



**Bilkent University**  
**Department of Computer Engineering**

**Senior Design Project**  
*T2310*  
*ReLink*

**Final Report**

*Ayberk Yaşı, 21801847, ayberk.yasa@ug.bilkent.edu.tr*  
*Cemhan Kaan Özaltan, 21902695, kaan.ozaltan@ug.bilkent.edu.tr*  
*Çağatay Şafak, 21902730, cagatay.safak@ug.bilkent.edu.tr*  
*Fatih Kaplama, 21802755, fatih.kaplama@ug.bilkent.edu.tr*  
*Görkem Ayten, 21802399, gorkem.ayten@ug.bilkent.edu.tr*

*Eray Tüzün*

*Erhan Dolak, Tağmaç Topal*

**18th of May 2023**

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

<b>1. Introduction</b>	<b>3</b>
<b>2. Requirements Details</b>	<b>3</b>
2.1. Functional Requirements	3
2.2. Nonfunctional Requirements	4
2.2.1. Usability	4
2.2.2. Reliability	5
2.2.3. Performance	5
2.2.4. Supportability	5
2.2.5. Scalability	5
<b>3. Final Architecture and Design Details</b>	<b>5</b>
3.1. Overview	5
3.2. Subsystem Decomposition	6
3.3. Persistent Data Management	7
3.4. Access Control and Security	7
3.5. Boundary Conditions	8
3.5.1. Initialization	8
3.5.2. Termination	8
3.5.3. Failure	8
<b>4. Development/Implementation Details</b>	<b>9</b>
4.1. Frontend	9
4.2. REST API	10
4.3. Data Retrieval	10
4.4. Link Recovery	11
4.4.1. Word Embeddings	11
4.4.2. Heuristic Checks	12
4.5. Graph Builder	13
<b>5. Test Cases and Results</b>	<b>13</b>
5.1. Functional Test Cases	13
5.2. Non-Functional Test Cases	27
5.3. Mean Reciprocal Rank Test Cases	31
<b>6. Maintenance Plan and Details</b>	<b>34</b>
<b>7. Other Project Elements</b>	<b>36</b>
7.1. Consideration of Various Factors in Engineering Design	36
7.2. Ethics and Professional Responsibilities	36
7.3. Teamwork Details	37
7.3.1. Contributing and functioning effectively on the team	37
7.3.2. Helping creating a collaborative and inclusive environment	37
7.3.3. Taking lead role and sharing leadership on the team	37
7.3.4. Meeting objectives	37
7.3.4.1. Requirements Report	37
7.3.4.2. UI Implementation	38
7.3.4.3. Backend Implementation	38
7.3.4.4. Design Report	38

7.3.4.5. ML Implementation	38
7.3.4.6. Testing	39
7.3.4.7. Case Study with Arçelik	39
7.3.4.8. Final Report	39
7.4. New Knowledge Acquired and Applied	39
7.4.1. Researching ML-Based Applications	40
7.4.2. Exploring ML-Based Tools and Algorithms	40
7.4.4. Dockerization of the Web Application	40
7.4.5. Integration of Acquired Knowledge	40
<b>8. Conclusion and Future Work</b>	<b>41</b>
<b>9. Glossary</b>	<b>41</b>
<b>10. References</b>	<b>42</b>

# 1. Introduction

In our age of technology, bugs, improvements, and features of a software project are stored and tracked in project management tools that provide development teams with an issue tracker. The rise of agile results in large amounts of data about issues in these project management tools which require much more effort to inspect manually [1]. Moreover, pull requests are fundamental artifacts used in the software development lifecycle (SDLC) [1], [2]. Whereas issues are stored in bug tracking and project management tools such as Jira, Bugzilla, and GitHub Issues, pull requests and commits are stored in servers utilizing a version control system (VCS) such as GitHub, GitLab, and Bitbucket. Such contemporary VCS options offer built-in features for manually linking issues with commits and pull requests through a predefined commit message syntax where the issue ID is explicitly given, allowing for a more structured and traceable project. However, despite the importance of traceability in software development, most pull requests are not explicitly linked to issues by developers due to the lack of the fixed rules [3]. The information contained in the missing links is therefore lost, reducing efficiency in processes such as bug localization, bug prediction, and feature improvements [3]. Since it is highly possible for developers to forget establishing such links or fail tagging a commit due to a typo and the manual cost of recovering these links is very high, we propose an automation tool for tackling this issue.

ReLink is a web application which serves as a tool for automating the issue-pull request linking process during software development. To be specific, ReLink detects missing links retrospectively and creates a link between issue-pull request pairs. This is done through certain machine learning (ML) algorithms. ReLink also provides the visualization of the analysis of missed links among pull requests and issues by certain filters such as by developer, by time, and by reason. Finally, ReLink offers a graphical interface for visualizing data on the historical evolution and progress of a specific project's issues. To be specific, there is a historical graph visualization for links among commits, pull requests, and issues.

The final structure of ReLink will be described in detail in the following sections.

## 2. Requirements Details

### 2.1. Functional Requirements

- Admin should be able to assign a project manager for each project.
- Admin should be able to manage(create, read, update, delete) users in the system.
- Admin should be able to manage(create, read, update, delete) projects of the users in the system.

- Project Managers should be able to log in, and log out the system.
- Developers should be able to sign up, log in, and log out the system.
- Developers should be able to link all past PRs and issues manually through possible match suggestions.
  - Using title similarity
  - Using heuristic rules
- Developers should be able to see the visualization of the analysis of missed links between PRs and issues.
  - By developer
  - By time
  - By reason, e.g., a typo, missing, or technical things
- Developers should be able to see a historical graph visualization for links among commits, pull requests, and issues, where the links are constructed by an ML algorithm.
- The Project Manager should be able to create and delete a project.
- The Project Manager should be able to give a developer access to the project.
- The Project Manager should be able to deprive a developer of access to the project.
- The system should be able to support GitHub and AzureDevOps as a version control system and Jira as a bug tracking system.

## **2.2. Nonfunctional Requirements**

### **2.2.1. Usability**

- There should exist a navigational sidebar visible from all pages for allowing the user to easily navigate the app.
- The titles on the navigation bar, the titles on each page, and the labels on all buttons should be meaningful and self-explanatory so that users who do not read the user manual are able to understand how the website is used from the titles on the navigation bar, the titles on the screen and the labels on the buttons.
- All screens including pop-ups should be reached by being clicked at most twice.
- In order to be a user-friendly website, it is imperative to be consistent in the website layout and design by following a design pattern in which all screens accessed from the sidebar use the same main template.
- Except for the pop-ups, none of the screens on the website must be connected to each other, so users don't have to go backward on the website.
- Web Content Accessibility Guidelines [9] must be followed, which makes content in the website more accessible to users with disabilities.

### **2.2.2. Reliability**

- Users must be able to access a historical graph visualization for links among commits, pull requests, and issues of their projects 90% of the time without failure.
- The data fetched from repositories must be used without changing.
- Users who close the browser without properly logging out of their account will be automatically logged out.

### **2.2.3. Performance**

- The load time of each page must be less than 2 seconds.
- The average response time of each button must be about 2 seconds.
- The website must keep the above-mentioned times the same, up to 100 simultaneous users.

### **2.2.4. Supportability**

- User feedback will be evaluated continuously and if there is any bug reported by users, it will be assigned to a developer within 24 hours.

### **2.2.5. Scalability**

- When the daily traffic of this website, which has 1000 visitors per day, exceeds this number, the max bandwidth limit of this website that is allocated to the hosting plan should not be exceeded.

## **3. Final Architecture and Design Details**

### **3.1. Overview**

The architecture of ReLink consists of 3 layers in terms of high level architecture: presentation, business, and persistence. The presentation layer is responsible for displaying the interface content of ReLink to the user through dynamic and interactive components implemented with React.js. Moreover, it is responsible for handling user requests and generating the appropriate responses.

The business layer provides the application's core logic and functionality, including business rules, data validation, and processing of user inputs. Here, the user data is processed and stored using models, using the Django library. The presentation layer depends on the endpoints provided by the REST API module. The link recovery module is responsible for determining possible issue-PR pairs of a given software projects using word embedding, metadata classification, and heuristic rules. This package is implemented with libraries such as Scikit-learn and Gensim. The data retrieval module is responsible for fetching pull request, commits and issues data of

user's repositories from Jira REST API and Azure DevOps REST API. These APIs are external systems and therefore do not reside in the ReLink system. This data, after being saved in the persistence layer, is used by the link recovery module, the graph builder module, and also the REST API module to provide it to the client.

The persistence layer is responsible for storing user and application data for persistence through different sessions. Here, the Django models are turned into database records and are stored.

## 3.2. Subsystem Decomposition

ReLink consists of 5 main subsystems, which are shown with the following diagram and are explained in detail in Section 4.

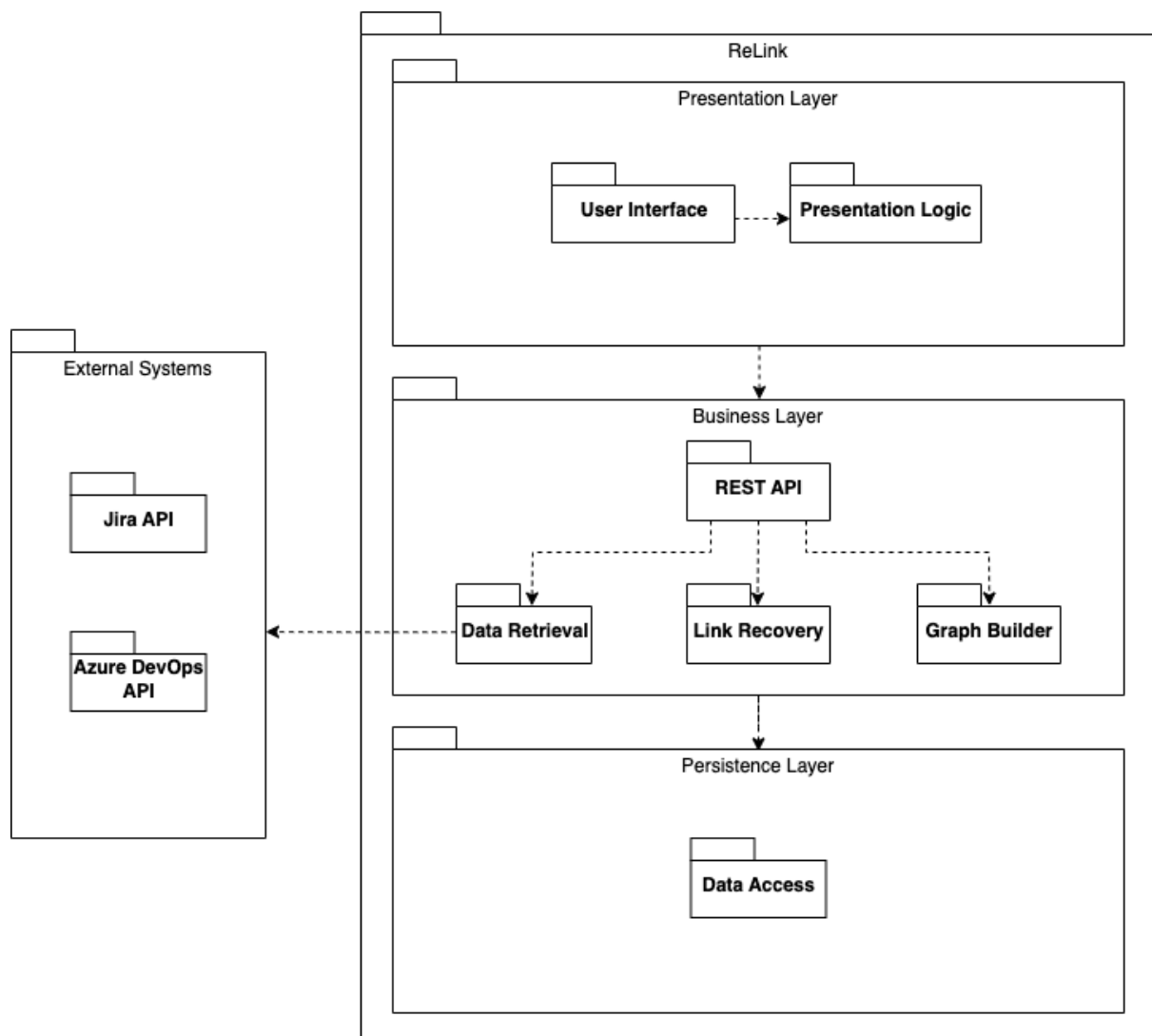


Fig. 1. Package diagram.

### 3.3. Persistent Data Management

In this project, a specific database is needed since clients send a request to the server to access the specific information about pull request, issue, repository. These data should be accessed quickly. In this project, PostgreSQL will be used. The reason PostgreSQL is chosen, it is an object relational database that is convenient to apply OOP principles. Moreover, PostgreSQL has various functionalities such as storing array and json as a column. Thanks to PostgreSQL, the project's database will be more functional, speed and safe. ReLink's main objects such as User, Developer, Project Manager, Project, Repository, Pull Request, Issue, Commit, and Links will be stored in this database that is deployed in our production server by using AWS [9]. Clients can make some operations such as sending GET, PUT, POST and DELETE requests to the server. It is important to provide such operations to client.

### 3.4. Access Control and Security

Our web application has access control and it provides security. It is provided access control by giving permission to the logged user with the matching used type. In this way, a user cannot use the properties of a user type that they do not belong to. Also, to improve the security of the web application, there will be a good password policy. Users must create a password of at least 8 characters including lowercase, uppercase, special characters and numbers. These improve the security and the access control of the application.

#### Matrix for Access Control

Actors \ Objects	Project	Repository	PR	Issue	Commit	Link	Analysis
Developer	R	R	R	R	R	R	R
Project Manager	RW	RW	R	R	R	RWX	RX

**Note:** In the context of our application, X (Execute) means that an actor should be able to start an operation orchestrated by the system on an object. In other words, the actor orders the system to complete the operation instead of directly performing the operation.



## 3.5. Boundary Conditions

### 3.5.1. Initialization

The initialization section of the boundary conditions specifies the conditions that must be met for the system to start operating correctly.

**Input validation:** The system should validate all input data to ensure it is within expected ranges and conforms to the required data types.

**Resource allocation:** The system should allocate all necessary resources, such as memory or network connections, prior to starting execution.

**Configuration loading:** The system should load any required configuration files, such as database credentials or API keys.

**Data loading:** The system should load data from the database, such as project, user, repository mining data.

**Data validation:** The system should check if access tokens of projects registered by users are expired. The system also should check if URLs of these projects are broken.

### 3.5.2. Termination

The termination section of the boundary conditions specifies the conditions that must be met for the system to shut down correctly.

**Resource deallocation:** The system should release all resources, such as memory or network connections, that were allocated during initialization.

**Data persistence:** The system should save any necessary data to a durable store, such as a database or file system, before shutting down.

**Cleanup:** The system should perform any necessary cleanup activities, such as cleaning cache, interrupting ongoing processes or terminating network connections.

### 3.5.3. Failure

The failure section of the boundary conditions specifies the conditions that must be met in the event of a failure or error.

**Error handling:** The system should gracefully handle any errors or exceptions that occur during execution, such as logging errors and providing appropriate error messages to the user.

**Rollback:** The system should revert to a known good state in the event of a failure, such as rolling back a transaction or restoring data from a backup.

**Notification:** The system should alert appropriate parties, such as system administrators or support personnel, in the event of a failure that requires manual intervention.

## 4. Development/Implementation Details

### 4.1. Frontend

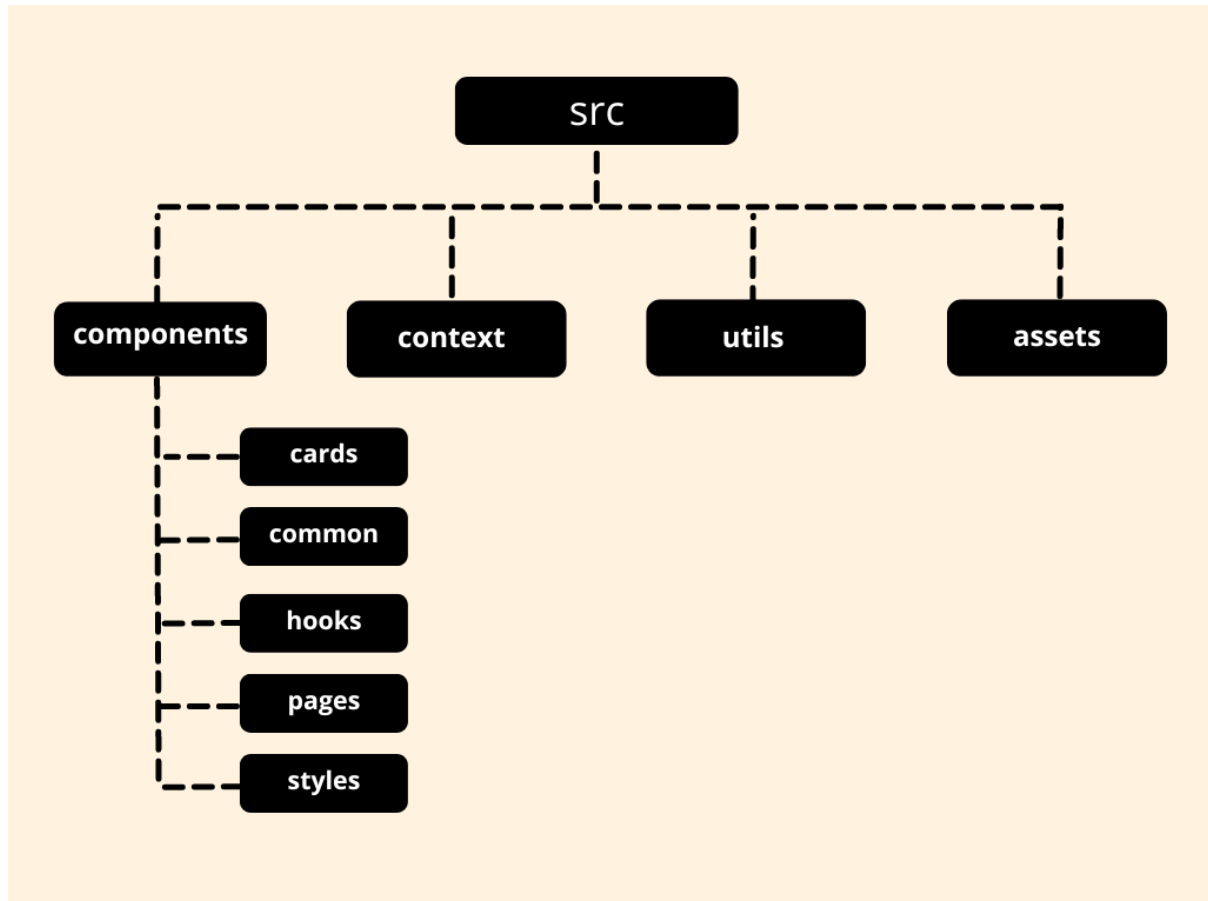


Fig. 2. High-level frontend structure.

In frontend architecture, **src** folder consists of 4 different folders. In the **context** folder there is a file called **Auth** and the task of this file is to store the authentication information in the context API. In the **utils** folder, there is a file named **service**. This file is responsible for communication with the backend. All Axios requests are written in this file and the request is sent to the backend thanks to the methods written in this file. In the **assets** folder there are logo files of the project. The **components** folder consists of 5 different folders. These are **cards**, **common**, **hooks**, **pages**, and **styles**. In the **cards** folder, there are files that are responsible for the cards in the project such as analysis card, github card, missing links card. In the **common** folder, there are components used commonly in the project such as navigation bar and side bar. In the **hooks** folder there are hooks components used for authentication and local storage. In the **pages** folder. There are page components of the project such as New Project, Historical Graph, Login. In the **styles** folder there are css and scss files that are responsible for the design of the project.

## 4.2. REST API

The ReLink API module is a powerful web service that enables communication between the presentation layer and the backend of your application. This advanced module is built using Django and Django REST Framework to provide a scalable, modifiable, and easy-to-use interface.

With ReLink API, resources such as pull requests, issues, users, and commits are exposed through a uniform interface using URIs (Uniform Resource Identifiers) and HTTP methods (GET, POST, PUT, DELETE) to perform actions on them. The API responds with representations of these resources in JSON (JavaScript Object Notation) format.

A key feature of ReLink API is its stateless architecture, which means that each request contains all the information necessary for the server to perform the request. This design makes the API highly scalable and easy to cache, leading to improved performance.

Authentication and authorization are essential aspects of ReLink API. Authentication is used to verify the identity of users or applications, while authorization is employed to determine whether a user or application has permission to perform a specific action on a resource. Okta authentication is used for authentication within the ReLink API, while different types of users, such as developers and project managers, have access to different authorization levels.

## 4.3. Data Retrieval

This module is responsible for fetching pull request, commit, and issue data from repositories of a project that is created by the user in the application. It interacts with two external systems which are Jira and Azure DevOps REST API to fetch these data. In this module, there are a total of 2 classes, each responsible for retrieving data from a different source. Facade design pattern is used so that the REST API subsystem can easily use the data retrieving action from this subsystem. In this way, the REST API subsystem can initiate the whole process with a single method call without needing to know the details of the data retrieving process. When the data retrieval process is completed, the Data Retrieval module interacts with the persistence layer and saves the data to the database. In this way, REST API, Link Recovery, and Graph Builder subsystems can utilize these data.

## 4.4. Link Recovery

### 4.4.1. Word Embeddings

Word embedding is a natural language processing (NLP) technique, where a vector representation is learned for each word. The vector representations of similar words are closer to each other in terms of geometrical distance (either cosine or Euclidian distance), making word embeddings a useful technique in many NLP domains such as sentiment analysis, text generation, and text similarity [5], [6]. The most common method for learning word embeddings is the Word2Vec algorithm, which utilizes a shallow neural network for training embedding models and can be accessed through the Python Gensim library. There exist two techniques for implementing a Word2Vec model. The first one is continuous bag of words (CBOW), where a target word is predicted according to the input context words, allowing for missing word prediction. On the other hand, the skip-gram model predicts the context words of the input word, in a similar fashion to an inverted CBOW model [7].

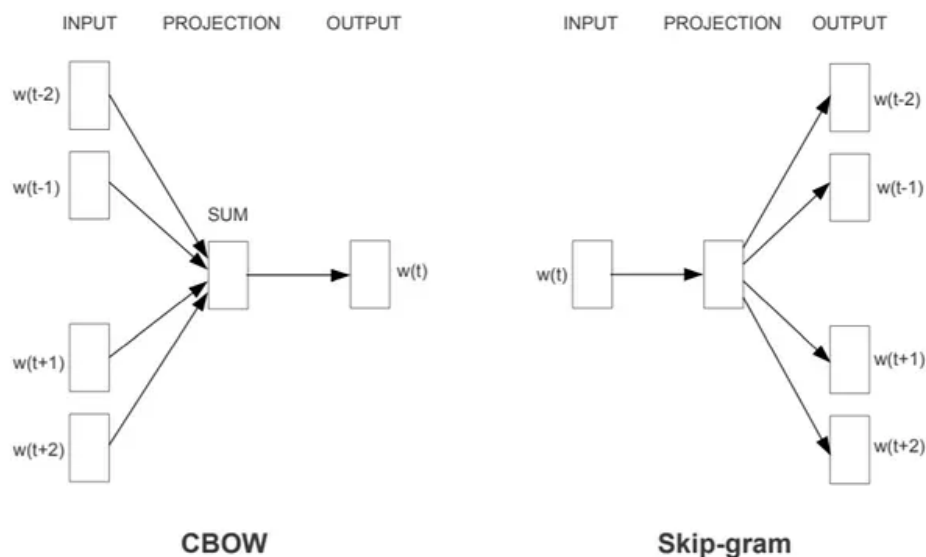


Fig. 3. CBOW vs. skip-gram [6].

Both model types can be trained with the Gensim library, without having to develop a neural network architecture from the ground up. Moreover, existing pre-trained models, such as Google's "GoogleNews-vectors-negative300" model trained on approximately 100 billion words using both the CBOW and skip-gram techniques, can also be used [8]. We used this model since we do not currently have sufficient data for model training and even if we did, it would be difficult to outperform a model trained on such a large dataset. Once the words are embedded into vectors, we must use these vectors for calculating text similarity. For this, we first pre-process our sentence corpus by removing stopwords (common words) and punctuation, changing non-English characters to their English counterparts, turning the entire

corpus to lowercase, and lemmatizing the corpus (removing suffixes that add no contextual information). A preprocessing example can be shown as follows:

```
pre_process(["Missing links page should provide more information"]) ->
["missing link page provide information"]
```

Since we have a model that is already trained for embedding (vectorizing) words, we use this model to find the weighted sum of the word vectors in each sentence. This weighting is done with the term frequency-inverse document frequency (TF-IDF) of each word being multiplied by them while taking the sentence sum, which increases the weights of words with a higher frequency in the corpus for maintaining contextual information. Once all the sentence embeddings are calculated, the cosine similarities of the issue summary and PR title vector sets are calculated by taking the dot products of their embedding vector matrices, as shown below (theta is the angle between vectors):

$$\text{cosine similarity} = S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

Fig. 4. Cosine similarity [6].

The cosine similarity is higher for vectors with a smaller angle in-between, meaning that their corresponding sentences are semantically and contextually similar, and the respective issue-PR pair is likely to be linked. This module of the link recovery subsystem is, therefore, responsible for detecting text similarity between issue summaries and PR titles.

A translation script is used to translate from Turkish to English, specifically for the case study with Arcelik.

#### 4.4.2. Heuristic Checks

Finally, we perform heuristic checks on the predicted links to determine whether they can really be linked according to a pre-determined set of rules. These rules are mainly based on mathematical calculations using date information of issues and PRs. Such rules strengthen the prediction accuracy of our models according to logical rules that must almost certainly hold in all cases. As stated, this is not an ML technique and is based on mathematical calculations such as jaccard index (intersection over union) [9].

## 4.5. Graph Builder

This module utilizes the pull request, issue, and commit data fetched by the Data Retrieval module and saved in the database. The purpose of this module is to create a graph data structure where each pull request is associated with commits and issues. For this, it detects all relationships for these three models in the database and groups them by pull requests. In order to create a visual graph data structure on the frontend, it organizes the data to be sent to the frontend and puts it in the appropriate form.

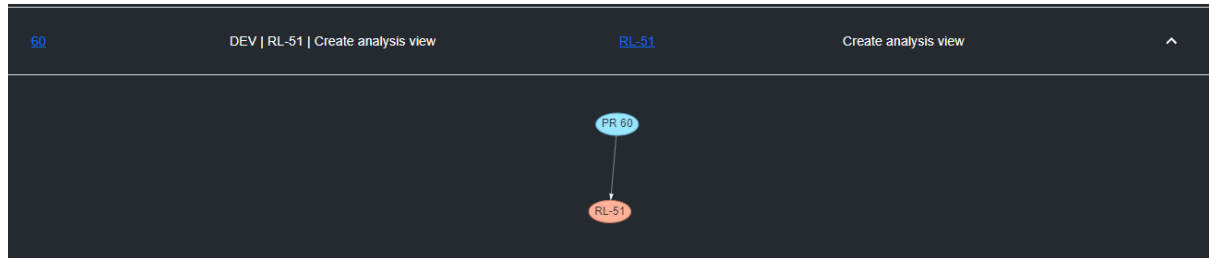


Fig. 6. Example network graph rendered from the data sent by Graph Builder.

## 5. Test Cases and Results

ReLink's main priority is to achieve a high accuracy when connecting issues and pull requests. For that reason, our ML model is the most important functionality of our system. In this section, both the tests done by humans and the automated script tests will be explained and evaluated. There are 38 test cases in total, including 24 functional test cases, 9 non-functional test cases, and 5 MMR test cases.

### 5.1. Functional Test Cases

<b>Test Case ID</b>	#F-01
<b>Test Type/Category</b>	Functional
<b>Summary/Title/Objective</b>	Test Case for the Login Feature with Valid Credentials.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"><li>• Launch the website and navigate to the login page.</li><li>• Enter the company email and password.</li><li>• Click on the "Login" button.</li><li>• Verify that the user is redirected to the overview page.</li><li>• Verify that the user's name is displayed in the top right corner of the dashboard.</li></ul>

<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>• The user should be able to log in with valid credentials.</li> <li>• The user should be redirected to the dashboard page after successful login.</li> <li>• The user's name should be displayed in the top right corner of the dashboard.</li> </ul>
<b>Priority/Severity</b>	Critical
<b>Date Tested</b>	March 13, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#F-02
<b>Test Type/Category</b>	Functional
<b>Summary/Title/Objective</b>	Test Case for the Login Feature with Invalid Credentials.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>• Launch the website and navigate to the login page.</li> <li>• Enter an invalid company email and password.</li> <li>• Click on the "Login" button.</li> <li>• Verify that an error message is displayed.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>• The user should not be able to log in with invalid credentials.</li> <li>• An error message should be displayed indicating the login was unsuccessful.</li> </ul>
<b>Priority/Severity</b>	Major
<b>Date Tested</b>	March 13, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#F-03
<b>Test Type/Category</b>	Functional
<b>Summary/Title/Objective</b>	Test Case for the registration feature with valid data.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>• Launch the website and navigate to the registration page.</li> <li>• Enter valid data in all required fields.</li> </ul>

	<ul style="list-style-type: none"> <li>Click on the "Register" button.</li> <li>Verify that the user is redirected to the login page.</li> <li>Verify that a success message is displayed.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>The user should be able to log in with valid credentials.</li> <li>The user should be redirected to the dashboard page after successful login.</li> <li>The user's name should be displayed in the top right corner of the dashboard.</li> </ul>
<b>Priority/Severity</b>	Critical
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#F-04
<b>Test Type/Category</b>	Functional
<b>Summary/Title/Objective</b>	Test Case for the Logout Feature.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>Launch the website and log in with valid credentials.</li> <li>Click on the "Logout" button.</li> <li>Verify that the user is redirected to the login page.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>The user should be able to log out successfully.</li> <li>The user should be redirected to the login page after successful logout.</li> </ul>
<b>Priority/Severity</b>	Critical
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#F-05
<b>Test Type/Category</b>	Functional
<b>Summary/Title/Objective</b>	Test Case for the Repository Filtering Feature.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>Launch the website and log in with valid credentials.</li> </ul>



	<ul style="list-style-type: none"> <li>• Navigate to the Project List screen.</li> <li>• Press the Open Project button.</li> <li>• Navigate to the Possible Links screen.</li> <li>• Select a repository from the list.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>• The user should be able to see only the possible links for the PR's of the selected repository.</li> </ul>
<b>Priority/Severity</b>	Minor
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#F-06
<b>Test Type/Category</b>	Functional
<b>Summary/Title/Objective</b>	Test Case for the "Open Project" Button
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>• Launch the website and log in with valid credentials.</li> <li>• Navigate to the Project List screen.</li> <li>• Click on the "Open Project" button of any of the listed projects and verify that the corresponding page is displayed.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>• The user should be able to navigate to the corresponding page of the project.</li> </ul>
<b>Priority/Severity</b>	Major
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#F-07
<b>Test Type/Category</b>	Functional
<b>Summary/Title/Objective</b>	Test Case for the "Manage Project Access" button.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>• Launch the website and log in with valid credentials.</li> <li>• Navigate to the Project List screen.</li> </ul>

	<ul style="list-style-type: none"> <li>Click on the “Manage Project Access” button of any of the listed projects and verify that the corresponding pop-up is displayed.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>The user should be able to see a corresponding pop-up.</li> </ul>
<b>Priority/Severity</b>	Major
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#F-08
<b>Test Type/Category</b>	Functional
<b>Summary/Title/Objective</b>	Test Case for the left frame for general view.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>Launch the website and log in with valid credentials.</li> <li>Navigate to any possible screen.</li> <li>Click on any of the 4 possible buttons at the left frame: “Project List”, “New Project”, “Help &amp; Support”, or “Logout”.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>The user should be able to navigate to the desired screens through the left frame.</li> </ul>
<b>Priority/Severity</b>	Minor
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#F-09
<b>Test Type/Category</b>	Functional
<b>Summary/Title/Objective</b>	Test Case for the left frame for project-specified view.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>Launch the website and log in with valid credentials.</li> <li>Navigate to any possible screen.</li> </ul>

	<ul style="list-style-type: none"> <li>Click on any of the 4 possible buttons at the left frame: "Overview", "Analysis", "Possible Links", or "Historical Links".</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>The user should be able to navigate to the desired screens through the left frame.</li> </ul>
<b>Priority/Severity</b>	Minor
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#F-10
<b>Test Type/Category</b>	Functional
<b>Summary/Title/Objective</b>	Test Case for the "Back to Project List" button.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>Launch the website and log in with valid credentials.</li> <li>Navigate to the Project List screen.</li> <li>Click on the "Back to Project List" button on the left sidebar and verify that it navigates to the project list.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>The user should be able to navigate to the project list.</li> </ul>
<b>Priority/Severity</b>	Major
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#F-11
<b>Test Type/Category</b>	Functional
<b>Summary/Title/Objective</b>	Test Case for Selecting the Number of Rows in the Possible Links Pagination.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>Launch the website and log in with valid credentials.</li> <li>Navigate to the Project List screen.</li> </ul>

	<ul style="list-style-type: none"> <li>• Navigate to the Project Overview screen through any of the possible projects.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>• The user should be able to view a list of numbers, which are 10, 25, and 50. The user should be able to set the number of rows in the paginated table of possible links.</li> </ul>
<b>Priority/Severity</b>	Major
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#F-12
<b>Test Type/Category</b>	Functional
<b>Summary/Title/Objective</b>	Test Case for the Graph Builder.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>• Launch the website and log in with valid credentials.</li> <li>• Navigate to the Project List screen.</li> <li>• Navigate to the Project Overview screen through any of the possible projects.</li> <li>• Navigate to the Project Analysis screen.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>• The user should be able to view the graphs for the already executed analysis.</li> </ul>
<b>Priority/Severity</b>	Major
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#F-13
<b>Test Type/Category</b>	Functional
<b>Summary/Title/Objective</b>	Test Case for Listing the Missing Links in a Project.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>• Launch the website and log in with valid credentials.</li> <li>• Navigate to the Project List screen.</li> </ul>

	<ul style="list-style-type: none"> <li>• Navigate to the Project Overview screen through any of the possible projects.</li> <li>• Navigate to the Project Missing Links screen.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>• The user should be able to see the missing links in the project, which has at least one proposed solution by the ML model.</li> </ul>
<b>Priority/Severity</b>	Major
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#F-14
<b>Test Type/Category</b>	Functional
<b>Summary/Title/Objective</b>	Test Case for Listing the Potential Solutions for Missing Links in a Project.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>• Launch the website and log in with valid credentials.</li> <li>• Navigate to the Project List screen.</li> <li>• Navigate to the Project Overview screen through any of the possible projects.</li> <li>• Navigate to the Project Missing Links screen.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>• The user should be able to see the potential relinks provided by the ML model, for each of the missing links.</li> <li>• The user should be able to see the details of a potential relink, such as the issue ID, confidence (accuracy) rate, name of the related repository, and the issue title.</li> </ul>
<b>Priority/Severity</b>	Major
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#F-15
<b>Test Type/Category</b>	Functional

<b>Summary/Title/Objective</b>	Test Case for Checking the Check-box(es) for the Potential Relinks.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>• Launch the website and log in with valid credentials.</li> <li>• Navigate to the Project List screen.</li> <li>• Navigate to the Project Overview screen through any of the possible projects.</li> <li>• Navigate to the Project Missing Links screen.</li> <li>• At the listed potential issues for any of the listed pull requests, choose the issue(s) which you think are relevant to the pull request.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>• The user should be able to choose the issues they like to relink with the pull request of interest.</li> <li>• The user should be able to see a warning message about selecting more than one issue is allowed, but not recommended.</li> </ul>
<b>Priority/Severity</b>	Major
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#F-16
<b>Test Type/Category</b>	Functional
<b>Summary/Title/Objective</b>	Test Case for the Relink Functionality.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>• Launch the website and log in with valid credentials.</li> <li>• Navigate to the Project List screen.</li> <li>• Navigate to the Project Overview screen through any of the possible projects.</li> <li>• Navigate to the Project Missing Links screen.</li> <li>• At the listed potential issues for any of the listed pull requests, choose the issue(s) which you think are relevant to the pull request.</li> <li>• Click on the “RELINK” button at the bottom of the dialog.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>• The user should be able to choose the issues they like to relink with the pull request of interest.</li> </ul>

	<ul style="list-style-type: none"> <li>• The user should be able to receive a status message about whether the relink operation is finished with success.</li> <li>• If the relinking operation is successful, the pull request should be removed from the screen.</li> </ul>
<b>Priority/Severity</b>	Critical
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#F-17
<b>Test Type/Category</b>	Functional
<b>Summary/Title/Objective</b>	Test Case for Ignoring a Possible Relink.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>• Launch the website and log in with valid credentials.</li> <li>• Navigate to the Project List screen.</li> <li>• Navigate to the Project Overview screen through any of the possible projects.</li> <li>• Navigate to the Project Missing Links screen.</li> <li>• At the listed potential issues for any of the listed pull requests, choose the issue(s) which you think are relevant to the pull request.</li> <li>• Click on the "IGNORE" button at the bottom of the dialog.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>• The user should be able to choose the issues they like to relink with the pull request of interest.</li> <li>• The user should be able to receive a status message about their choice.</li> <li>• The pull request should be removed from the screen.</li> </ul>
<b>Priority/Severity</b>	Major
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#F-18
---------------------	-------

<b>Test Type/Category</b>	Functional
<b>Summary/Title/Objective</b>	Test Case for Add Another Issue Functionality to a Possible Relink.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>• Launch the website and log in with valid credentials.</li> <li>• Navigate to the Project List screen.</li> <li>• Navigate to the Project Overview screen through any of the possible projects.</li> <li>• Navigate to the Project Missing Links screen.</li> <li>• At the listed potential issues for any of the listed pull requests, choose the issue(s) which you think are relevant to the pull request.</li> <li>• Click on the “ADD ANOTHER ISSUE” button at the bottom of the dialog.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>• The user should be able to see a dialog where they can add another issue by typing the issue key (like RL-1234, XYZ-123, etc.).</li> </ul>
<b>Priority/Severity</b>	Major
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#F-19
<b>Test Type/Category</b>	Functional
<b>Summary/Title/Objective</b>	Test Case for Handling Bad Requests in the Add Another Issue Functionality to a Possible Relink.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>• Launch the website and log in with valid credentials.</li> <li>• Navigate to the Project List screen.</li> <li>• Navigate to the Project Overview screen through any of the possible projects.</li> <li>• Navigate to the Project Missing Links screen.</li> <li>• At the listed potential issues for any of the listed pull requests, choose the issue(s) which you think are relevant to the pull request.</li> <li>• Click on the “ADD ANOTHER ISSUE” button at the bottom of the dialog.</li> </ul>



	<ul style="list-style-type: none"> <li>Write an issue key which is not available in Jira.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>The user should be able to see a dialog where they can add another issue by typing the issue key (like RL-1234, XYZ-123, etc.).</li> <li>When the issue key is not valid, the user should not be able to use the add another issue functionality.</li> </ul>
<b>Priority/Severity</b>	Major
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#F-20
<b>Test Type/Category</b>	Functional
<b>Summary/Title/Objective</b>	Test Case for the “Manage Project Access” button, for users who are not the PM (Project Manager).
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>Launch the website and log in with valid credentials.</li> <li>Navigate to the Project List screen.</li> <li>Verify that the corresponding “Manage Project Access” button is not clickable.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>If the user is not PM, then the user should be able to see that the corresponding button is disabled, i.e., not clickable.</li> </ul>
<b>Priority/Severity</b>	Major
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#F-21
<b>Test Type/Category</b>	Functional
<b>Summary/Title/Objective</b>	Test Case for Filtering Possible Links by PR ID.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>Launch the website and log in with valid credentials.</li> </ul>

	<ul style="list-style-type: none"> <li>• Navigate to the Project List screen.</li> <li>• Navigate to the Project Overview page by clicking to the Open Project button for one of the projects listed.</li> <li>• Navigate to the Possible Links page by using the sidebar.</li> <li>• In the search bar above, enter a search query on the "Search by PR ID" field and click to the search button.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>• The user should be able to see the PR with the desired ID. If there is no such a PR, then a blank table should be displayed.</li> </ul>
<b>Priority/Severity</b>	Major
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Failed

<b>Test Case ID</b>	#F-22
<b>Test Type/Category</b>	Functional
<b>Summary/Title/Objective</b>	Test Case for Filtering Possible Links by PR Author.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>• Launch the website and log in with valid credentials.</li> <li>• Navigate to the Project List screen.</li> <li>• Navigate to the Project Overview page by clicking to the Open Project button for one of the projects listed.</li> <li>• Navigate to the Possible Links page by using the sidebar.</li> <li>• In the search bar above, enter a search query on the "Search by PR Author" field and click to the search button.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>• The user should be able to see the PR with the desired author. If there is no such a PR, then a blank table should be displayed.</li> </ul>
<b>Priority/Severity</b>	Major
<b>Date Tested</b>	May 17, 2023

<b>Test Result</b>	Failed
--------------------	--------

<b>Test Case ID</b>	#F-23
<b>Test Type/Category</b>	Functional
<b>Summary/Title/Objective</b>	Test Case for Filtering Possible Links by PR Summary.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>• Launch the website and log in with valid credentials.</li> <li>• Navigate to the Project List screen.</li> <li>• Navigate to the Project Overview page by clicking to the Open Project button for one of the projects listed.</li> <li>• Navigate to the Possible Links page by using the sidebar.</li> <li>• In the search bar above, enter a search query on the “Search by PR Summary” field and click to the search button.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>• The user should be able to see the PR with the desired summary. If there is no such a PR, then a blank table should be displayed.</li> </ul>
<b>Priority/Severity</b>	Major
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Failed

<b>Test Case ID</b>	#F-24
<b>Test Type/Category</b>	Functional
<b>Summary/Title/Objective</b>	Test Case for Filtering Possible Links by PR Status.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>• Launch the website and log in with valid credentials.</li> <li>• Navigate to the Project List screen.</li> <li>• Navigate to the Project Overview page by clicking to the Open Project button for one of the projects listed.</li> <li>• Navigate to the Possible Links page by using the sidebar.</li> </ul>

	<ul style="list-style-type: none"> <li>In the search bar above, select either “Open” or “Closed” on the “Search by PR Status” field and click to the search button.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>The user should be able to see the PR with the desired status. If there is no such a PR, then a blank table should be displayed.</li> </ul>
<b>Priority/Severity</b>	Major
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Failed

## 5.2. Non-Functional Test Cases

<b>Test Case ID</b>	#NF-01
<b>Test Type/Category</b>	Performance
<b>Summary/Title/Objective</b>	Test Case for Load Time of Homepage.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>Open the website homepage in a web browser.</li> <li>Start the timer as soon as the website starts loading.</li> <li>Stop the timer as soon as the website is fully loaded.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>The website homepage should load within 2 seconds.</li> </ul>
<b>Priority/Severity</b>	Major
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#NF-02
<b>Test Type/Category</b>	Performance
<b>Summary/Title/Objective</b>	Test Case for Load Time of Project Page.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>Open the Project Overview screen in a web browser.</li> <li>Start the timer as soon as the page starts loading.</li> </ul>

	<ul style="list-style-type: none"> <li>Stop the timer as soon as the page is fully loaded.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>The product page should load within 2 seconds.</li> </ul>
<b>Priority/Severity</b>	Minor
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#NF-03
<b>Test Type/Category</b>	Security
<b>Summary/Title/Objective</b>	Test Case for Using Functionalities without Logging In.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>Open the website homepage in a web browser.</li> <li>Try to navigate to one of the screens other than /login and /signup without logging in.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>The user should not be able to see screens.</li> </ul>
<b>Priority/Severity</b>	Major
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#NF-04
<b>Test Type/Category</b>	Usability
<b>Summary/Title/Objective</b>	Test Case for Navigation.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>Attempt to navigate to a specific page on the website using the navigation menu (left frame).</li> <li>Verify that the navigation is intuitive and easy to use.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>The navigation should be easy to use and should take no more than 2 clicks to reach any page on the site.</li> </ul>

<b>Priority/Severity</b>	Minor
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#NF-05
<b>Test Type/Category</b>	Compatibility
<b>Summary/Title/Objective</b>	Test Case for Browser Compatibility.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>• Open the website in different browsers (e.g. Chrome, Firefox, Safari, Edge).</li> <li>• Verify that the website displays correctly and functions properly in each browser.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>• The website should be compatible with all major browsers and should display correctly and function properly in each one.</li> </ul>
<b>Priority/Severity</b>	Minor
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Passed (tested in Chrome, Firefox, and Edge)

<b>Test Case ID</b>	#NF-06
<b>Test Type/Category</b>	Security
<b>Summary/Title/Objective</b>	Test Case for Login Authentication.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>• Attempt to log in using incorrect login credentials.</li> <li>• Verify that the system does not grant access.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>• The system should not allow access to the account if the login credentials are incorrect.</li> </ul>
<b>Priority/Severity</b>	Critical
<b>Date Tested</b>	May 17, 2023

<b>Test Result</b>	Passed
--------------------	--------

<b>Test Case ID</b>	#NF-07
<b>Test Type/Category</b>	Usability
<b>Summary/Title/Objective</b>	Test Case for Mobile Responsiveness.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>• Open the website on a mobile device and navigate to different pages.</li> <li>• Verify that the website is mobile responsive and easy to use on a smaller screen.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>• The website should be mobile responsive and easy to use on a smaller screen.</li> </ul>
<b>Priority/Severity</b>	Minor
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Partially passed (Passed for a majority of screens. Moreover, some components did not look well in landscape mode.)

<b>Test Case ID</b>	#NF-08
<b>Test Type/Category</b>	Performance
<b>Summary/Title/Objective</b>	Test Case for Concurrent User Load.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>• Simulate multiple concurrent users accessing the website at the same time.</li> <li>• Verify that the website can handle the load without performance issues or errors.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>• The website should be able to handle multiple concurrent users without performance issues or errors.</li> </ul>
<b>Priority/Severity</b>	Major
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#NF-09
<b>Test Type/Category</b>	Usability
<b>Summary/Title/Objective</b>	Test Case for User Interface Consistency.
<b>Procedure of testing steps</b>	<ul style="list-style-type: none"> <li>• Navigate to different pages on the website.</li> <li>• Verify that the user interface (UI) design elements (colors, fonts, icons, etc.) are consistent across all pages.</li> </ul>
<b>Expected results/outcome</b>	<ul style="list-style-type: none"> <li>• The user interface should be consistent across all pages, providing a cohesive and professional look and feel.</li> </ul>
<b>Priority/Severity</b>	Minor
<b>Date Tested</b>	May 17, 2023
<b>Test Result</b>	Passed

### 5.3. Mean Reciprocal Rank Test Cases

Mean Reciprocal Rank (MRR) is a widely used evaluation metric in the field of information retrieval and machine learning. In this section, we will use MRR to evaluate the performance of our machine learning model and compare it against other models or baseline approaches [12].

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i}.$$

Fig. 7. Mean reciprocal rank formula.

After 5 different test cases for the ML model, by mean reciprocal rank, we obtained 3 out of these cases that resulted with the best success, while one of them gave the correct issue in the 2nd option, and one of them gave the correct issue in the 6th option. Therefore, we obtained the mean reciprocal rank as  $(\frac{1}{6} + 1 + 1 + \frac{1}{2} + 1)/5 = 22/30$ , or about 0.73.

<b>Test Case ID</b>	#MRR-01
<b>Test Type/Category</b>	Mean Reciprocal Rank
<b>Given Data</b>	<ul style="list-style-type: none"> <li>• Pull request ID: 07</li> <li>• Pull request name: Implement retrieval methods.</li> </ul>



<b>Expected link</b>	<ul style="list-style-type: none"> <li>Issue ID: 10002</li> <li>Issue name: Create Data Retrieval Service for Azure.</li> </ul>
<b>Resulting links</b>	<ol style="list-style-type: none"> <li>(10004, 0.6538699070958309)</li> <li>(10043, 0.5920638481740015)</li> <li>(10006, 0.5867772714700163)</li> <li>(10061, 0.5657587874506174)</li> <li>(10059, 0.5537630073462833)</li> <li><b>(10002, 0.5424273213403735)</b></li> <li>(10056, 0.5096294871954614)</li> </ol>
<b>Rank</b>	6
<b>Reciprocal rank</b>	1/6
<b>Priority/Severity</b>	Major
<b>Date Tested</b>	March 13, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#MRR-02
<b>Test Type/Category</b>	Mean Reciprocal Rank
<b>Given Data</b>	<ul style="list-style-type: none"> <li>Pull request ID: 10</li> <li>Pull request name: Add branch information to PR model</li> </ul>
<b>Expected link</b>	<ul style="list-style-type: none"> <li>Issue ID: 10047</li> <li>Issue name: Branch information should exist in PR data</li> </ul>
<b>Resulting links</b>	<ol style="list-style-type: none"> <li><b>(10047, 0.800569248490578)</b></li> <li>(10033, 0.5269733242444746)</li> <li>(10054, 0.5238443933565953)</li> <li>(10050, 0.4821396587077893)</li> <li>(10019, 0.4798525475325077)</li> <li>(10004, 0.47978947211175793)</li> <li>(10045, 0.4787526031388459)</li> </ol>
<b>Rank</b>	1
<b>Reciprocal rank</b>	1
<b>Priority/Severity</b>	Major
<b>Date Tested</b>	March 13, 2023

<b>Test Result</b>	Passed
--------------------	--------

<b>Test Case ID</b>	#MRR-03
<b>Test Type/Category</b>	Mean Reciprocal Rank
<b>Given Data</b>	<ul style="list-style-type: none"> <li>● Pull request ID: 01</li> <li>● Pull request name: Initialize Django project on backend</li> </ul>
<b>Expected link</b>	<ul style="list-style-type: none"> <li>● Issue ID: 10001</li> <li>● Issue name: Initialize Django project on backend</li> </ul>
<b>Resulting links</b>	<ol style="list-style-type: none"> <li>1. <b>(10001, 1.0)</b></li> <li>2. (10023, 0.7692845430370506)</li> <li>3. (10061, 0.5951890946975835)</li> <li>4. (10041, 0.5625866478927344)</li> <li>5. (10059, 0.5554017555065811)</li> <li>6. (10048, 0.5519805606736923)</li> <li>7. (10003, 0.5513433182599485)</li> </ol>
<b>Rank</b>	1
<b>Reciprocal rank</b>	1
<b>Priority/Severity</b>	Major
<b>Date Tested</b>	March 13, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#MRR-04
<b>Test Type/Category</b>	Mean Reciprocal Rank
<b>Given Data</b>	<ul style="list-style-type: none"> <li>● Pull request ID: 13</li> <li>● Pull request name: Create Possible Link model</li> </ul>
<b>Expected link</b>	<ul style="list-style-type: none"> <li>● Issue ID: 10054</li> <li>● Issue name: Create PossibleLink database model</li> </ul>
<b>Resulting links</b>	<ol style="list-style-type: none"> <li>1. (10050, 0.6267598773665553)</li> <li>2. <b>(10054, 0.6061555594382552)</b></li> <li>3. (10033, 0.5803833028086245)</li> <li>4. (10031, 0.5692814990365802)</li> <li>5. (10032, 0.5505409504680864)</li> </ol>

	6. (10030, 0.5045170025238684) 7. (10019, 0.5033349571558927)
<b>Rank</b>	2
<b>Reciprocal rank</b>	1/2
<b>Priority/Severity</b>	Major
<b>Date Tested</b>	March 13, 2023
<b>Test Result</b>	Passed

<b>Test Case ID</b>	#MRR-05
<b>Test Type/Category</b>	Mean Reciprocal Rank
<b>Given Data</b>	<ul style="list-style-type: none"> <li>• Pull request ID: 09</li> <li>• Pull request name: Add code formatter to backend project</li> </ul>
<b>Expected link</b>	<ul style="list-style-type: none"> <li>• Issue ID: 10023</li> <li>• Issue name: Add code formatter to backend project</li> </ul>
<b>Resulting links</b>	1. ( <b>10023, 0.9999999999999999</b> ) 2. (10001, 0.7692845430370506) 3. (10059, 0.6877586067003254) 4. (10061, 0.6394603342250512) 5. (10057, 0.6075983829301943) 6. (10020, 0.5734452453783456) 7. (10063, 0.5652294032663305)
<b>Rank</b>	1
<b>Reciprocal rank</b>	1
<b>Priority/Severity</b>	Major
<b>Date Tested</b>	March 13, 2023
<b>Test Result</b>	Passed

## 6. Maintenance Plan and Details

To ensure the continuous and efficient operation of ReLink, it is necessary to have a robust maintenance plan in place. This plan will allow for the regular correction and

improvement of ReLink to deliver an optimal user experience, as well as assure the maintenance of the underlying infrastructure for high-quality, automated PR-issue linkage. The maintenance plan for ReLink includes the following components:

### **Regular Updates**

Routine updates are crucial to keep ReLink performing at its best and to respond to user feedback or industry changes. These updates will include improvements to the user interface, increasing ML algorithm efficiency and accuracy, and adding new features or filters for the visualization of the analysis.

### **Security Patches**

Security is a top priority for ReLink. Regularly scheduled security patches will be applied to protect sensitive data and prevent unauthorized access. Security audits will also be performed to identify any potential vulnerabilities and address them promptly. This is very important since ReLink handles sensitive customer information.

### **Data Backup and Recovery**

To ensure data integrity, regular backups will be performed. Additionally, a comprehensive data recovery plan will be in place to minimize any potential data loss and maintain the continuity of service in the event of unexpected circumstances.

### **Performance Monitoring**

To maintain a high-quality user experience, system performance will be monitored consistently. This includes tracking system load, response time, and other key performance indicators (KPIs) to ensure ReLink runs efficiently.

### **User Support and Documentation**

Providing comprehensive support and up-to-date documentation is essential for users to effectively utilize ReLink. The maintenance plan will involve regular updates to the user manual and FAQs, alongside timely responses to user queries. It is also essential that the algorithm behind ReLink remains understandable for the customers as it evolves in order to preserve transparency and trust.

### **Regular Reviews and Audits**

Regular reviews and audits of both the codebase and infrastructure will be conducted to ensure that best practices are being followed and to identify areas for improvement or optimization.

In terms of maintenance details, ReLink's maintenance will be handled by our team members, and the proposed plan will periodically be checked and updated according to changing needs.

## 7. Other Project Elements

### 7.1. Consideration of Various Factors in Engineering Design

There is no external factor except for the economic factor limiting us during the development of ReLink. We need to consider some economic factors during the design phase of the project. These economic factors are database costs and server costs, respectively. Since we will be doing repository mining, we need a database where we can store the data of the project with any scale from GitHub, Jira, or Azure DevOps. This database will require a substantial budget. Moreover, we may need external CPUs to train our models, so we will need more performance than a normal web application. This will increase server costs.

Factor	Effect(s)	Level of Effect (0-10)
Public Health	Has no effect	0
Safety	Has no effect	0
Global	Has no effect	0
Cultural	Has no effect	0
Social	Has no effect	0
Environmental	Has no effect	0
Economic	Database Costs Server Costs	7

### 7.2. Ethics and Professional Responsibilities

We give importance to privacy by design which is a system design that takes into account the data protection at the beginning of the design and architecture of the application. While ReLink mines the repositories of the open-source projects, it has to access the pull request and issue data of the projects on GitHub, Azure DevOps, and Jira. ReLink uses data mining to match PRs and issues. To protect such data, ReLink never publishes its database contents with third parties, along with never writing on sensitive project metadata other than the necessary modifications for linkage.

## 7.3. Teamwork Details

### 7.3.1. Contributing and functioning effectively on the team

**Ayberk Yaşa:** Writing reports, implementation of REST API, implementation of data retrieval, implementation of graph builder modules, regularly attending meetings.

**Cemhan Kaan Özaltan:** Writing reports, implementation of REST API, implementation of ML modules, regularly attending meetings.

**Çağatay Şafak:** Writing reports, sketching UI in Figma, implementation of frontend, regularly attending meetings.

**Fatih Kaplama:** Writing reports, sketching UI in Figma, implementation of frontend, regularly attending meetings.

**Görkem Ayten:** Writing reports, implementation of REST API, regularly attending meetings.

### 7.3.2. Helping creating a collaborative and inclusive environment

**Ayberk Yaşa:** Demonstrated an encouraging and critical approach in meetings.

**Cemhan Kaan Özaltan:** Demonstrated an encouraging and critical approach in meetings.

**Çağatay Şafak:** Demonstrated an encouraging and critical approach in meetings.

**Fatih Kaplama:** Demonstrated an encouraging and critical approach in meetings.

**Görkem Ayten:** Demonstrated an encouraging and critical approach in meetings.

### 7.3.3. Taking lead role and sharing leadership on the team

**Ayberk Yaşa:** Took overall leadership in the project management and task distribution along with retrieval modules.

**Cemhan Kaan Özaltan:** Took leadership in the decision making and implementation of ML related modules.

**Çağatay Şafak:** Took leadership in the UI/UX design and decision making phases of the project.

**Fatih Kaplama:** Took leadership in the front-end implementation and decision making stages.

**Görkem Ayten:** Took leadership in the backend implementation, modeling, and decision making.

### 7.3.4. Meeting objectives

Throughout the project, we worked towards achieving the objectives and milestones outlined in our project plan. In this section, we will assess the level of completion for each objective and discuss any deviations from the original timeline.

#### 7.3.4.1. Requirements Report

**Objective:** Complete the Requirements Report by the first half of November.

**Level of Completion: Met**

We successfully met the objective of delivering the Requirements Report within the specified timeframe. By conducting thorough discussions with stakeholders and gathering information, we were able to outline the project requirements comprehensively, providing a clear understanding of the project requirements.

**7.3.4.2. UI Implementation**

**Objective:** Complete the UI Implementation between the second half of November and the end of December.

**Level of Completion: Met**

Our team successfully executed the UI Implementation stage within the designated timeline. We translated the project requirements into an intuitive and visually appealing user interface, ensuring a comfortable user experience. By using our skills in frontend development, we achieved the desired UI implementation.

**7.3.4.3. Backend Implementation**

**Objective:** Complete the Backend Implementation between the second half of November and the first half of January.

**Level of Completion: Met**

The Backend Implementation stage was carried out within the expected timeframe. Leveraging our knowledge in backend development, we successfully implemented the necessary functionalities, ensuring effective data handling, efficient processing, and effective integration with the frontend components. The completed backend implementation enabled the prediction module to be integrated into ReLink.

**7.3.4.4. Design Report**

**Objective:** Complete the Design Report between the second half of December and the first half of February.

**Level of Completion: Met**

We met the objective of delivering the Design Report within the allocated time frame. Through careful analysis and consideration, we documented the system architecture, design choices, and implementation details in the Design Report. This report served as a reference for the development process, ensuring a clear understanding of the project's technical aspects and guiding the next stages.

**7.3.4.5. ML Implementation**

**Objective:** Complete the ML Implementation between the second half of January and the end of March.

**Level of Completion: Met**

The ML module was implemented in the planned time frame. While we started the ML Implementation stage as planned, we encountered several difficulties during its implementation and optimization due to its novel and complex nature. Despite the encountered difficulties, we successfully implemented the ML components within the

application. The extensive research and optimization allowed us to fine-tune the algorithms, resulting in improved accuracy and performance. As a result, the ML-based link recovery module now acts as the engine behind ReLink's main functionality.

#### **7.3.4.6. Testing**

**Objective:** Complete the Testing stage between the second half of March and the end of April.

**Level of Completion:** Partially met

Although we initiated the Testing stage on time, we experienced some delays that extended the completion until the first half of May. Nonetheless, we devoted significant effort to thoroughly testing the application, including unit testing, integration testing, and user acceptance testing. These comprehensive testing measures ensured the stability, reliability, and quality of the final product.

#### **7.3.4.7. Case Study with Arçelik**

**Objective:** Conduct a Case Study with Arçelik between the second half of April and the end of April.

**Level of Completion:** Partially met

We successfully introduced our ML-based web application to Arçelik within the specified timeframe, enabling them to gain an understanding of its capabilities. However, due to the need for additional preparation and coordination, Arçelik will be utilizing our application in their business in the second half of May. This slight delay in the case study does not significantly impact the overall success of the project, as it remains within an acceptable range.

#### **7.3.4.8. Final Report**

**Objective:** Complete the Final Report between the first half of May and the end of May.

**Level of Completion:** Partially met

As of now, we are working on the final report, which will hopefully include an overall explanation of ReLink and its development.

### **7.4. New Knowledge Acquired and Applied**

In the process of developing our ML-based web application, we recognized the need to acquire new knowledge and learning strategies to effectively implement ML techniques. While our team possessed extensive experience in web application development, none of us had prior experience in developing such an ML-based application. In order to bridge this knowledge gap, we followed both theoretical and practical learning paths. This allowed us to solve the problem at hand appropriately.



### **7.4.1. Researching ML-Based Applications**

To gain insights into ML-based application development, we conducted an in-depth review of relevant literature. This involved studying conference papers and journal articles focused specifically on ML-based link recovery [11, 12, 13]. By investigating these sources, we aimed to understand the underlying principles, challenges, and best practices associated with integrating machine learning algorithms with the problem at hand. This research enabled us to familiarize ourselves with the state-of-the-art tools, algorithms, and methodologies employed in this domain. The main purpose of this research, in short, was to determine the necessary prediction architecture that we could implement.

### **7.4.2. Exploring ML-Based Tools and Algorithms**

Building on our research findings, we dedicated significant efforts to exploring ML-based tools and algorithms that could be leveraged within our web application. Through extensive experimentation and evaluation, we identified suitable algorithms and frameworks that aligned with our project requirements. Through these tools, such as Scikit-learn and Gensim, we were able to implement link recovery.

### **7.4.4. Dockerization of the Web Application**

In addition to acquiring ML-specific knowledge, we also expanded our expertise in deploying and managing web applications through containerization. We learned how to dockerize a web application consisting of multiple containers, ensuring its portability and scalability. By employing Docker, we effectively separated the application components, such as the ML model, the web server, the database, Celery, and Redis into individual containers. This allowed us to deploy the application on a remote server, simplifying the deployment process and enhancing the overall robustness and efficiency of our ML-based web application.

### **7.4.5. Integration of Acquired Knowledge**

The knowledge acquired through our research, exploration of ML-based tools, learning deep learning concepts, and Dockerizing the web application collectively empowered our team to successfully develop and deploy a robust ML-based web application. By applying our newfound understanding of ML principles and leveraging state-of-the-art frameworks, we were able to implement useful features and functionality, enhancing the overall user experience. Moreover, the Dockerization process facilitated seamless deployment and scalability, ensuring the smooth operation of the application on a remote server.

Through our proactive approach to acquiring and applying new knowledge, we have significantly expanded our skill set, enabling each of our team members to expand their knowledge base.

## 8. Conclusion and Future Work

In conclusion, ReLink has achieved significant metrics on open-source repositories like WildFly [10], such as a recall of 0.85 for the 5 recommendations per PR case. In addition, ReLink has also proved its commercial viability through a case study with a real-life customer, namely Arçelik. Considering these factors and the professional development process of ReLink, we hope that it may one day become a widely-used software development tool. For future work, we aim to reach a larger user base, along with adding new features such as support for other services such as GitHub issues and Azure DevOps bug tracking platforms in addition to their version control systems (VCSs) that are already supported. We also aim to support linking commits with issues, in addition to supporting real-time linkage during PR merging with a browser extension.

## 9. Glossary

**Issue-PR / Issue-Commit Link:** Semantic associations created between the given software artifacts, either manually or by the ReLink system.

**Commit:** A change on a file, or a set of files, in a version control system repository.

**Pull Request (PR):** The event of merging new code to the code repository in software development.

**Issue:** Descriptions of needed changes and tasks on a software project that allow tracking progress.

**Traceability:** The capacity to explicitly relate associated software artifacts throughout development [12].

## 10. References

- [1] T. W. Aung, H. Huo, and Y. Sui, "A literature review of automatic traceability links recovery for software change impact analysis," *Proceedings of the 28th International Conference on Program Comprehension*, 2020.
- [2] M. Rath, J. Rendall, J. L. Guo, J. Cleland-Huang, and P. Mäder, "Traceability in the wild," *Proceedings of the 40th International Conference on Software Engineering*, 2018.
- [3] C. M. Lüders, T. Pietz, W. Maalej. "Automated Detection of Typed Links in Issue Trackers". *2022 IEEE 30th International Requirements Engineering Conference (RE)*, 2022.
- [4] S. Liu, Y. Wang, X. Yang, B. Lei, L. Liu, S. X. Li, D. Ni, and T. Wang, "Deep Learning in medical ultrasound analysis: A Review," *Engineering*, vol. 5, no. 2, pp. 261–275, 2019. [Online]. Available: <https://doi.org/10.1016/j.eng.2018.11.020>. [Accessed: 17-May-2023].
- [5] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv*, vol. 1301, no. 3781, 2013.
- [6] "Comparison of different word embeddings on text similarity-a use case in NLP," *Medium*, Oct. 2019. [Online]. Available: <https://intellica-ai.medium.com/comparison-of-different-word-embeddings-on-text-similarity-a-use-case-in-nlp-e83e08469c1c>. [Accessed: 10-Mar-2023].
- [7] O. Levy and Y. Goldberg, "Dependency-based word embeddings," *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2, pp. 302–308, 2014.
- [8] "Word2vec-google-news-300," *Hugging Face*. [Online]. Available: <https://huggingface.co/fse/word2vec-google-news-300>. [Accessed: 05-Mar-2023].
- [9] S. Glen, "Jaccard index / similarity coefficient," Statistics How To, <https://www.statisticshowto.com/jaccard-index/>. [Accessed: 18-May-2022].
- [10] "Wildfly," *WildFly*, <https://www.wildfly.org/>. [Accessed: 18-May-2022].
- [11] T. W. Aung, H. Huo, and Y. Sui, "A literature review of automatic traceability links recovery for software change impact analysis," *Proceedings of the 28th International Conference on Program Comprehension*, 2020.
- [12] M. Rath, J. Rendall, J. L. Guo, J. Cleland-Huang, and P. Mäder, "Traceability in the wild," *Proceedings of the 40th International Conference on Software Engineering*, 2018.
- [13] C. M. Lüders, T. Pietz, W. Maalej. "Automated Detection of Typed Links in Issue Trackers". *2022 IEEE 30th International Requirements Engineering Conference (RE)*, 2022.