters e-mail and password and clicks Login button.
Preconditions:	User must have already registered.
Post conditions:	User logins to the system successfully.
Normal Flow:	1. User clicks on “Login” button. 2. User enters his/her credentials in the are provided for that. 3. System checks if the entered login parameters are valid. 4. System creates a new session for the user.
Alternative Flow:	-
Exceptions:	If the user provides invalid login information, the system notifies the user and redirects him to login section.
Priority:	High
Frequency of Use:	Every time a user wants to login.
Assumptions:	If the user doesn’t enter the necessary information correctly, s/he cannot login.

Table 3: LoginToTheSystem

Use-case ID:	UC-003
Use-case Name:	Edit Profile
Actors:	Member
Description:	Member can change his/her settings such as name, password etc.
Trigger:	Member clicks “Profile” button.
Preconditions:	User must have already logged in.
Post conditions:	Settings are updated according to selections.
Normal Flow:	1. User clicks “Profile” button. 2. User choose the item that s/he wants to change its settings. 3. User clicks “Change” button. 4. System shows successful message.
Alternative Flow:	If a user gives up editing profile, s/he can cancel this process by clicking “Cancel” button.
Exceptions:	-
Priority:	High
Frequency of Use:	Every time a user wants to edit profile.
Assumptions:	User must be registered to the system. User must login to the system with the right password.

Table 4: EditProfile

Use-case ID:	UC-004
Use-case Name:	Vote the Papers
Actors:	Member
Description:	Members can vote the papers.
Trigger:	Member clicks stars.
Preconditions:	User must have already logged in.
Post conditions:	Stars are updated according to selections.
Normal Flow:	1. User clicks stars. 2. The system finds average star. 3. The system shows new average star.
Alternative Flow:	-
Exceptions:	-
Priority:	Medium
Frequency of Use:	Every time a user wants to vote a paper.
Assumptions:	User must be registered to the system. User must login to the system with the right password.

Table 5: VoteThePapers

Use-case ID:	UC-005
Use-case Name:	Add to Favorites
Actors:	Member
Description:	Members can add papers the favorites.
Trigger:	Member clicks “Add to Favorites” button.
Preconditions:	User must have already logged in.
Post conditions:	Papers are added to user’s favorites.
Normal Flow:	1. User clicks “Add to Favorites” button. 2. The system shows successful message.
Alternative Flow:	-
Exceptions:	-
Priority:	Medium
Frequency of Use:	Every time a user wants to add paper the favorites.
Assumptions:	User must be registered to the system. User must login to the system with the right password.

Table 6: AddToFavorites

Use-case ID:	UC-006
Use-case Name:	Filter the Papers
Actors:	Non-member, Member
Description:	Users can filter the papers.
Trigger:	User enters the filters s/he wants.
Preconditions:	-
Post conditions:	Papers are shown by filtering.
Normal Flow:	1. User enters the filters. 2. User can see papers that match her/his filters.
Alternative Flow:	1. User search with words only. 2. User can see papers that match the default filters.
Exceptions:	-
Priority:	High
Frequency of Use:	Every time a user applies filters.
Assumptions:	-

Table 7: FilterThePapers

Use-case ID:	UC-007
Use-case Name:	Show the Code
Actors:	Non-member, Member
Description:	Users can see the code.
Trigger:	User clicks “Show the Code” button.
Preconditions:	-
Post conditions:	User will be redirected to the GitHub.
Normal Flow:	1. User clicks “Show the Code” button. 2. The system redirects the user to the GitHub link.
Alternative Flow:	-
Exceptions:	If the link is deleted or there is a problem with the redirect, the system warns the user.
Priority:	High
Frequency of Use:	Every time a user wants to see the code.
Assumptions:	-

Table 8: ShowTheCode

Use-case ID:	UC-008
Use-case Name:	View PDF
Actors:	Non-member, Member
Description:	Users can see the PDF.
Trigger:	User clicks “View PDF” button.
Preconditions:	-
Post conditions:	User will be redirected to the PDF.
Normal Flow:	1. User clicks “View PDF” button. 2. The system redirects the user to the PDF.
Alternative Flow:	-
Exceptions:	If the PDF is deleted or there is a problem with the redirect, the system warns the user.
Priority:	High
Frequency of Use:	Every time a user wants to see the PDF.
Assumptions:	-

Table 9: ViewPDF

Use-case ID:	UC-009
Use-case Name:	Receive Mail
Actors:	Member
Description:	Members can receive e-mails about current and interesting papers.
Trigger:	User turns on the option to receive email.
Preconditions:	User must be registered to the system.
Post conditions:	User starts receiving e-mail.
Normal Flow:	1. User turns on the option to receive email. 2. User applies filters.
Alternative Flow:	1. User turns on the option to receive email. 2. User can see papers that match the default filters.
Exceptions:	-
Priority:	High
Frequency of Use:	Every time a user wants to receive e-mail and there is an paper or current paper that matches the user’s interest.
Assumptions:	-

Table 10: ReceiveMail

Use-case ID:	UC-010
Use-case Name:	Set the Similarity Rate
Actors:	Member
Description:	Member can set the similarity rate for recommendation.
Trigger:	User sets the similarity rate on the Profile page.
Preconditions:	User must have already logged in.
Post conditions:	Papers are shown by filtering.
Normal Flow:	1. User clicks “Profile” button. 2. User sets the similarity rate. 3. User can see papers that match her/his filters.
Alternative Flow:	User can see papers that match the default filters.
Exceptions:	-
Priority:	High
Frequency of Use:	Every time a user wants to set the similarity rate.
Assumptions:	User must be registered to the system. User must login to the system with the right password.

Table 11: SetTheSimilarityRate

Use-case ID:	UC-011
Use-case Name:	Set the Number of Papers
Actors:	Member
Description:	Member can set the maximum number of papers for recommendation.
Trigger:	User sets the maximum number of papers on the Profile page.
Preconditions:	User must have already logged in.
Post conditions:	Papers are shown by filtering.
Normal Flow:	1. User clicks “Profile” button. 2. User sets the maximum number of papers. 3. User can see papers that match her/his filters.
Alternative Flow:	User can see papers that match the default filters.
Exceptions:	-
Priority:	High
Frequency of Use:	Every time a user wants to set the maximum number of papers.
Assumptions:	User must be registered to the system. User must login to the system with the right password.

Table 12: SetTheNumberOfPapers

Use-case ID:	UC-012
Use-case Name:	Logout
Actors:	Member
Description:	Member can logout.
Trigger:	User clicks “Logout” button.
Preconditions:	User must have already logged in.
Post conditions:	User will be redirected to the login page.
Normal Flow:	1. User clicks on “Logout” button. 2. System shows successful message. 3. System redirects the user to login page.
Alternative Flow:	-
Exceptions:	-
Priority:	High
Frequency of Use:	Every time a user wants to logout.
Assumptions:	User must be registered to the system. User must login to the system with the right password.

Table 13: Logout

3.4.3 Functional Modeling (DFD)

3.4.3.1 DFD Diagram

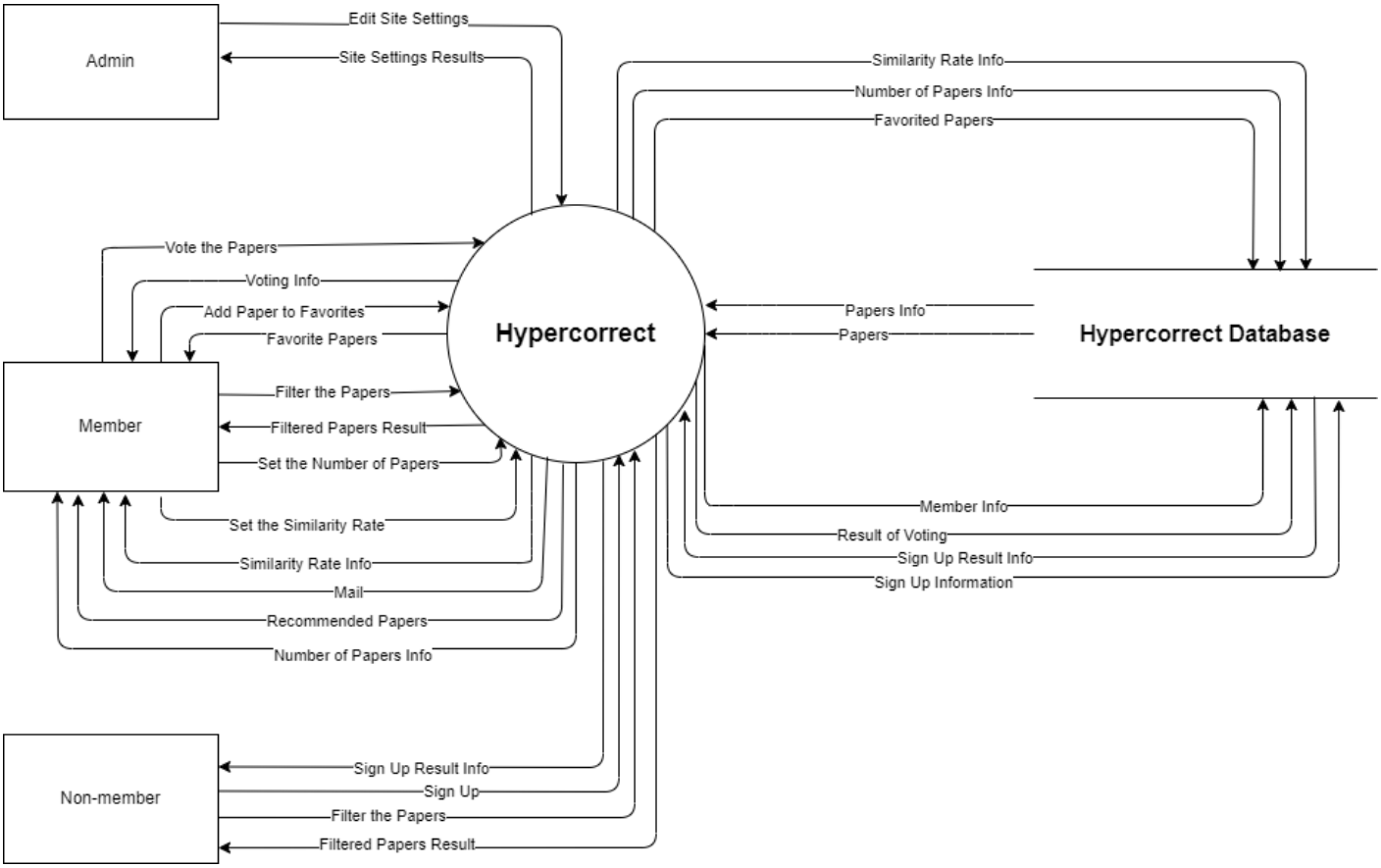

Figure 3: DFD Diagram

3.4.3.2 Data Dictionary

Access	Type	Name	Description
private	string	userName	User's name.
private	string	mailAddress	User's mail address.
private	string	password	User's password.
private	int	userID	User's unique ID.
private	string	role	User's role. (Admin, member)
private	int	paperNumber	User's filter for recommendation.
private	double	similarityRate	User's filter for recommendation.

Table 14: User

Access	Type	Name	Description
private	string	paperTitle	Papers's title.
private	string	paperAuthors	Paper's authors.
private	date	paperDate	Paper's publish date.
private	int	paperID	Paper's unique ID.
private	string	GithubLink	Paper's GitHub link.
private	string	paperPDF	Paper's PDF link.
private	double	starRate	Paper's rate.
private	string	paperAbstract	Paper's abstract.
private	int	viewCounter	Paper's click counter.
private	Int array	task_types	Connects task and paper.

Table 15: Paper

Access	Type	Name	Description
private	int	taskID	Task’s unique ID.
private	string	taskName	Task’s name.
private	string	description	Task’s description.
private	int	areaType	Connects task and area.

Table 16: Task

Access	Type	Name	Description
private	int	areaID	Area’s unique ID.
private	string	areaName	Area’s name.

Table 17: Area

4. Software Design Description

4.1 Introduction

The following subsections are an overview of the entire Software Design Description (SDD) document.

4.1.1 Purpose of This Document

The purpose of this paper is to detail the features and criteria of the "Hypercorrect". It will also provide a sample program interface, which will give you an idea of how the result will look. Furthermore, the steps that will be followed during the implementation will be thoroughly discussed. As a result, the reader of this SDD report will at least have a general understanding of the project.

4.1.2 Overview

Below is a list of the remaining sections and their contents.

The System Overview section explains the overall structure of the system as well as generic requirements.

Architectural Design is the third section, and it outlines the project development process. It also includes the system's class diagram and activity diagram, as well as the simulation's architectural design, which details actors, exceptions, basic sequences, priorities, pre-conditions, and post-conditions. In this part, a block diagram of the system is exhibited and discussed, which is developed according to the use cases in the SRS paper.

The fourth section is on the environment. We've displayed example frames of environment from the prototype and detailed the situation in this part.

The fifth section focuses on demonstrating the link between requirements and other artifacts.

4.1.3 Definitions and Acronyms

Abbreviation/Acronym	Definition
PR	Paper recommendation aims to recommend new articles or classical articles that match researchers' interests.
API	Application Programming Interface (API) is a set of definitions created for a software to use its functions defined in another software. API; It can be used for web application, operating system, database, hardware or software library.
BERT	Bidirectional Encoder Representations from Transformers (BERT) is a transformer-based machine learning technique for natural language processing (NLP) pre-training developed by Google.
RS	Recommender system or a recommendation system is a subclass of information filtering system that seeks to predict the “rating” or “preference” a user would give to an item.
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.
SQL	Structured Query Language (SQL) is a standardized programming language that’s used to manage relational databases and perform various operations on the data in them
UC	Use Case (UC) is a software and system engineering term that describes how a user uses a system to accomplish a particular goal. A use case acts as a software modeling technique that defines the features to be implemented and the resolution of any errors that may be encountered.
UML	The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.

Table 18: Definitions and Acronyms

4.2 System Overview

The system's overall design is based on both functional and non-functional criteria.

We examined best practices and valuable use cases in the functional requirements section and implemented them throughout the project. Then, within the framework of the primary structure and principles we defined in this part, we built the necessary diagrams for the project.

To keep running, our system relies on backend and frontend programs and structures. Python will be used to gather and preprocess data and machine learning tasks. HTML, CSS, JavaScript and React will be used for the front-end part. Java and Spring Boot Framework will be used for the back end. Postgres will be used for data storage. While pretrained BERT model will be used to express the texts as vectors, cosine similarity metric will be used to compare texts with each other.

4.3 System Design

4.3.1 Architectural Design

Client-server architecture is appropriate for our system because it is a web-based solution.

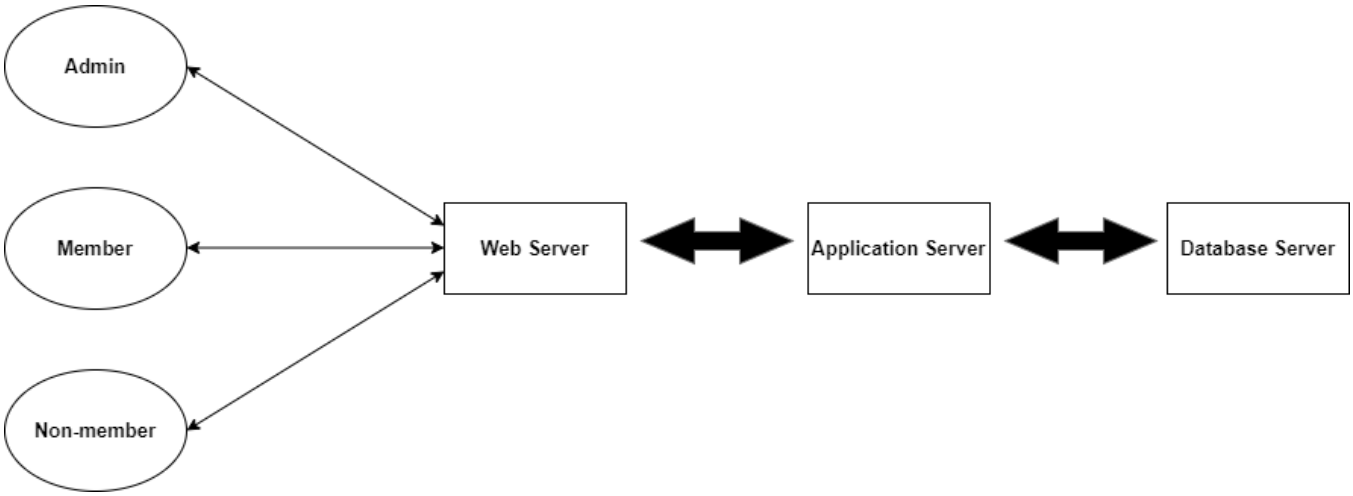

Figure 4: Architectural Design

Web Server: The web server's primary goal is to store, process, and distribute web pages to users, as well as to display website content. The Hypertext Transfer Protocol (HTTP) is used for this intercommunication.

Application Server: A server that hosts apps and is used to execute web applications and process data supplied from another server is known as an application server.

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.

4.3.2 Decomposition Design

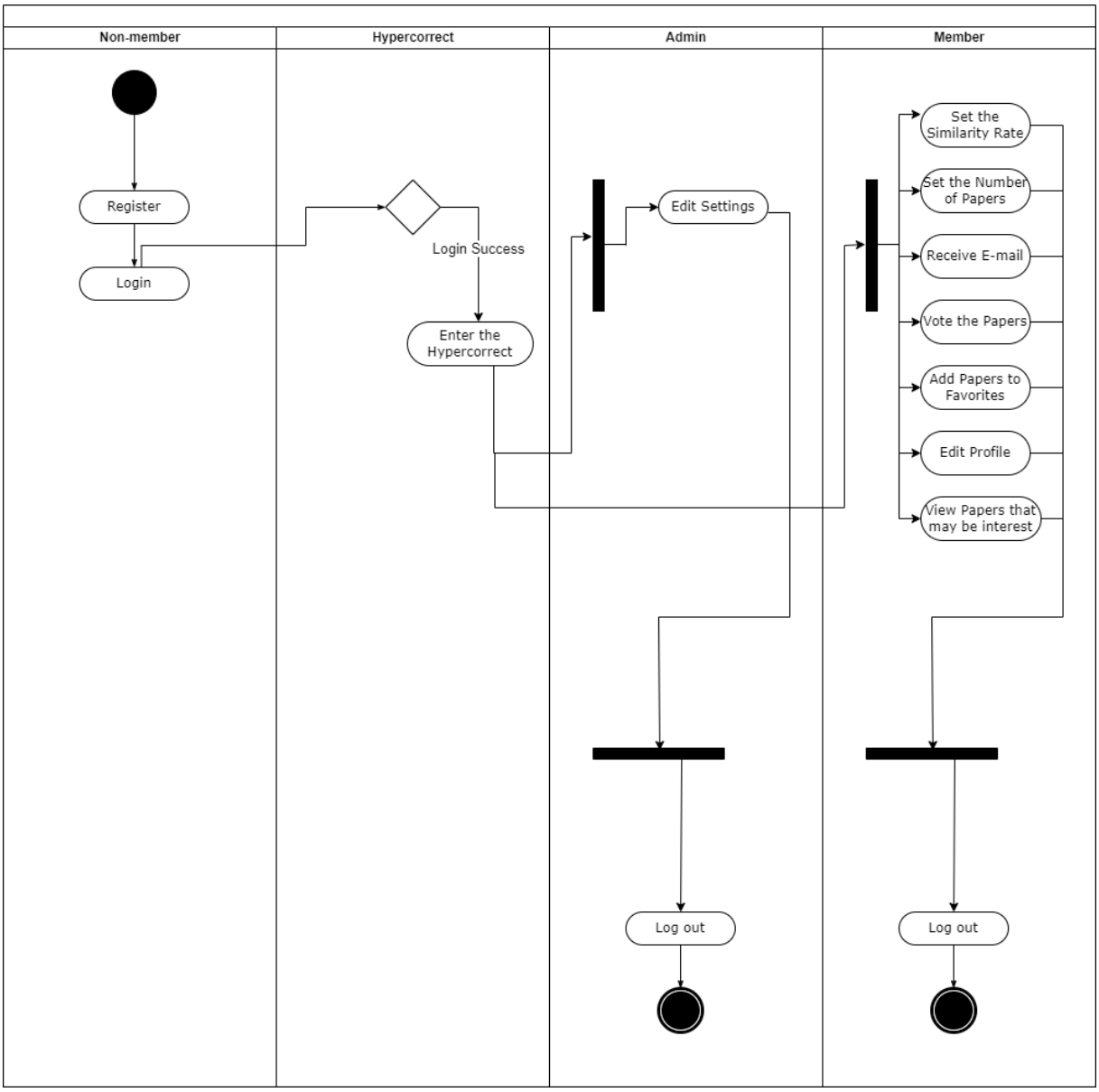

Figure 5: Activity Diagram

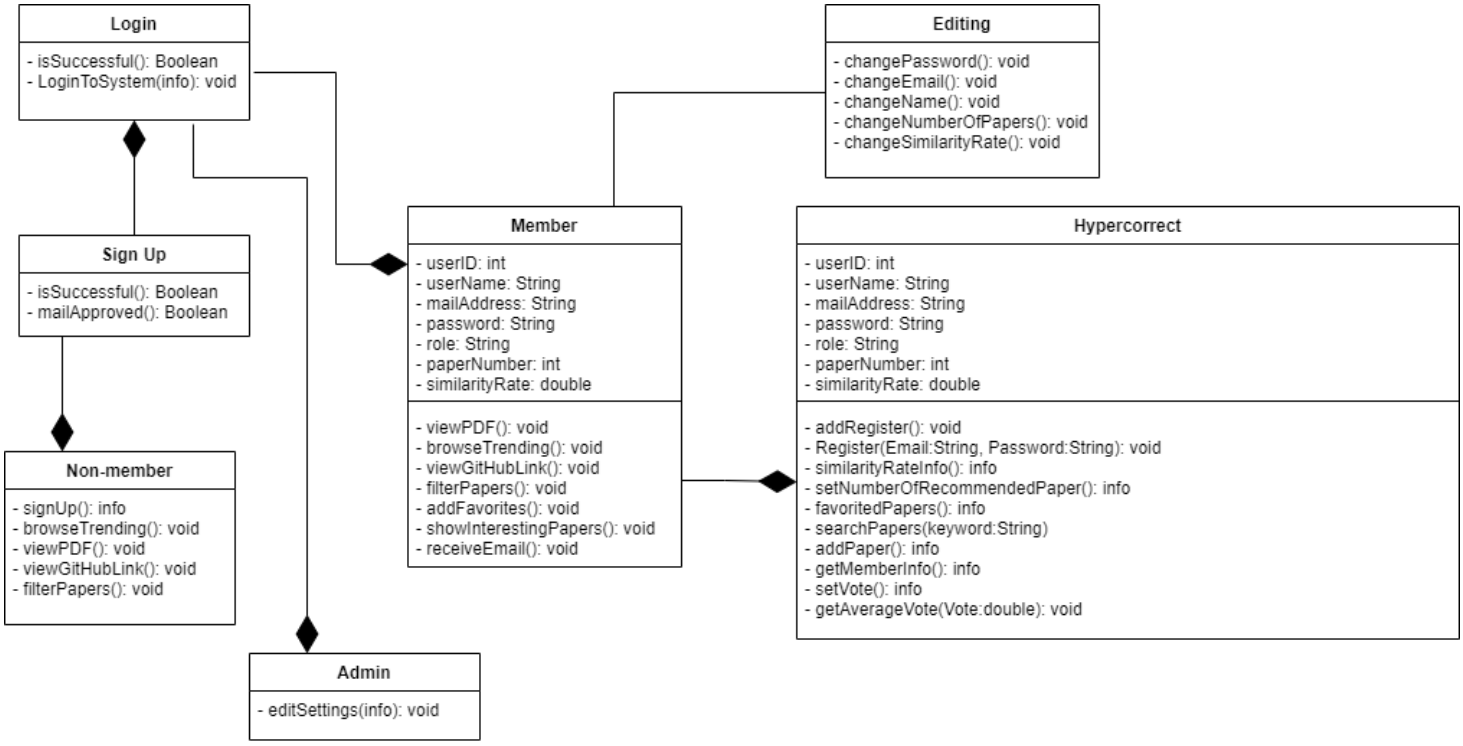

Figure 6: Class Diagram

4.3.2.1 Login Class

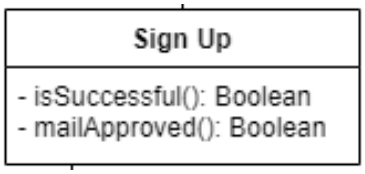

Figure 7: Login Class

*4.*3.2.1.1 Functions

- isSuccessful: This function is called when a user attempts to log in but the system only allows registered users to do so.
- LoginToSystem: This function is called when a user wants to log in. This function calls isSuccesful and mailApproved if all of them is true user can log in to the system.

4.3.2.2 Non-member Class

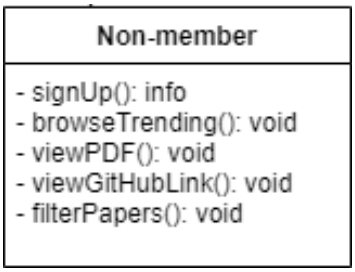

Figure 8: Non-member Class

4.3.2.2.1 Functions

- signUp: This function is called when a non-member hits “Sign Up” button in the “Sign Up” page.
- browseTrending: This function is called when a non-member hits “Home Page” button.
- viewPDF: This function is called when a non-member hits “View PDF” button.
- viewGitHubLinks: This function is called when a non-member hits “GitHub” button.
- filterPapers: This function is called when a non-member searches by applying filter.

4.3.2.3 Member Class

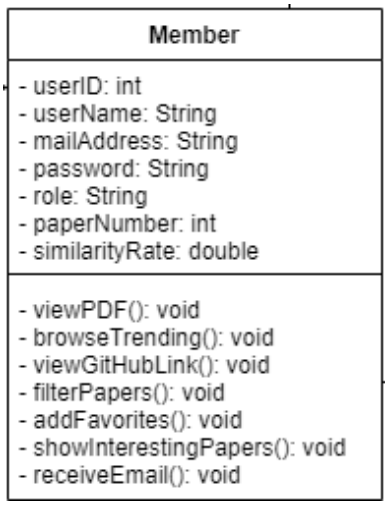

Figure 9: Member Class

*4.*3.2.3.1 Fields

- userID: This is an integer value that is unique for every user. This id is chosen by the system.
- userName: This is the name of the user.
- mailAddress: This is e-mail of the user. Every user must have unique e-mails.
- password: Every user has his/her own password that is selected by him/her.
- role: This is the role of user. (Admin, member)
- paperNumber: This is the filter of user for recommendation. It represents how many papers should be recommended on a page.
- similarityRate: This is the filter of user for recommendation. It represents the similarity ratio between papers and user interests.

*4.*3.2.3.2 Functions

- viewPDF: This function is called when a non-member hits “View PDF” button.
- browseTrending: This function is called when a member hits “Home Page” button.
- viewGitHubLinks: This function is called when a member hits “GitHub” button.
- filterPapers: This function is called when a member searches by applying filter.
- addFavorites: This function is called when a member hits “Add to Favorites” button.
- showInterestingPapers: This function is called when a member hits “Papers You May Like” button.
- receiveEmail: This function is called when a new paper is found that might be of interest to the member.

4.3.2.4 Editing Class

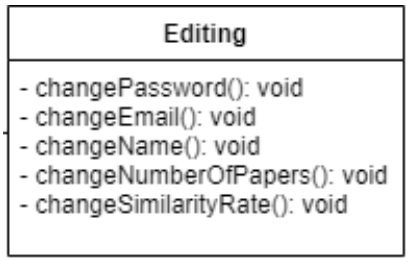

Figure 10: Editing Class

*4.*3.2.4.1 Functions

- changePassword: This function is called when a user hits “Change Password” button.
- changeEmail: This function is called when a user hits “Change E-mail” button.

- changeName: This function is called when a user hits “Change Name” button.
- changeNumberOfPaper: This function is called when a user changes her/his Paper Number filter.
- changeSimilarityRate: This function is called when a user changes her/his Similarity Rate filter.

4.3.2.5 Admin Class

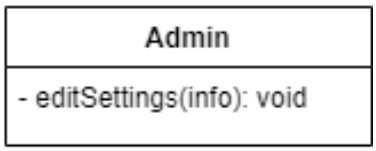

Figure 11: Admin Class

*4.*3.2.5.1 Functions

- editSettings: This function is called when admin changes the site settings.

4.3.2.6 Hypercorrect Class

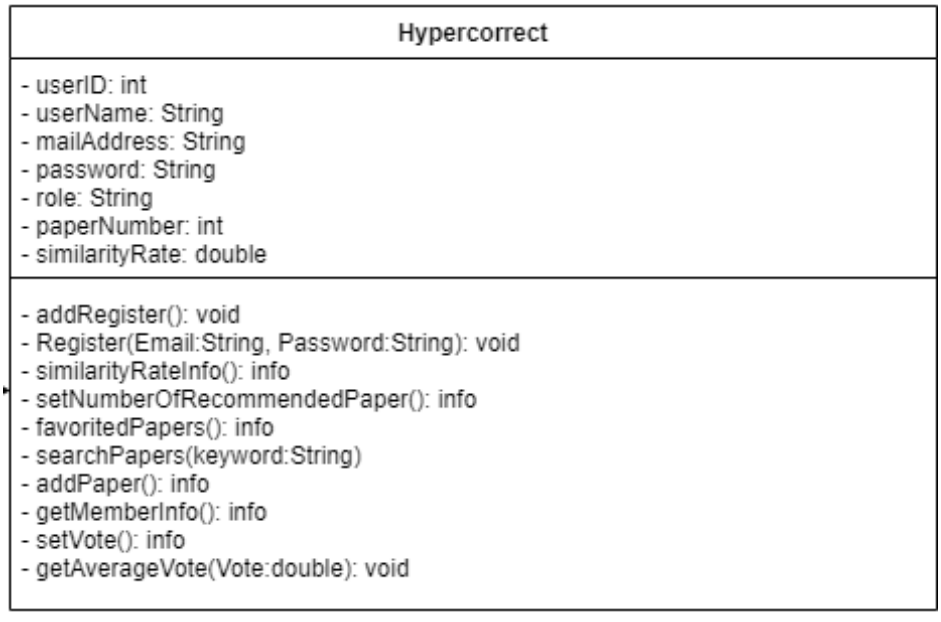

Figure 12: Hypercorrect Class

4.3.2.6.1 Fields

- userID: This is an integer value that is unique for every user. This id is chosen by the system.
- userName: This is the name of the user.
- mailAddress: This is e-mail of the user. Every user must have unique e-mails.
- password: Every user has his/her own password that is selected by him/her.
- role: This is the role of user. (Admin, member)
- paperNumber: This is the filter of user for recommendation. It represents how many papers should be recommended on a page.
- similarityRate: This is the filter of user for recommendation. It represents the similarity ratio between papers and user interests.

4.3.2.6.1 Functions

- addRegister: This function is called when user hits “Sign Up” button in Sign Up Page.
- Register: This function is called when user hits the “Sign Up” button and fill out the form in the Sign-Up Page.
- similarityRateInfo: This function is called when user apply similarity rate filter.
- setNumberOfRecommendedPaper: This function is called when user set number of recommended paper.
- favoritedPapers: This function is called when user add a paper to her/his favorites.
- searchPapers: This function is called when a user hits the “Search” button in the Home Page.
- addPaper: This function is not accessible by system users. This function is used by an auxiliary system that adds papers to be stored in the system.
- getMemberInfo: This function is called when a non-member fills out the form in the “Sign Up” page and become a member.
- setVote: This function is called when a member votes a paper.
- getAverageVote: This function is called in order to get average stars of the paper.

4.4 User Interface Design

4.4.1 Home Page User Interface

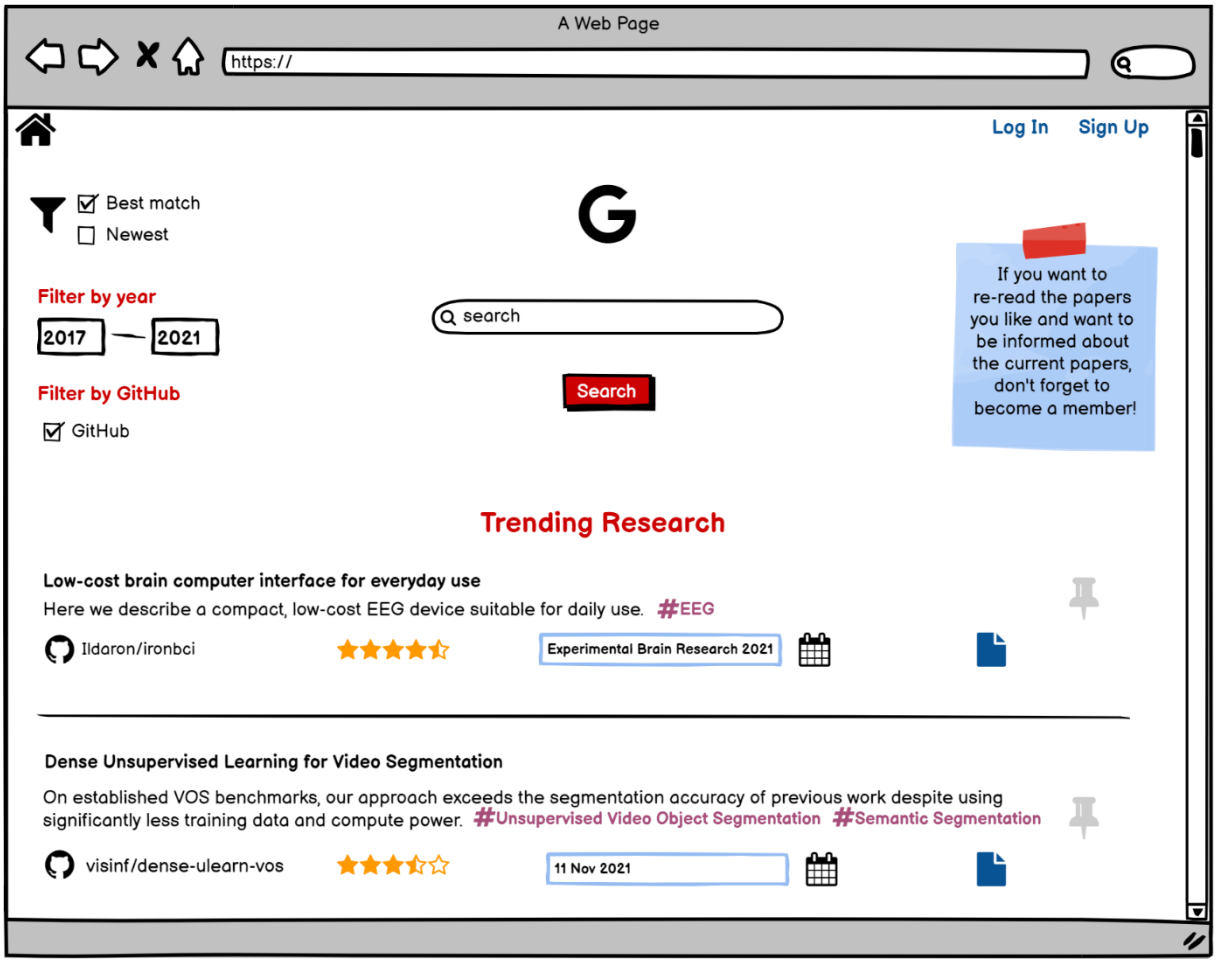

Figure 23: Home Page

4.4.2 Sign Up User Interface

Sign up interface of the system has three attributes in terms of name, e-mail, and password.

Figure 14: Sign Up User Interface

4.4.3 Login User Interface

Login interface of the system has two attributes in terms of e-mail and password.

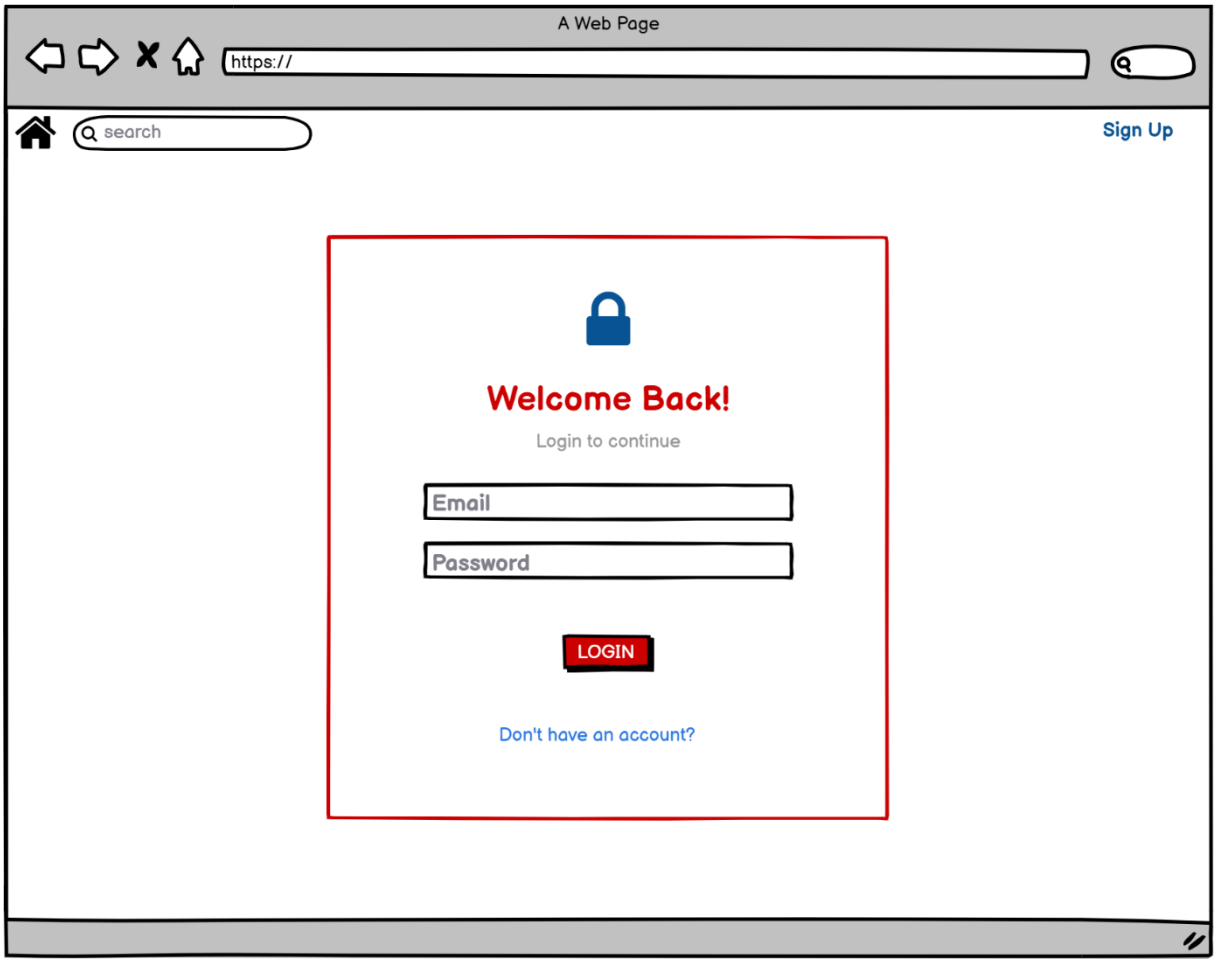

Figure 15: Login User Interface

4.4.4 Profile User Interface

User can change her/his information and filters.

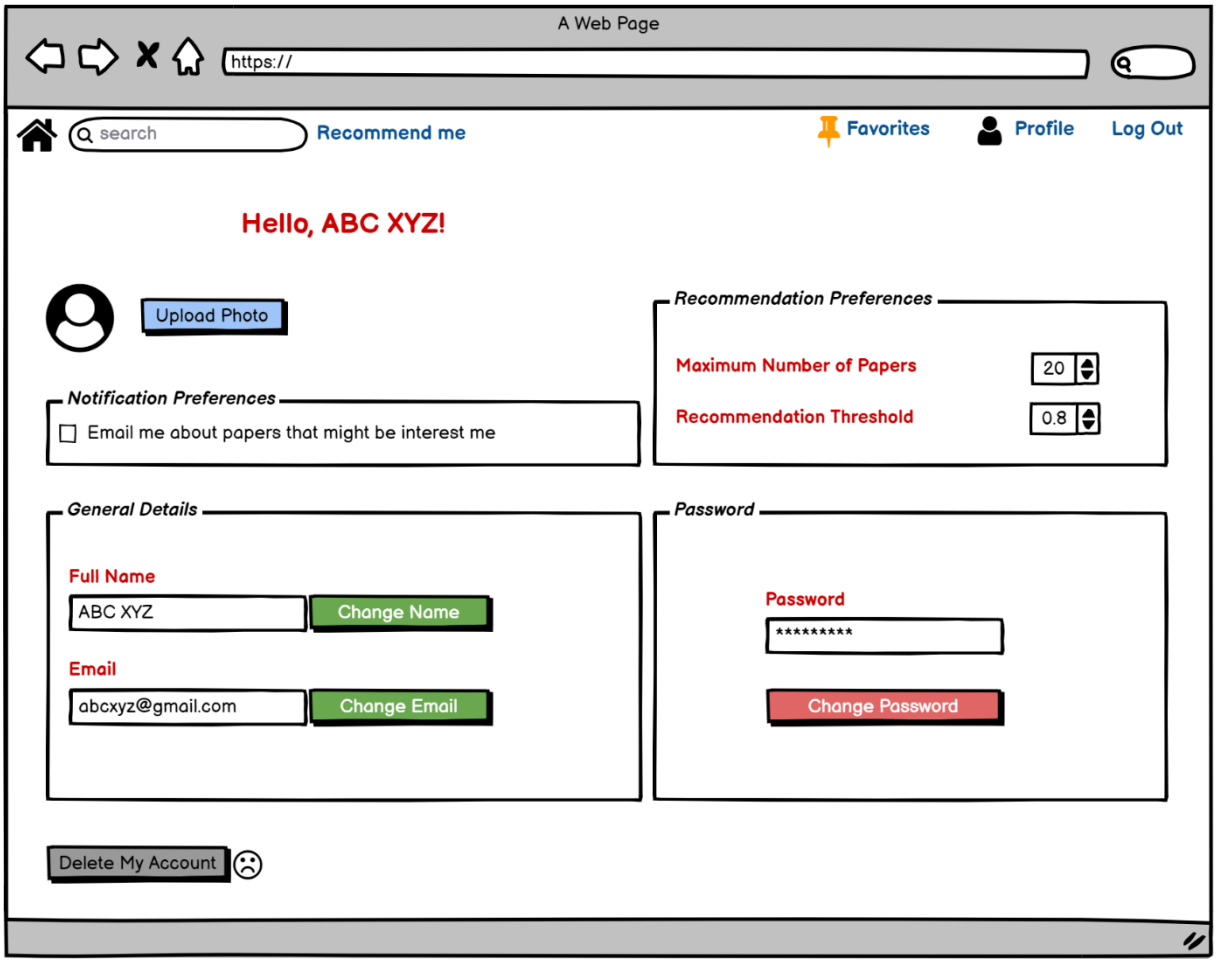

Figure 16: Profile User Interface

4.4.5 Home Page with Filters User Interface

User can apply some filters on the home page.

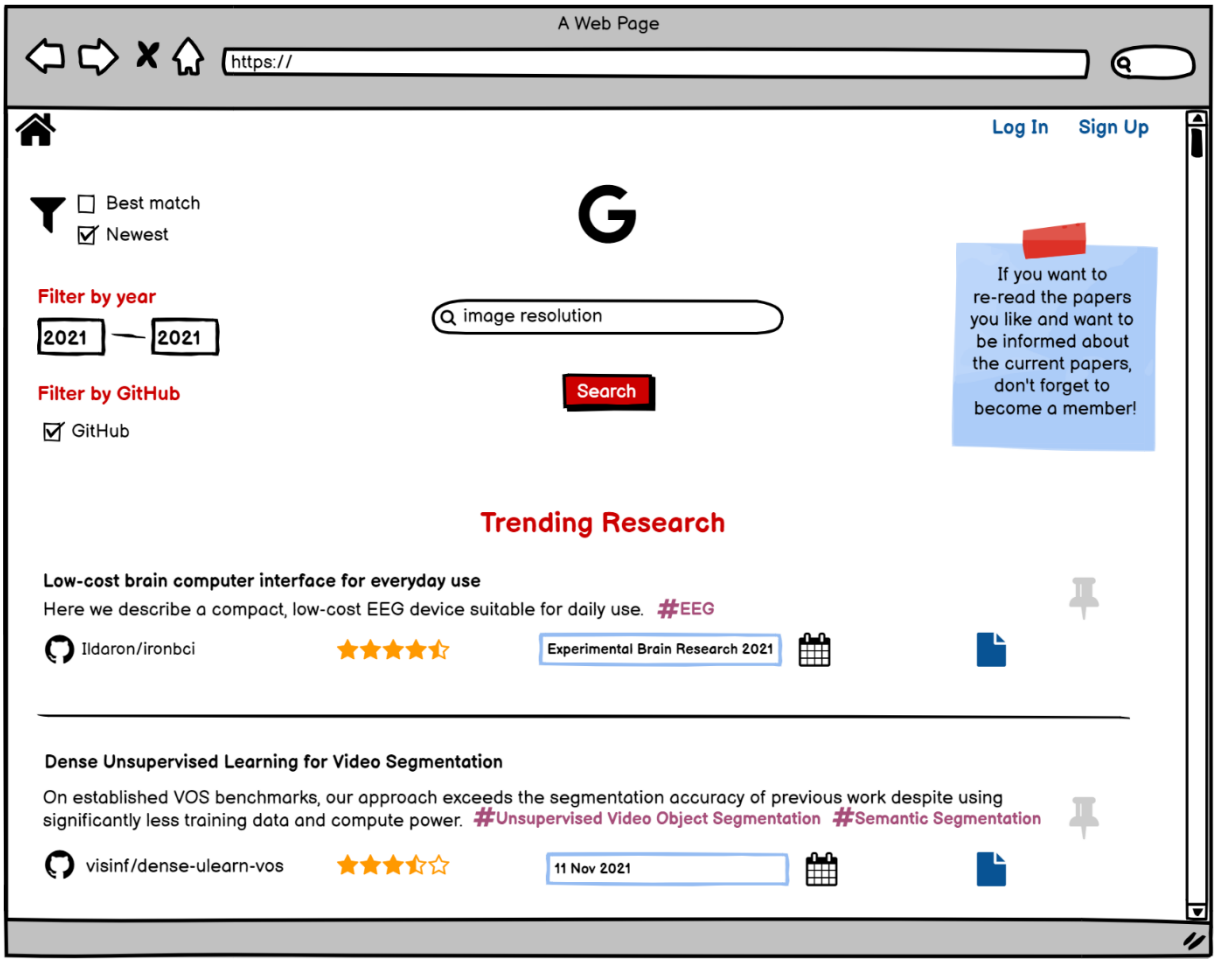

Figure 17: Home Page with Filters User Interface

4.4.6 Searching User Interface

This page shows recommended papers based on the user's filters and searched words on the home page.

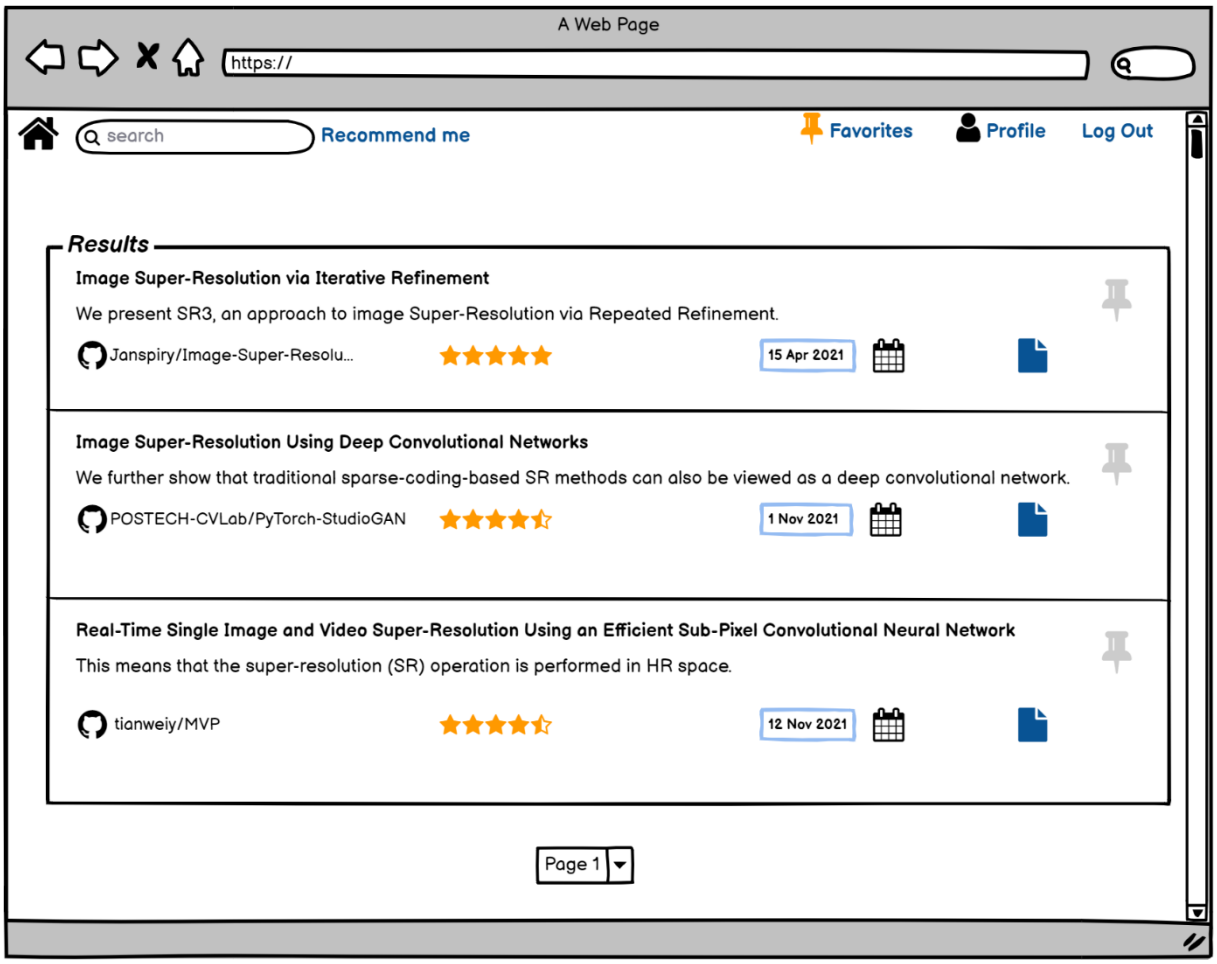

Figure 18: Searching User Interface

4.4.7 Paper Detail User Interface

This page shows detail about the paper that the user clicked.

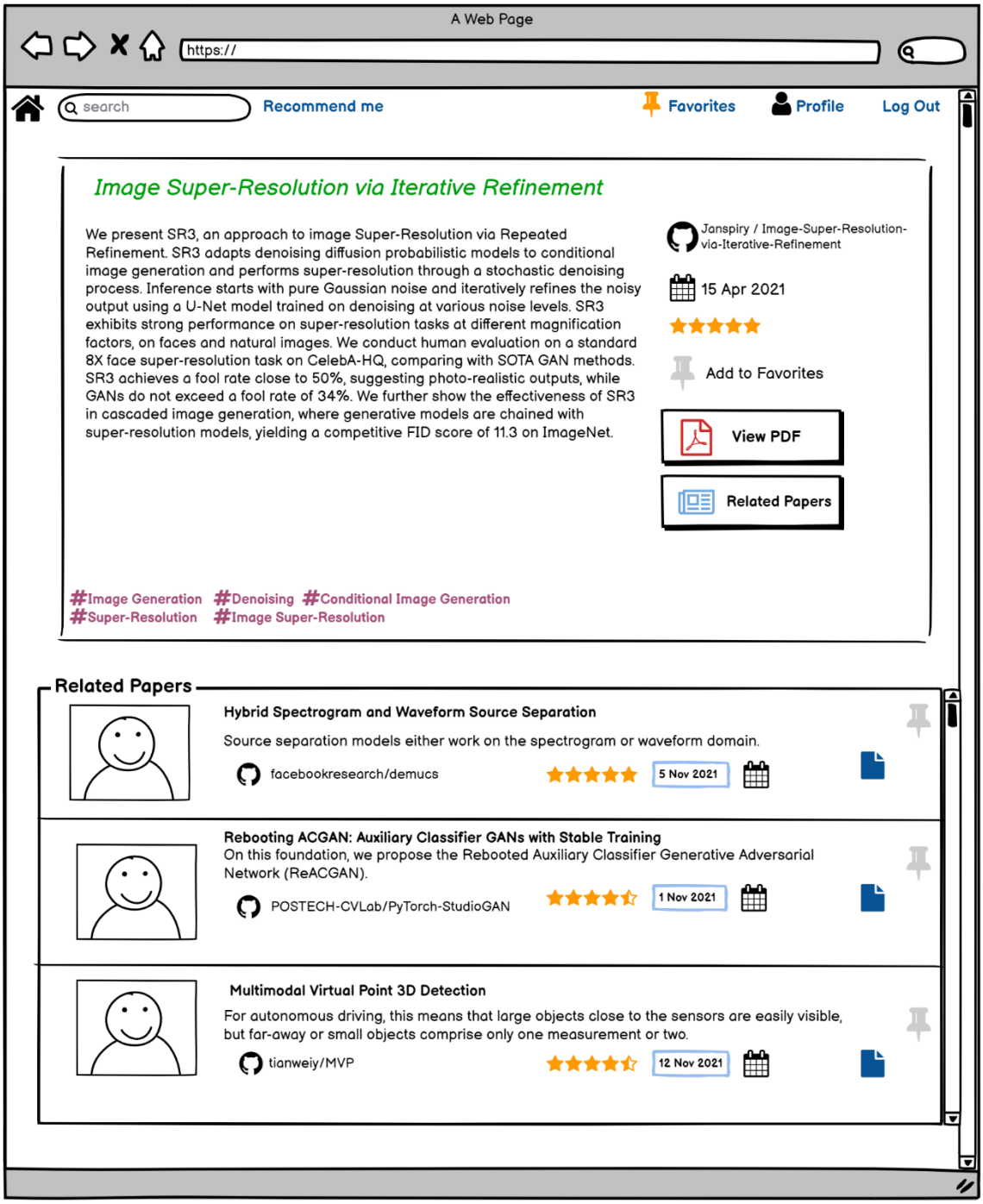

Figure 19: Paper Detail User Interface

4.4.8 My Favorites User Interface

This page shows the papers that the user has favorited.

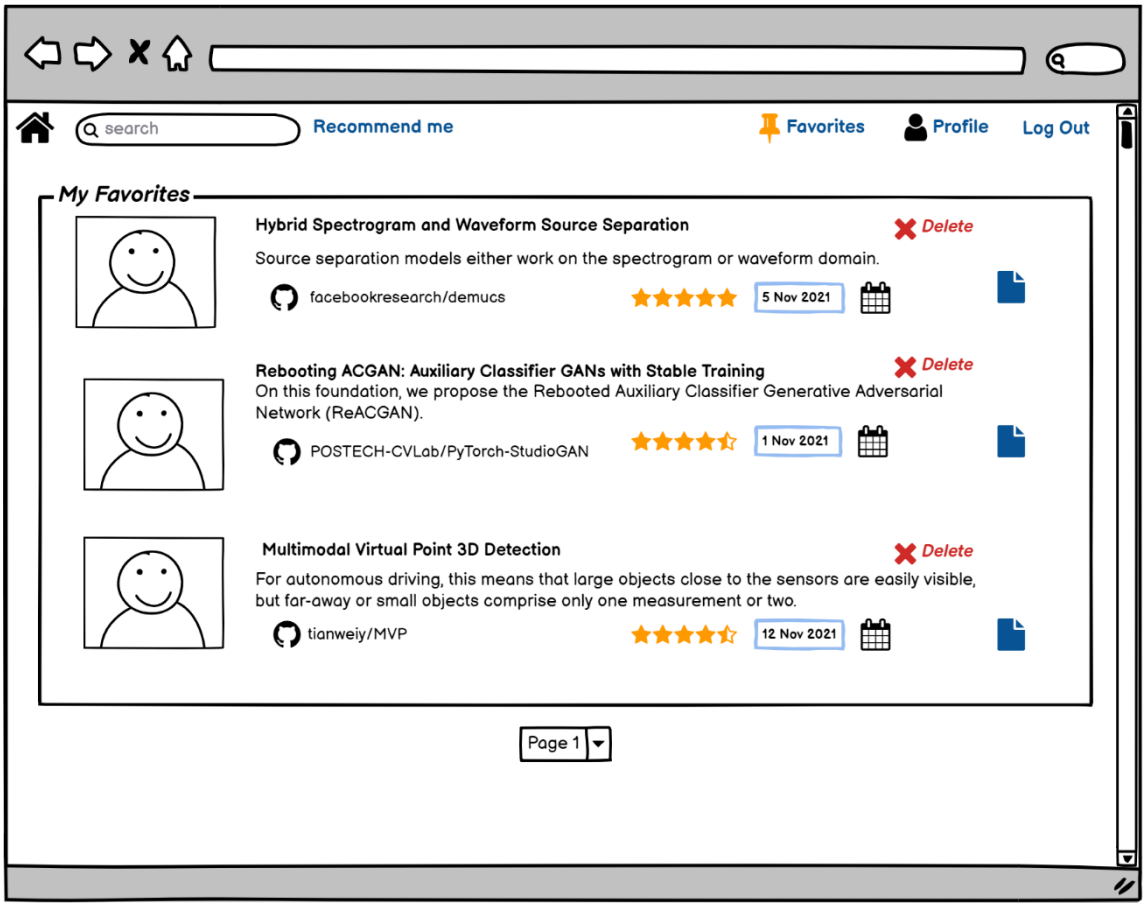

Figure 20: My Favorites User Interface

4.4.9 Papers You May Like User Interface

On this page, the papers that the user may like are displayed according to the filters applied by the user.

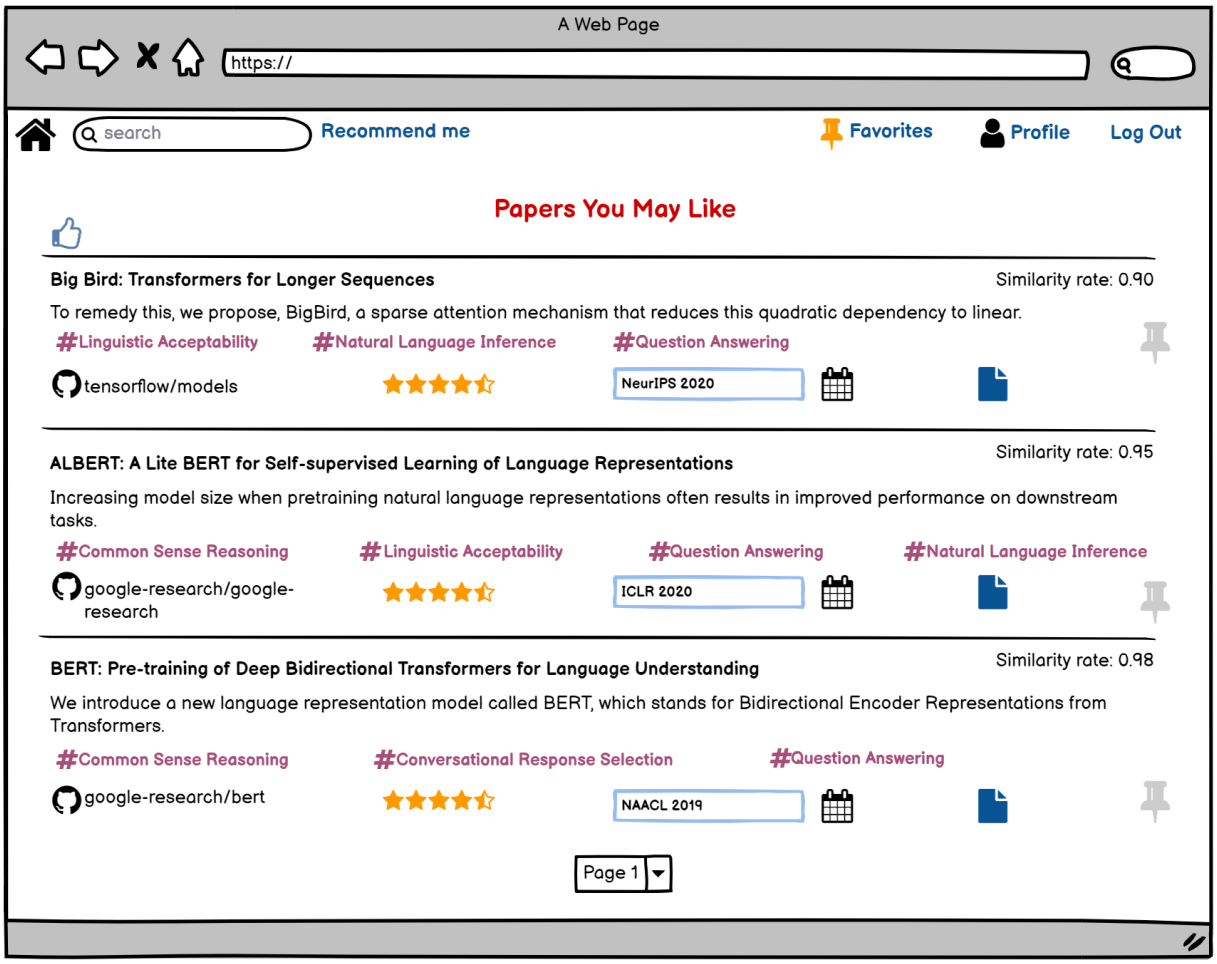

Figure 21: Papers You May Like User Interface

4.5 Requirements Matrix

ID	Ass. ID	Requirements Description	Business Need Justification	Functional Requirement / Use Case	Priority	Test Case
1	1.1	Sign Up Module	Clients require a method to access protected content.	Sign Up to the System	High	UC-001
1	1.2	Login Module	Clients require a method to access protected content.	Login	High	UC-002
1	1.3	Logout Module	We must log out users for security reasons.	Logout	High	UC-012
2	2.1	Notification Module	To inform users, client need a notification system.	Receive E-mail	High	UC-009
3	3.1	Edit Module	Users can edit their profile.	Edit Profile	High	UC-003
4	4.1	Voting Module	Users can vote the papers.	Vote the Papers	Medium	UC-004
5	5.1	Favorite Module	Users can add papers the favorites.	Add to Favorites	Medium	UC-005
6	6.1	Filter Operation	Users can filter the papers.	Filter the Papers	High	UC-006
6	6.2	Filter Operation	Users can set the similarity rate for recommendation.	Set the Similarity Rate	High	UC-010
6	6.3	Filter Operation	Users can set the number of papers for recommendation.	Set the Number of Papers	High	UC-011
7	7.1	Showing Operation	Users can see the code that related with paper.	Show the Code	High	UC-007
7	7.2	Showing Operation	Users can see the PDF of paper.	View PDF	High	UC-008

Table 19: Requirements Matrix

5. Conclusion & Discussion

In our CENG 407 project, we researched and explained academic research recommendation system. This document includes information about how our project Hypercorrect will turn out and list benefits to its users in a detailed manner. Aim of our site is to provide users appropriate recommendations of academic papers through their interest and meanwhile designing this project we really found this kind of services very useful. As a result of our research, choosing Python in the data collection and preprocess phase; preferring HTML, CSS, JavaScript, and React for front-end; Java for backend and we realized that choosing Spring Boot Framework and choosing Postgres for data storage will help us achieve more accurate results. While the texts will be expressed as vectors using a pre-trained BERT model, the cosine similarity measure will be utilized to compare them.

Project Work Plan

TASKS	START DATE	END DATE	STATUS	OCTOBER				NOVEMBER				DECEMBER				JANUARY			
				1.WEEK	2.WEEK	3.WEEK	4.WEEK	5.WEEK	6.WEEK	7.WEEK	8.WEEK	9.WEEK	10.WEEK	11.WEEK	12.WEEK	13.WEEK	14.WEEK	15.WEEK	16.WEEK
Project Proposal Form	18 OCT 2021	26 OCT 2021	Completed																
Team Setup	19 OCT 2021	26 OCT 2021	Completed																
Project Selection Form	20 OCT 2021	26 OCT 2021	Completed																
GitHub Repository	31 OCT 2021	05 NOV 2021	Completed																
Project Work Plan	10 NOV 2021	12 NOV 2021	Completed																
Literature Review	20 OCT 2021	12 NOV 2021	Completed																
Software Requirements Specification	12 NOV 2021	10 DEC 2021	Completed																
Project Webpage	11 DEC 2021	17 DEC 2021	Completed																
Software Design Description	18 DEC 2021	31 DEC 2021	Completed																
Project Report / Project Tracking Form	01 JAN 2022	07 JAN 2022	Completed																
Presentation	17 JAN 2022	25 JAN 2022	Not Started																

REFERENCES

[1] “Software Requirements Specification Prepared by BlueQuotersfor the project QuoteShot”, [Online]. Available: https://senior.ceng.metu.edu.tr/2016/bluequoters/docs/BlueQuoters-SRS.pdf

[2] “Use Case Diagrams (Register & Login)” [Online]. Available: https://sites.google.com/site/louismccallapcp/home/development/use-cases/use-case%20diagram

[3] “Data flow diagram of a system” [Online]. Available: https://www.researchgate.net/figure/Data-flow-diagram-of-a-system-login-and-registration-feature_fig17_262014927

[4] “Unified Modeling Language (UML) | Activity Diagrams”, [Online]. Available: https://www.geeksforgeeks.org/unified-modeling-language-uml-activity-diagrams/#:~:text=An%20activity%20diagram%20is%20a,the%20activity%20is%20being%20executed.

[5] “UML Class Diagram Tutorial”, [Online]. Available: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/

[6] “IEEE Standard for Information Technology-Systems Design-Software Design Descriptions”, [Online]. Available: http://cengproject.cankaya.edu.tr/wp-content/uploads/sites/10/2017/12/SDD-ieee-1016-2009.pdf

[7] “Diagram Draw Tool”, [Online]. Available: https://app.diagrams.net/

[8] “Mendeley Research Manager: Why Mendeley?”, [Online]. Available: https://guides.lib.uci.edu/Mendeley

[9] “Mendeley”, [Online]. Available: https://tr.wikipedia.org/wiki/Mendeley