
# Bilkent University
# Department of Computer Engineering

# Senior Design Project
# T2404
# Compedia

# Analysis and Requirement Report

Yaşar Tatlıcıoğlu - 22003856 - yasar.tatlicioglu@ug.bilkent.edu.tr
Serhat Yılmaz - 22002537 - serhaty@ug.bilkent.edu.tr
Bartu Albayrak - 22101640 - bartu.albayrak@ug.bilkent.edu.tr
Anıl Altuncu - 21901880 - anil.altuncu@ug.bilkent.edu.tr
Ece Beyhan - 22003503 - ece.beyhan@ug.bilkent.edu.tr

## Supervisor: Uğur Doğrusöz
## Course Instructors: Atakan Erdem, Mert Bıçakçı

**16 December 2024**

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

## Contents

# Analysis and Requirement Report

*T2404: Compedia*

## 1 Introduction

Nowadays, many companies offer similar products, services, or solutions, making it harder for businesses to find the relevant partner or the right service provider that matches their specific needs. A common way of overcoming this problem is using the internet to search for companies that fulfill the needs of businesses. However, this method is often time-consuming and results in a need for more relevant data. Businesses face many challenges in standing out and finding the right partners. Advertisements are often not effective due to the crowdedness of the market. Therefore, for smaller businesses with limited budgets, competing with others is nearly impossible. Moreover, the visibility of these companies is another problem. Search engines promote well-established companies more, making it nearly impossible for smaller companies to be noticed. The credibility of promoted companies is another area for improvement in today's competitive market. Businesses sometimes choose companies with fake reviews or biased information, resulting in unsuccessful partnerships. There can be an overwhelming number of irrelevant results using search engines. These challenges cause a decrease in the effectiveness of the decision-making process and waste valuable partnerships.

## 2 Current System

Currently, some businesses are searching to find proper partners. On the other hand, other businesses are trying to introduce themselves by providing their services. There are some platforms such as Clutch, GoodFirms, Built In, and LinkedIn. These platforms mainly focus on keyword-based search algorithms, static filters, or limited user reviews, which mostly result in not finding the optimal or perfect match. While platforms like Clutch and GoodFirms categorize companies only by static filters like industry, service type, and maybe location, they lack the ability to understand the client’s needs and results in offering generic results. Despite the platform's professionality, LinkedIn provides opportunities to connect, but the solutions are aimed more toward individual networking rather than networking for businesses.

Some platforms are trying to attempt to innovate. For instance, Y Combinator’s Startup Directory under the Companies bar provides many startups as a list but remains limited to match with keyword-based search. Company pages are built well and include comprehensive information, but with the help of artificial intelligence can be improved. Similarly, tools like Clutch’s review and portfolio features help verify credibility but, again lack of providing semantic understanding.

## 3 Proposed System

### 3.1 Overview

Compedia aims to provide a connection way for businesses and advance through a platform designed to meet modern business-to-business needs. Serving as an Inference Engine, Compedia makes use of Large Language Models (LLMs) like GPT or BERT to create highly detailed, structured, and comprehensive profiles for companies. The platform will enable businesses to include foundational details such as location, founding year, employee count,

overall score, and more complex information like offered services, industrial specialties, and past collaborations.

Compedia uses Semantic Search to match user queries with the most relevant companies, understanding the intent and context of the search rather than relying on exact keyword-based search matches. For example, a query like “AI-driven solutions for logistics and supply chain management in Europe” would give relevant results.

The other usage of the platform is its focus on provider companies as key actors. Compedia allows these companies to promote their services, increase visibility, and compete in the market effectively. Since the companies with greater overall scores are promoted more, companies will be eager to develop efficiently.

Project management is another important feature of this system. Compedia ensures facilitating collaboration between clients and provider companies. While clients can track progress, and update their needs, providers manage completion and service delivery. The platform cares about transparency throughout the project lifecycle. Providers can mark projects as completed, and clients can evaluate performance through scores and comments, which influence provider visibility and accountability. On the other hand, other clients can view this feedback, offering them a detailed first impression of the provider company. This allows clients to increase their chances of selecting the optimal partnership.

## 3.2 Functional Requirements

### 3.2.1 User

- The user can register/login with email and password.
- The user can reset their password.
- Users can view user profiles registered with the same company.
- The user can search for company profiles using semantic queries that are processed with modern LLM and NLP techniques and view those profiles.
- The user can view and edit their profile.
- The users with an approved email address of the company can create their company account and are assigned as the root user.
- The root user, the creator of the company page, has all permissions such as view, edit, add user and remove permission from other users.
- The root user, the creator of the company page, can add (invite) other users to the company page by specifying each user’s permissions.
- The user with the “view” permission can only view the company profile.
- The user with the “edit” permission can both view and update the company profile.
- The user with the “add user” permission can both view and add other users to the company profile.

- The root user, the creator of the company page, can modify or remove permissions assigned to other users.
- Users can upload company profile information in standard formats like JSON and CSV.
- The client company that searches for a provider company for its project can submit project details to the platform.
- The provider company can review project details and decide to accept or reject a project.
- The provider company can update or delete project details if necessary.
- The provider company can mark a project as completed once the work is finalized.
- The client company can submit a grade for the project, including a score and comments/feedback.
- The provider company can confirm the grade submission received from the client company.

### 3.2.2 System

- The system will approve the company email address of the root user.
- The system will approve the initial company information that is provided by the root user.
- The platform uses Large Language Models (LLMs) and vector databases to:
    - Process semantic search queries.
    - Recommend companies matching the user's query specifications.
- Results are returned based on meaning rather than exact keyword matches.
- The system ensures secure management of user roles and permissions:
    - Root users can manage roles (view-only, edit, add user) for company profiles.
    - Permissions determine actions allowed for each user role.
- The system can add company pages directly without connecting a root user for those pages. (mentioned in the progress meeting about how company data will be obtained)
- User data and company data are encrypted and stored securely in compliance with GDPR.
- Access to sensitive features is restricted based on authentication and role-based permissions.
- The system saves submitted project details to the relational database.
- The system notifies the provider company when a new project is assigned.

- When the provider company accepts the project, the system updates the status to **"Accepted"**.
- When the provider company rejects the project, the system updates the status to **"Rejected"**.
- The system sends notifications to the client company based on the provider company's acceptance or rejection decision.
- The system processes any updates or deletions made by the provider company and updates the project record in the database.
- The system updates the project status to **"Completed"** when the provider company marks it as completed.
- The system saves the grade submitted by the client company, including score and comments, and updates the project's grade record.
- The system notifies the provider company that the client has submitted a grade for the project.

## 3.3 Non-functional Requirements

### 3.3.1 Usability

Compedia will be a website. Therefore, Compedia can be reachable from anywhere with an internet connection to the web. The users can easily use our website. As an example, before login, users enter their email addresses and if the email address exists in our database, the user will be redirected to the login page, otherwise the user will be redirected to the register page. The website will have a user-friendly interface accessible to each user. Our overall website will have a responsive design and function well on each device with an internet connection. The website has a sidebar so that the user can easily navigate to where they are looking for. Search button will always be in the topbar which facilitates the user's job as the main feature is the searching for the company pages and user profiles. All in-progress requests are visible to the user with a loading indicator.

### 3.3.2 Reliability

The company data that is initially entered by the root user should be approved by the system. The company data that will be added by the system will be also reliable since the company data will be obtained from reliable resources as discussed in progress meetings. The company

information that the editor user of the relevant company page updated should be approved by the root user. Since enhanced semantic search should return results that are relevant to the specifications in the query, the search function should always work without any irrelevant results. Data backups will be performed weekly, we will store the updated company and user information to our databases.

### 3.3.3 Performance

- The system should respond to users within 2 seconds under normal load conditions.
- The website should support at least 10000 users simultaneously without significant performance degradation.
- AI-powered search queries should return results within 5 seconds, even for complex searches involving hybrid search techniques.
- As we will implement a check for email existence at the initial page of the website, the load for checking the email existence will be handled at first and will initially reduce the workload of the system.

### 3.3.4 Interoperability

- The website will integrate seamlessly with external AI services (e.g., GPT, BERT) and databases.
- The user that will create the company page can import files in standard formats (e.g. JSON, CSV) for company profile information.
- The website will be compatible with popular web browsers (e.g., Chrome, Firefox, Safari, Edge) and their latest two versions.

### 3.3.5 Scalability

New users can easily enter the website by logging in from their browsers. Therefore, our backend servers, databases and LLM integration must work to serve nearly 10000 users concurrently. Moreover, adding new features to the website won't affect the working of other systems since we will use Kafka to integrate our servers and let independent development of these servers. Since Milvus is the leading option in terms of scaling among vector databases, we will scale efficiently to accommodate up to 10 million company profiles [1].

### 3.3.6 Extensibility

- The architecture of the website will support modularity, which will allow new features (e.g., user profiles, semantic search, and company profiles) to be developed, updated, or replaced without affecting other modules.

- REST API will increase the extensibility as it will give developers an opportunity to add new functionalities such as third party libraries.
- Also, the AI models (GPT, BERT, etc.) could be changed to enable future upgrades without causing excessive downtime.
- Application will also allow extra AI models or services to increase the range of features such as predictive analytics.
- Addition of new user roles with associated permissions will be supported to allow for future expansion without major changes to the codebase.
- We will use Kafka for maintenance of data pipelines to enable real-time data flow and processing which will simplify the addition of new data sources or services.

### 3.4 Pseudo Requirements

- GitHub will be used as the main Version Control platform.
- Trello will be used for project task management and team collaboration.
- React will be used for front-end development
- ASP.NET Core will be used in the backend side of the application.
- For data management and store, MicrosoftSQL will be used.
- The platform will be hosted on Azure.
- Milvus, a vector database, will store processed complex data to facilitate semantic search.
- Kafka will be integrated to enable efficient, real-time processing and communication between microservices.

### 3.5 System Models

#### 3.5.1 Scenarios

| Scenario 1 | Sign in |
|---|---|
| Actor | Registered User |
| Flow of Events | 1. The actor enters their email address.<br>2. The system determines that email is registered in the database<br>3. The actor enters their password. |

| | |
|---|---|
| | 4. The system checks the information provided.<br>5. If the password is invalid, an error is displayed. |
| Entry Condition(s) | The actor opens the application |
| Exit Condition(s) | The actor is logged in and redirected to their dashboard. |

| | |
|---|---|
| Scenario 2 | Sign Up |
| Actor | Unregistered User |
| Flow of Events | 1. The actor enters their email address<br>2. The system determines that email is not registered in the database<br>3. The actor enters their personal information.<br>4. The system validates the provided information and creates a new user. |
| Entry Condition(s) | The actor opens the application |
| Exit Condition(s) | The user is successfully registered and can proceed to log in. |

| | |
|---|---|
| Scenario 3 | Creating New Company Profile |
| Actor | Root user |
| Flow of Events | 1. The actor clicks the “Create Company” in the navigation menu.<br>2. The actor fills the form with the necessary information such as Company Name, Foundation Year and Address.<br>3. The system validates the details and creates a new company.<br>4. The actor is automatically assigned as the root user of the company |
| Entry Condition(s) | The actor is logged in.<br>The actor clicks on the “Create Company” button. |
| Exit Condition(s) | The actor clicks the back button or closes the application.<br>A new company profile is created and then navigates to the company profile. |

| Scenario 4 | Adding User to Company |
|---|---|
| Actor | Root User |
| Flow of Events | 1. The actor navigates to the “Manage Users” section under the company profile and clicks the “Add User” button.<br>2. The actor enters the new user’s email while specifying the permission given.<br>3. The system adds the user to the company and assigns the specified permission to the user.<br>4. Company size has increased by one.<br>5. The system sends a notification to the newly added user. |
| Entry Condition(s) | The actor is logged in.<br>The actor has authorization to add users to a company. |
| Exit Condition(s) | The user is added to the company with the specified permissions. |

| Scenario 5 | Edit User Claims |
|---|---|
| Actor | Root User |
| Flow of Events | 1. The actor navigates to the “Manage Users” section under the company profile and clicks the “Edit Claims” button.<br>2. The actor selects a user from the list.<br>3. The actor assigns claims to a user such as “view”, “edit”, “add user” by checking the radio buttons.<br>4. The system updates the claims of the user. |
| Entry Condition(s) | The actor is logged in.<br>The actor has authorization to edit user claims. |
| Exit Condition(s) | User claims are updated. |

| | |
|---|---|
| Scenario 6 | View Company Profile |
| Actor | All Users(Case 1 and 2), Company Users(Case 2) |
| Flow of Events | Case 1: Search Bar<br>1. The actor enters the company name or related keyword in the search bar<br>2. The system shows the matching companies and displays the results in a list.<br>3. The actor clicks onto a company from the list to view the company profile.<br>4. The actor is redirected to the corresponding company profile.<br>Case 2: Direct View<br>1. The actor clicks the company profile from the navigation menu.<br>2. The actor is redirected to the company profile. |
| Entry Condition(s) | The company exists in the system. |
| Exit Condition(s) | The actor views the selected company's profile. |

| | |
|---|---|
| Scenario 7 | View User Profile from Company |
| Actor | Root User |
| Flow of Events | 1. The actor navigates to the “Manage Users” section under the company profile and clicks the “View Users” button.<br>2. The system shows the list of the enrolled users of the company.<br>3. The actor clicks on top of a user that is desired.<br>4. The actor navigates to the user’s profile. |

| | |
|---|---|
| Entry Condition(s) | The actor is authenticated. |
| Exit Condition(s) | The actor views the selected user's profile. |

| | |
|---|---|
| Scenario 8 | Adding Company Project to Profile |
| Actor | Root User, Company User |
| Flow of Events | 1. The actor navigates to the "Company Projects" section either from the navigation menu or the company profile.<br>2. The system shows the list of the company projects completed.<br>3. The actor can add a new project to a list manually by specifying the details of the project.<br>4. The actor can determine which projects will be displayed on the profile by checking the radio boxes. |
| Entry Condition(s) | The actor has edit permission from that company. |
| Exit Condition(s) | Desired projects are shown on the company profile. |

| | |
|---|---|
| Scenario 9 | Editing User Profile |
| Actor | All Users |
| Flow of Events | 1. The actor navigates to their profile page.<br>2. The actor clicks the "Edit Profile" button.<br>3. The actor updates fields such as name, phone number, or services.<br>4. The actor clicks to save the changes.<br>5. The system checks the validity of the inputs.<br>6. If the inputs are in the right format, the system updates the user profile. Else an error message will be displayed.<br>7. The actor is navigated to their profile page. |

| | |
|---|---|
| Entry Condition(s) | The actor is logged in. |
| Exit Condition(s) | The profile page is updated and shown. |

| | |
|---|---|
| Scenario 10 | Submitting Project Details |
| Actor | Client Company |
| Flow of Events | 1. The actor clicks the add project button on their company profile.<br>2. The actor enters the details of the project.<br>3. The entered provider company information will wait for approval but the remaining details don't need approval.<br>4. The system sends a notification to the provider company. |
| Entry Condition(s) | The actor's company exists in the system.<br>The actor must have the related permission to submit the project details. |
| Exit Condition(s) | The actor clicks the save, cancel, or back button and returns to the company profile. |

| | |
|---|---|
| Scenario 11 | Accepting Projects |
| Actor | Provider Company |
| Flow of Events | 1. The actor clicks on the notifications and accepts the corresponding project request.<br>2. The system sends a notification to the collaborator company.<br>3. The system adds the project to the project list of both of the collaborating companies as unfinished. |
| Entry Condition(s) | The project is not created by the actor. |
| Exit Condition(s) | The actor clicks on the accept icon or anywhere except the notification box. |

| Scenario 12 | Rejecting Projects |
|---|---|
| Actor | Provider Company |
| Flow of Events | 1. The actor clicks on the notifications and rejects the corresponding project request.<br>2. The system sends a notification to the collaborator company. |
| Entry Condition(s) | The project is not created by the actor. |
| Exit Condition(s) | The actor clicks on the reject icon or anywhere except the notification box. |

| Scenario 13 | Updating Project Details |
|---|---|
| Actor | Client Company, Provider Company |
| Flow of Events | 1. The actor enters their company profile and clicks on the edit project for the specific project.<br>2. The actor edits the project details. |
| Entry Condition(s) | The actor's company exists in the system.<br>The actor must have the related permission to edit the project details. |
| Exit Condition(s) | The actor clicks the save, cancel, or back button and returns to the company profile. |

| Scenario 14 | Editing Company Profile |
|---|---|
| Actor | Company Users |
| Flow of Events | 1. The actor clicks the company profile from the navigation menu.<br>2. The actor is redirected to the company profile.<br>3. The actor clicks to edit the company profile button.<br>4. The actor edits the company profile and saves the changes. |
| Entry Condition(s) | The company exists in the system.<br>The actor has edit permission from that company. |
| Exit Condition(s) | The actor clicks the save, cancel, or back button and returns to the company profile. |

| Scenario 15 | Searching Company |
|---|---|
| Actor | All Users |
| Flow of Events | 1. The actor clicks the search button on the top bar or on the homepage.<br>2. The actor enters the company name or related skill-based keywords into the search bar.<br>3. The system returns the results of the provided search.<br>4. The user can see the company cards that they searched for. |
| Entry Condition(s) | The actor opens the application |
| Exit Condition(s) | The actor clicks the back button or closes the application. |

| Scenario 16 | Submitting Project Grade |
|---|---|
| Actor | Client Company |
| Flow of Events | 1. The actor clicks on the company profile.<br>2. The actor views the finished project in their company profile.<br>3. The actor clicks the project that they want to grade.<br>4. The actor grades the provider company for that project. |
| Entry Condition(s) | The company exists in the system.<br>The actor has edit permission from that company. |
| Exit Condition(s) | The actor clicks the save, cancel, or back button and returns to the company profile. |

### 3.5.2 Use-Case Model

#### 3.5.2.1 User Management UML Use Case Diagram

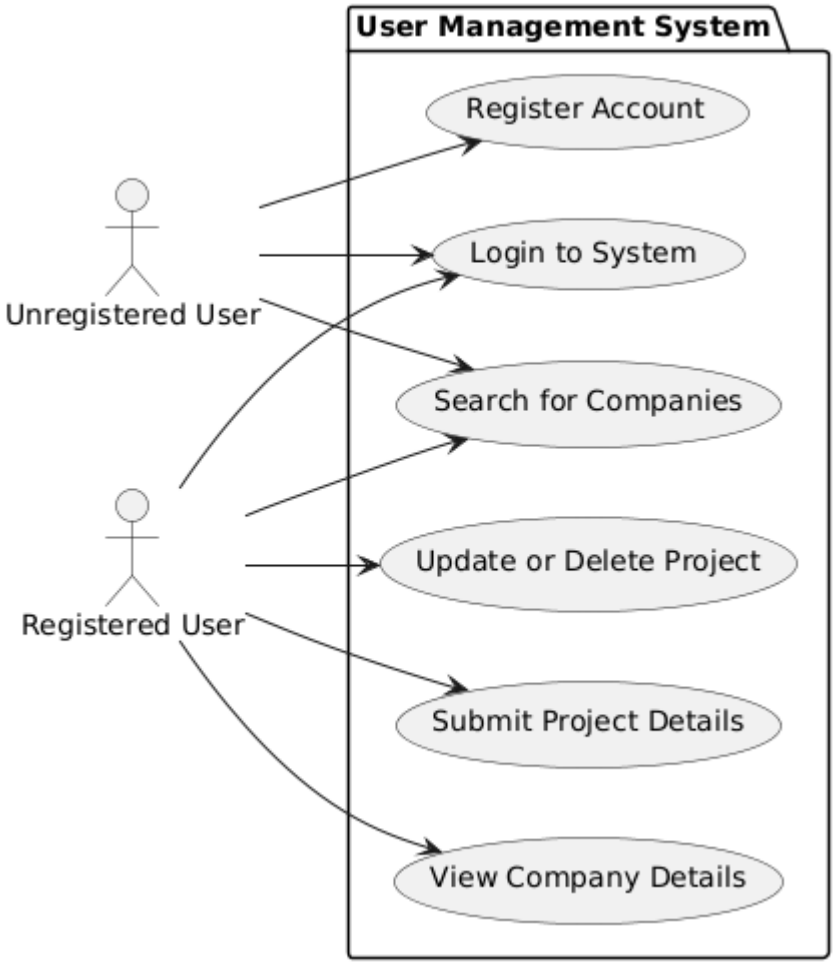


Fig.1 User Management Use Case Diagram

### 3.5.2.2 Company Management UML Use Case Diagram

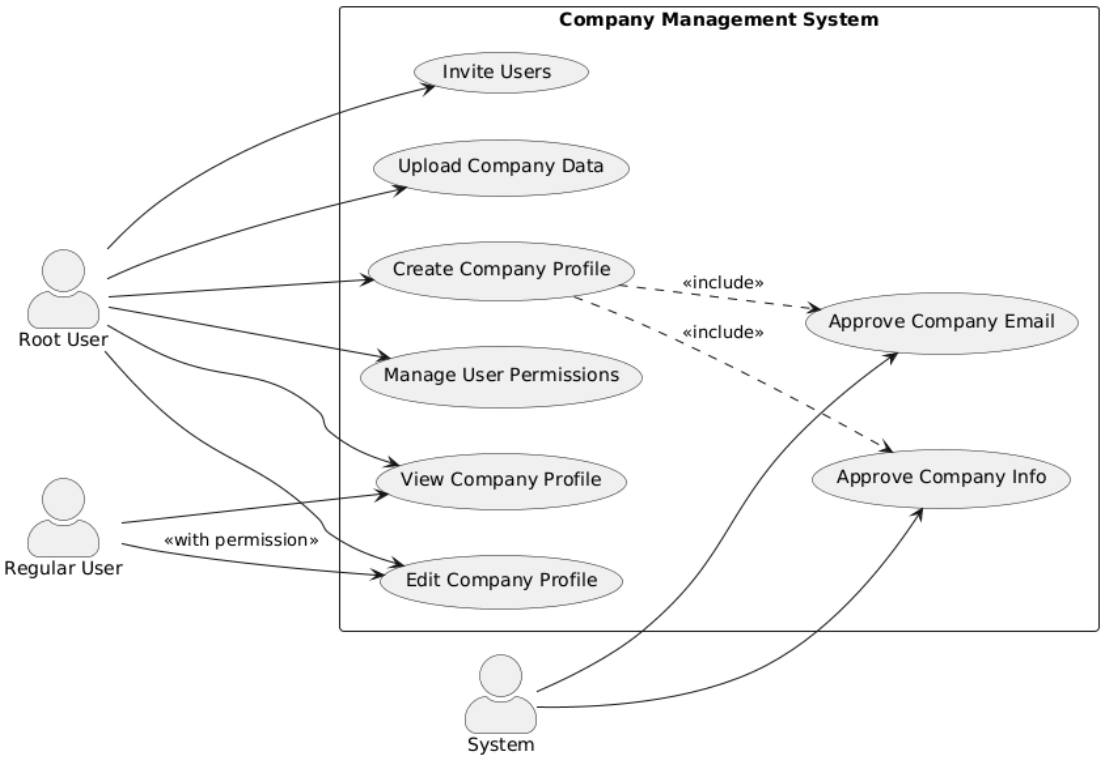


Fig.2 Company Management Use Case Diagram

#### 3.5.2.3 Project Management UML Use Case Diagram

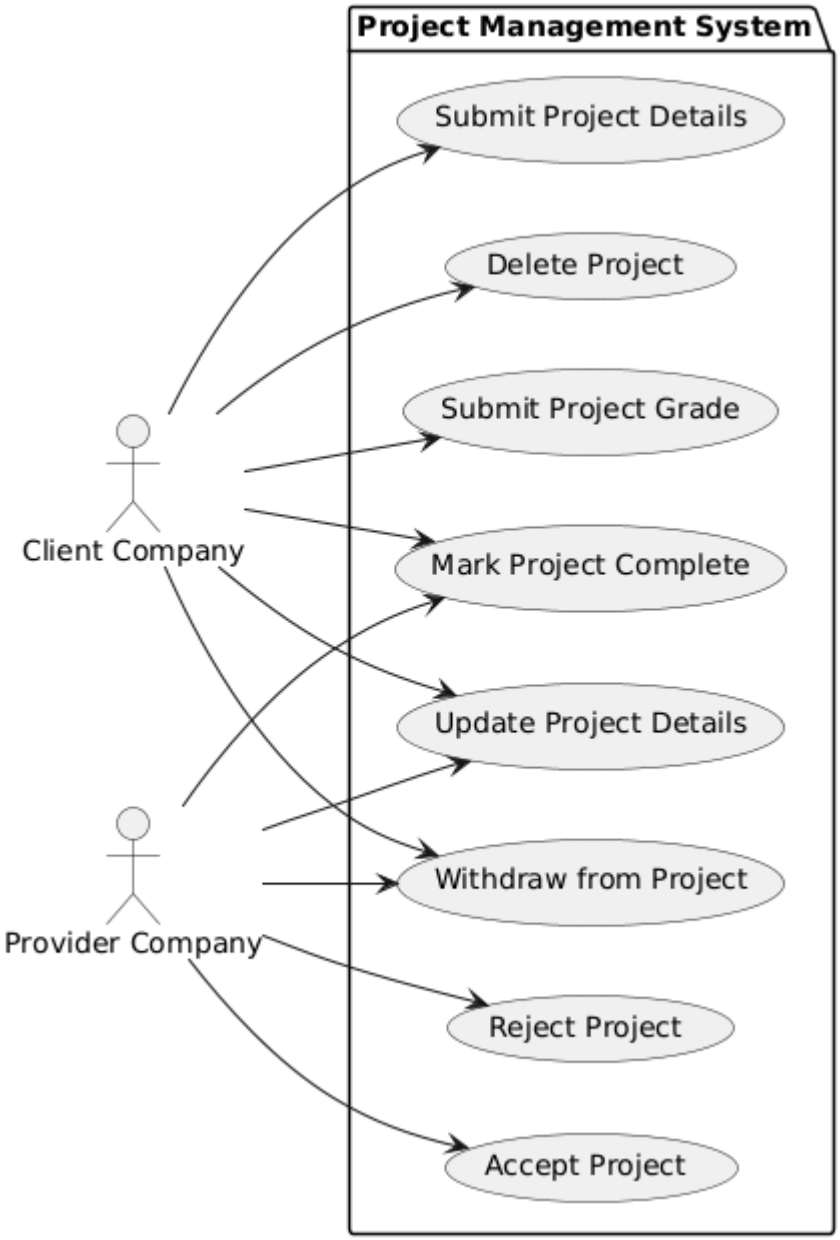


Fig. 3 Project Management Use Case Diagram

#### 3.5.2.4 System Functions UML Use Case Diagram

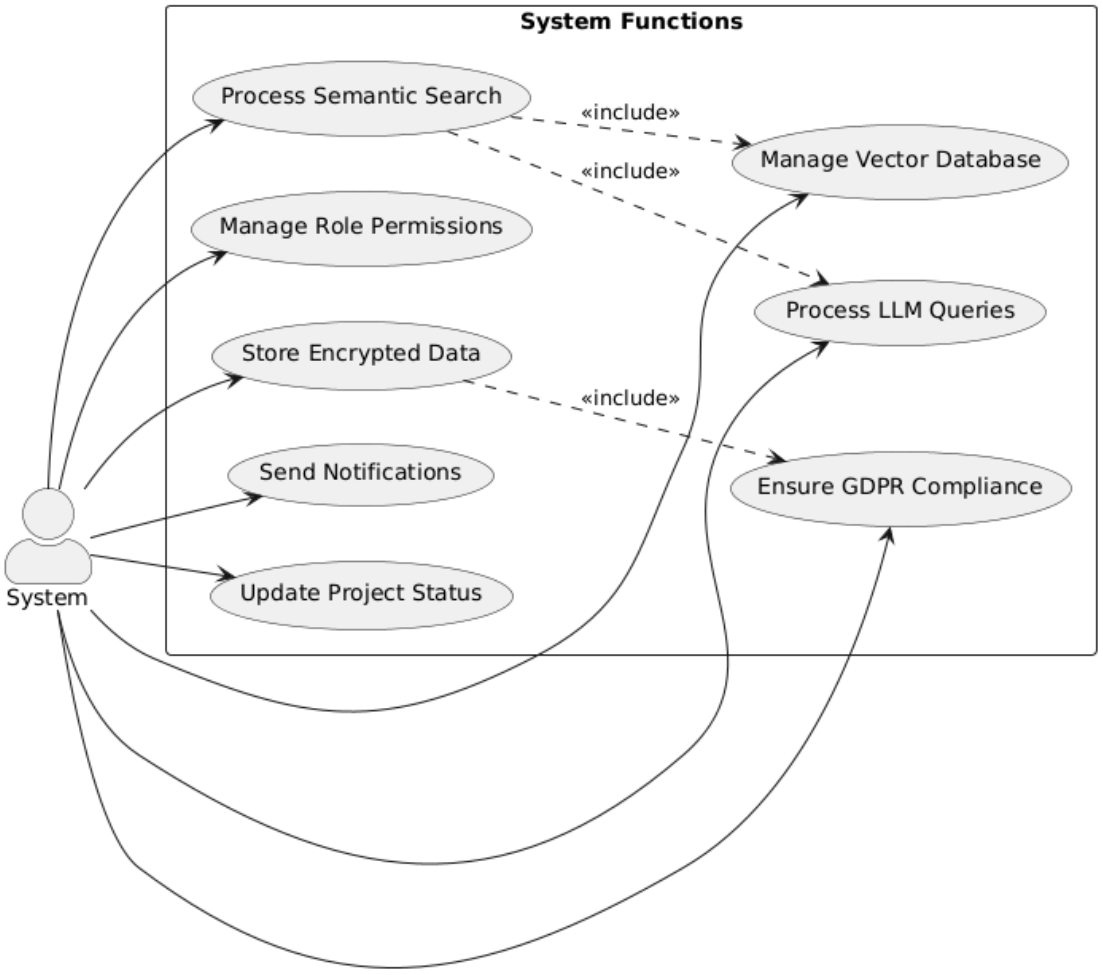


Fig. 4 System Functions Use Case Diagram

### 3.5.3 Object and Class Model

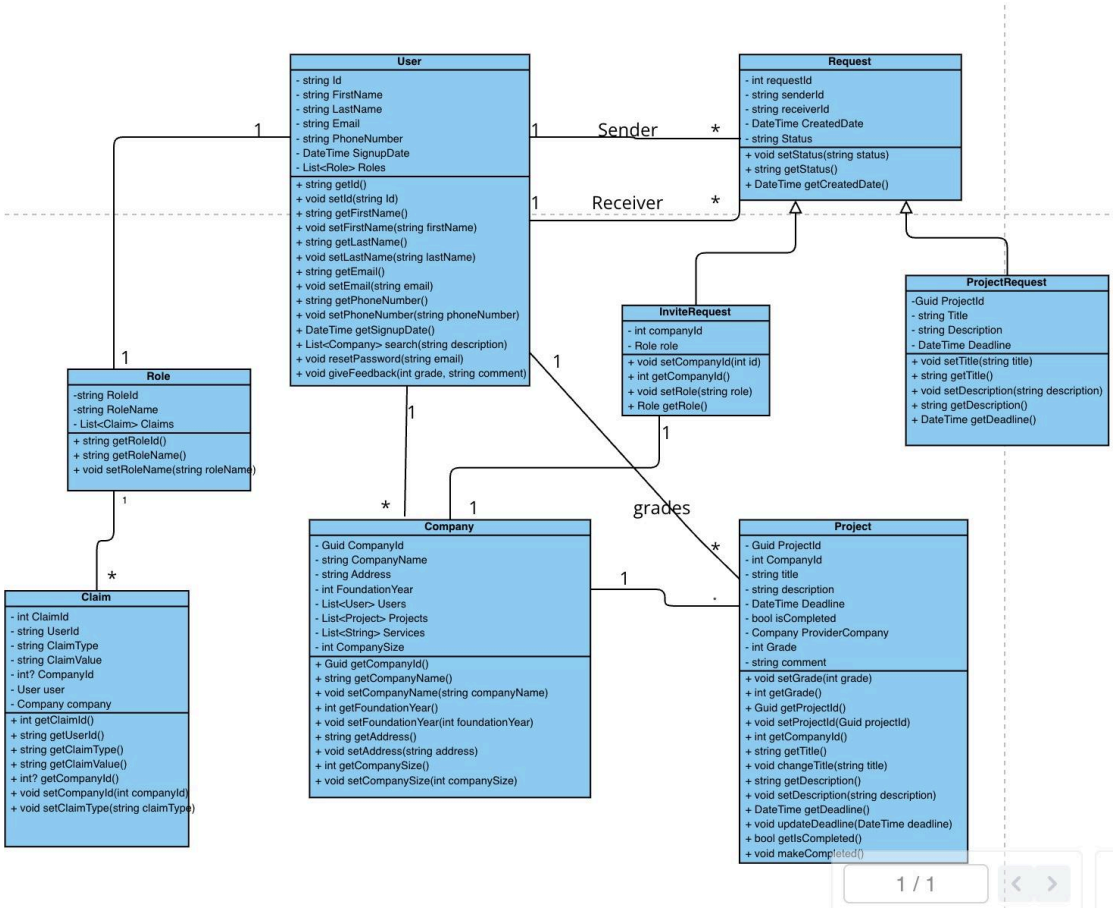


Fig. 5 UML Class Diagram

### 3.5.4 Dynamic Models

#### 3.5.4.1 Search Dynamic Models

##### 3.5.4.1.1 Search Sequence Diagram

The below sequence diagram shows what happens when the user enters a search query to do company search.

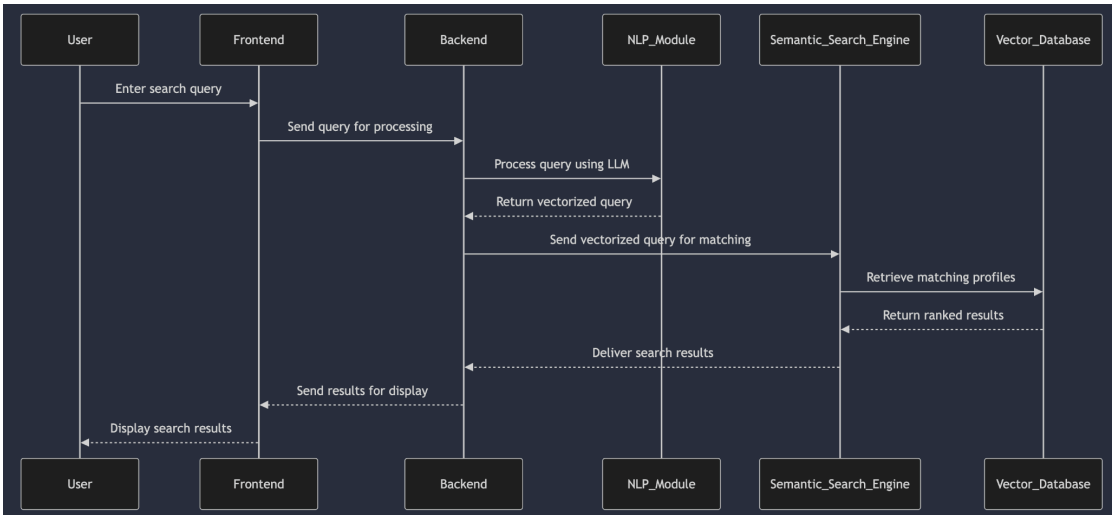


Fig. 6 Company Search Sequence Diagram

### 3.5.4.1.2 Search Activity Diagram

The below activity diagram depicts the company search process which starts with the user's search query.

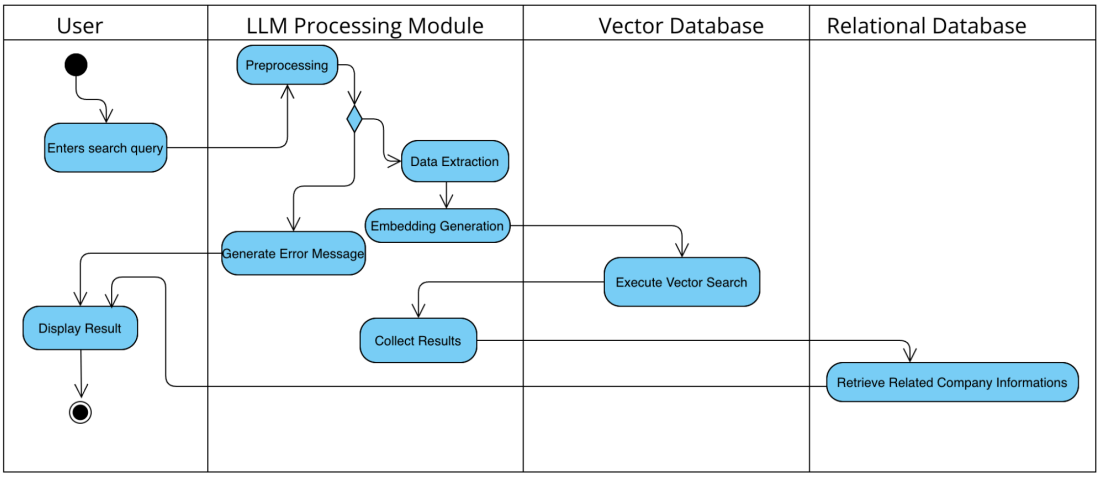


Fig. 7 Company Search Activity Diagram

## 3.5.4.2 Company Profile Dynamic Models

### 3.5.4.2.1 Company Profile Sequence Diagram

The below sequence diagram depicts the company data insertion process.

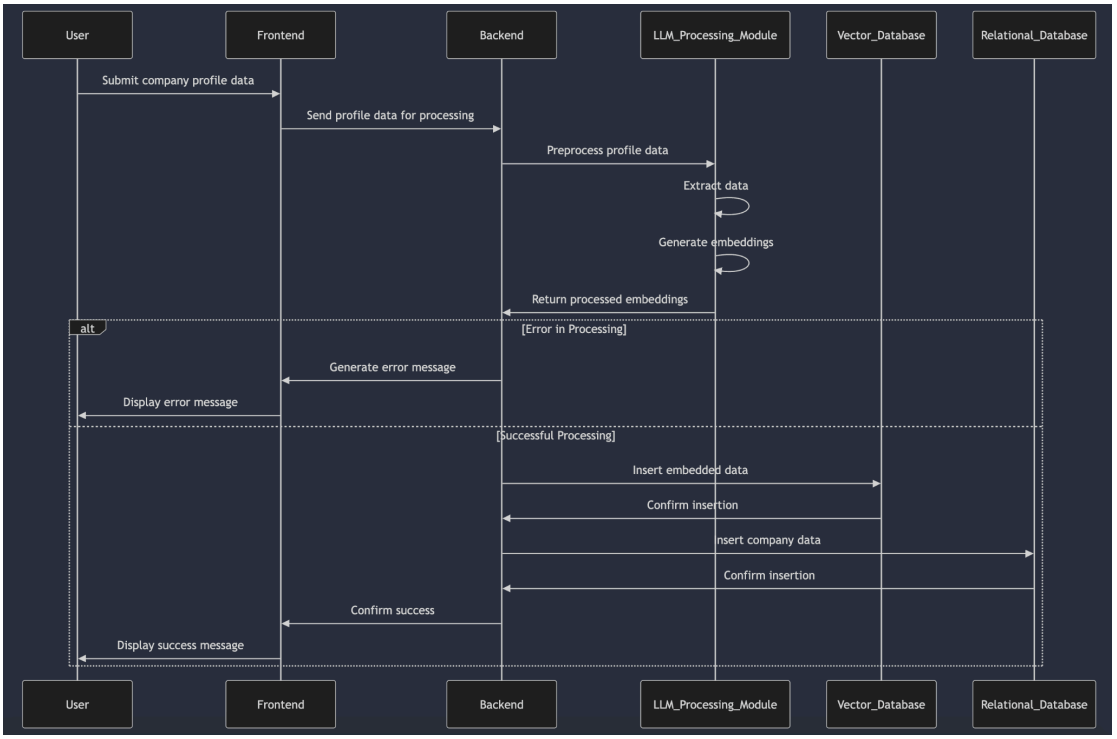


Fig. 8 Company Profile Sequence Diagram

### 3.5.4.2.2 Company Profile Activity Diagram

The below activity diagram depicts the company data insertion process.

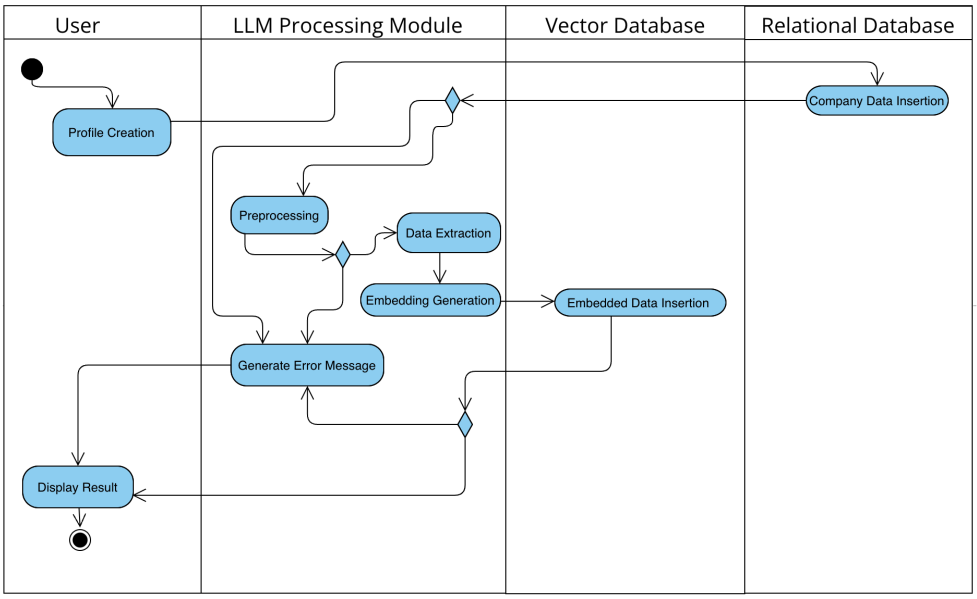


Fig. 9 Company Profile Activity Diagram

### 3.5.4.3 Project Management Dynamic Models

### 3.5.4.3.1 Project Management Sequence Diagram

The below sequence diagram shows the project management process starting from the project creation until the end of the project and grading.


Fig. 10 Project Management Sequence Diagram

### 3.5.4.3.2 Project Management Activity Diagram

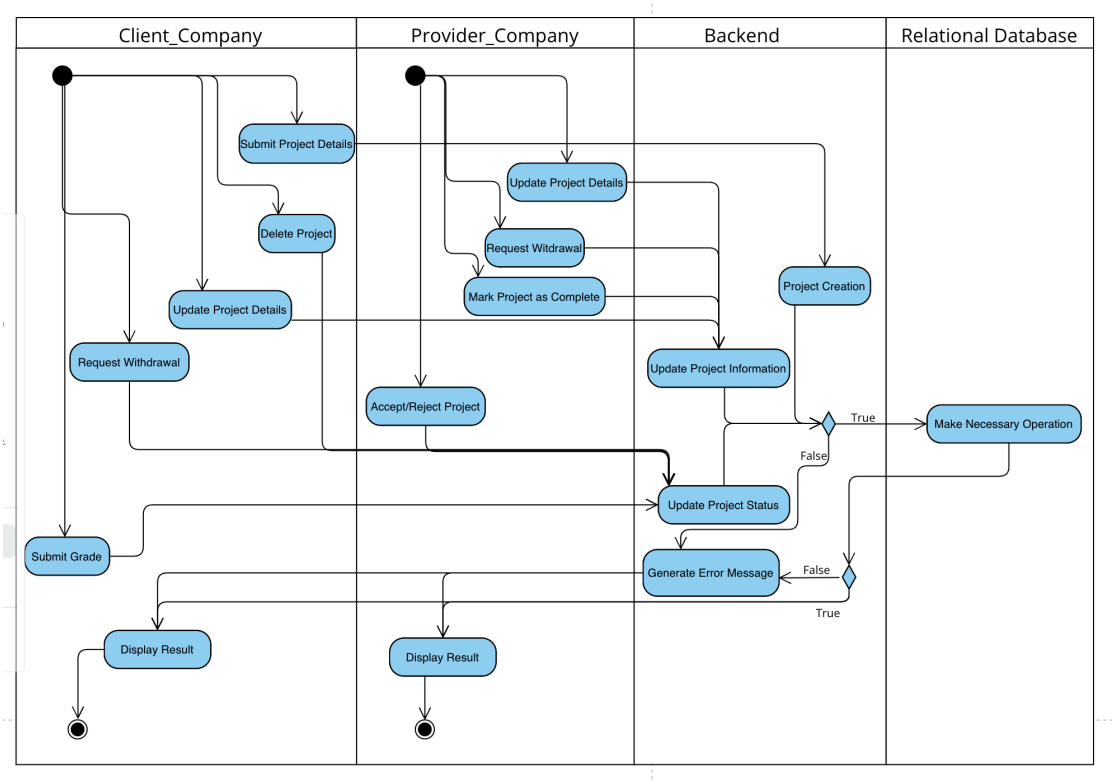


Fig. 11 Project Management Activity Diagram

### 3.5.5 User Interface

#### 3.5.5.1. Registered User

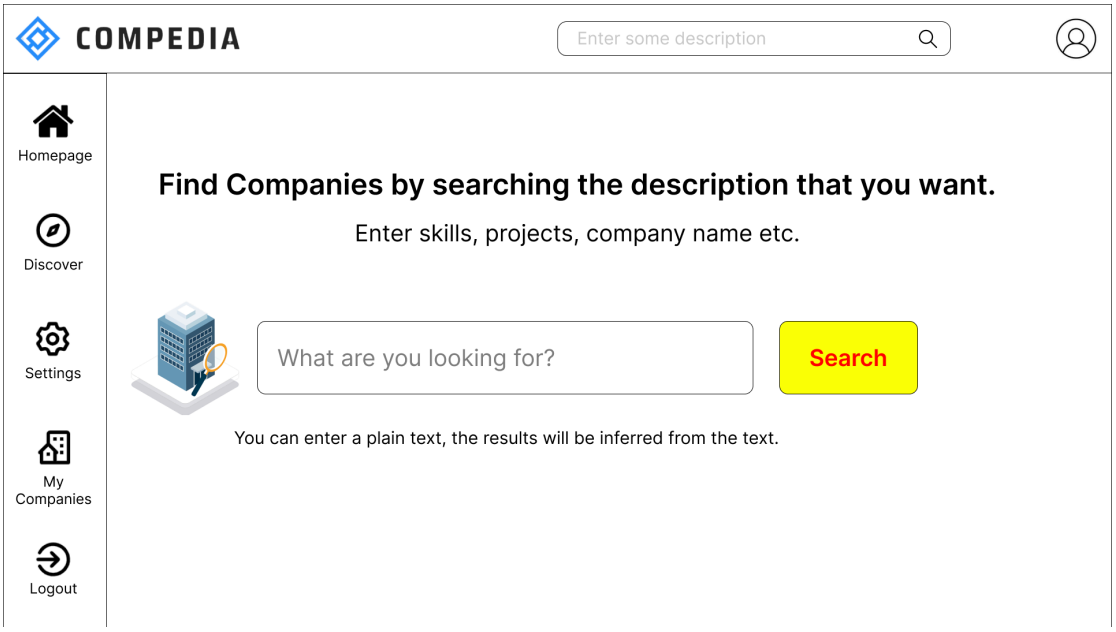


Fig. 12 Homepage

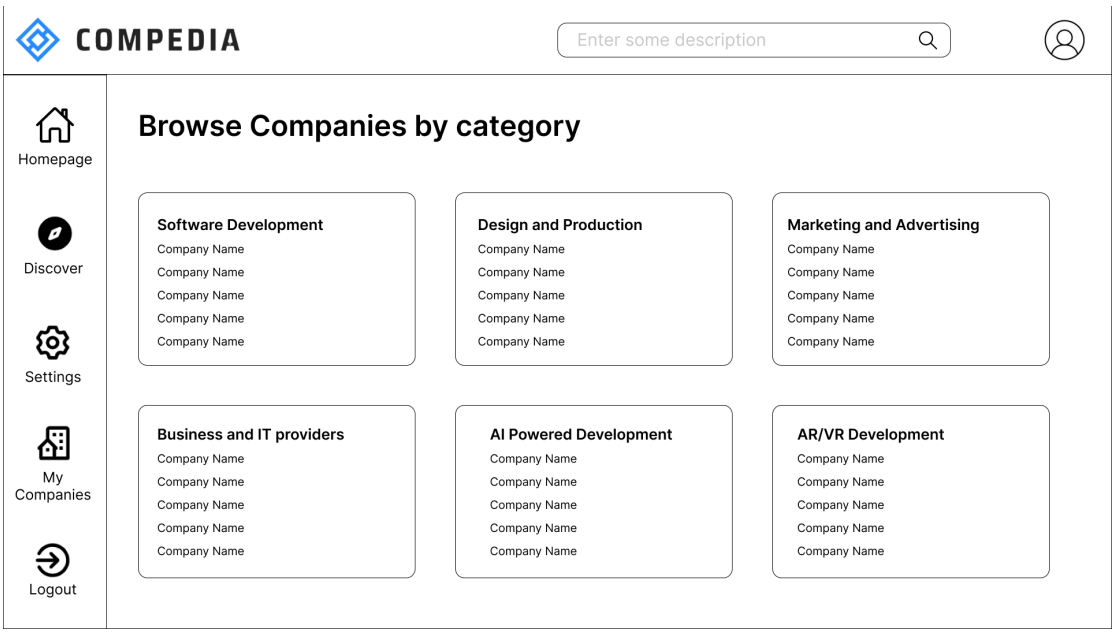


Fig. 13 Discover Page

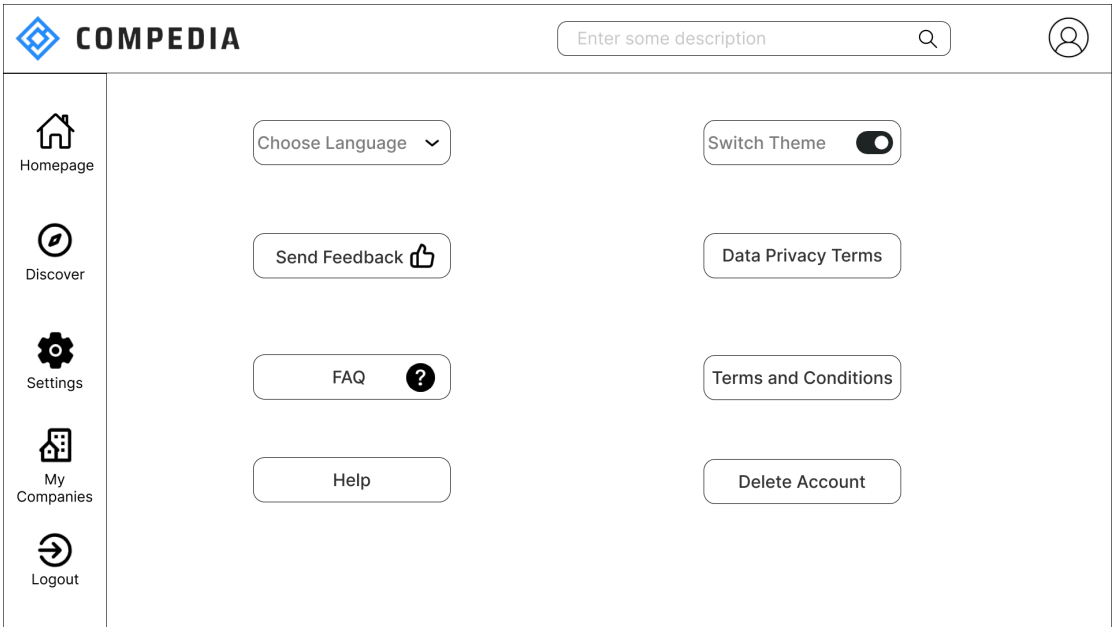


Fig. 14 Settings Page

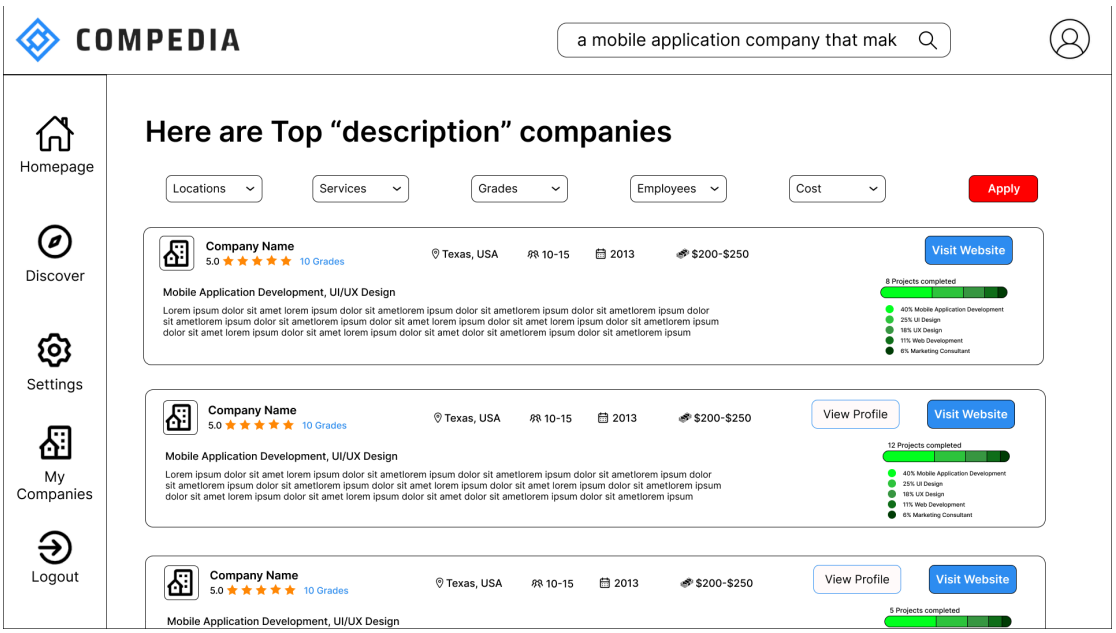


Fig. 15 Search Page

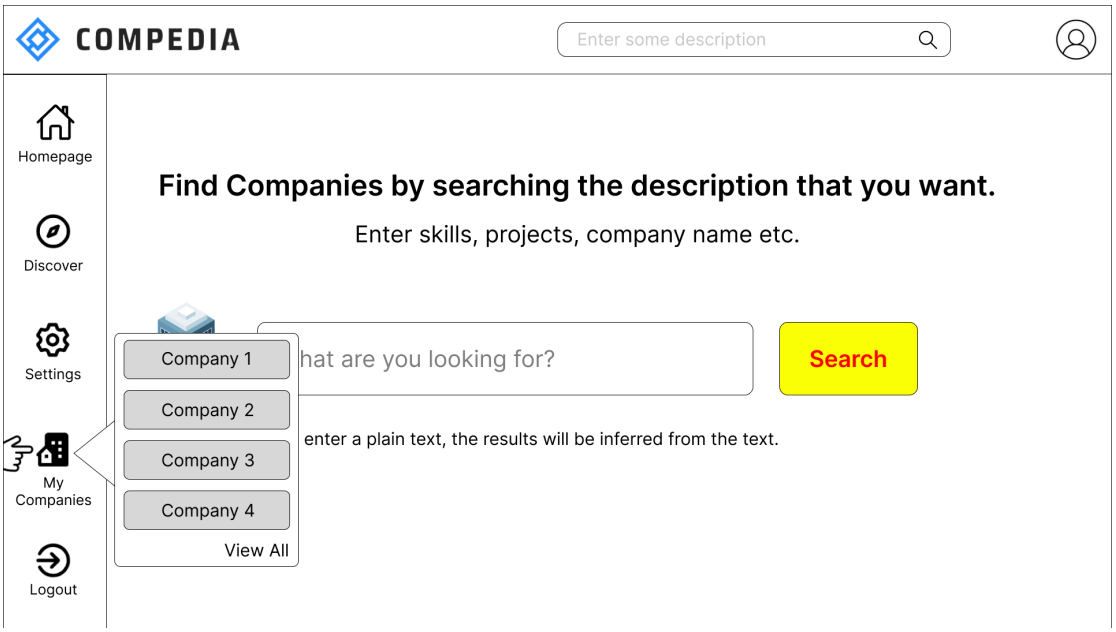


Fig. 16 My Companies

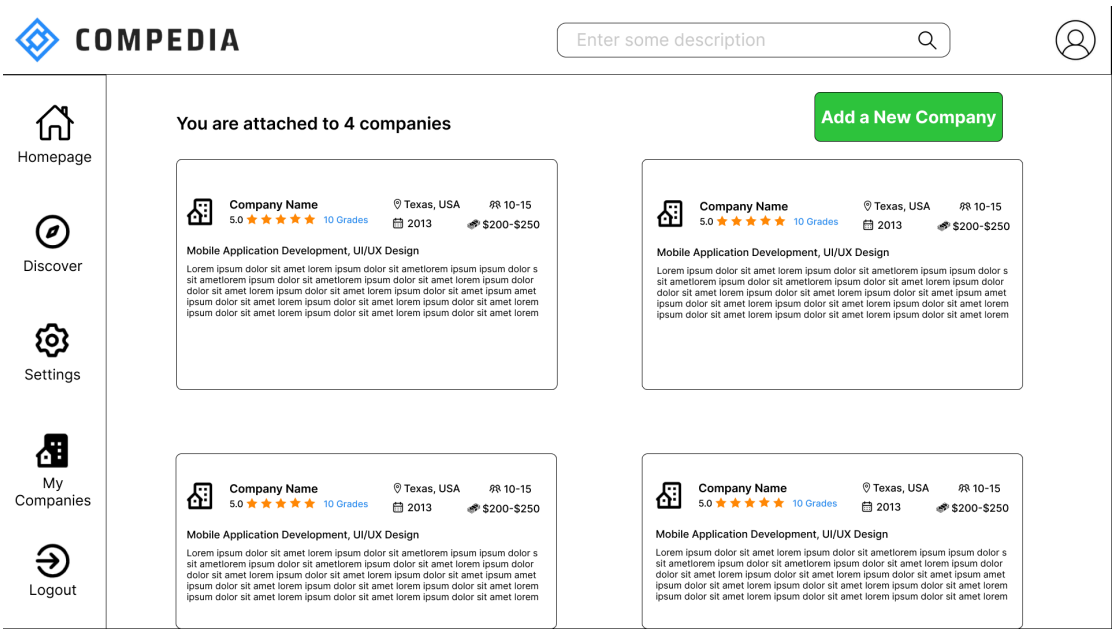


Fig. 17 My Companies Page

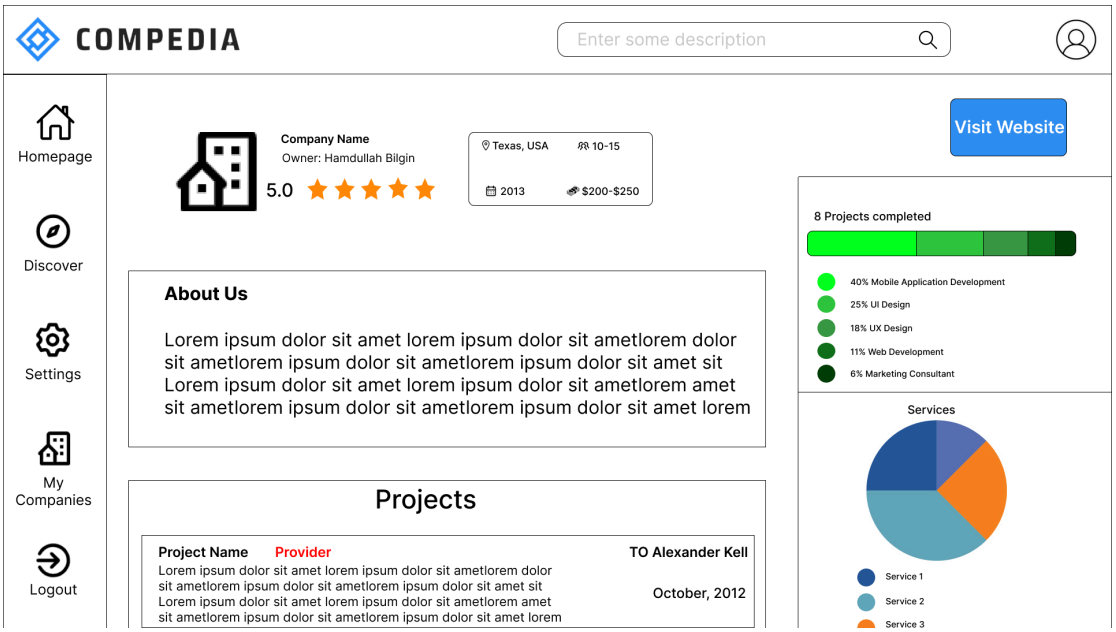


Fig. 18 Company Profile Page

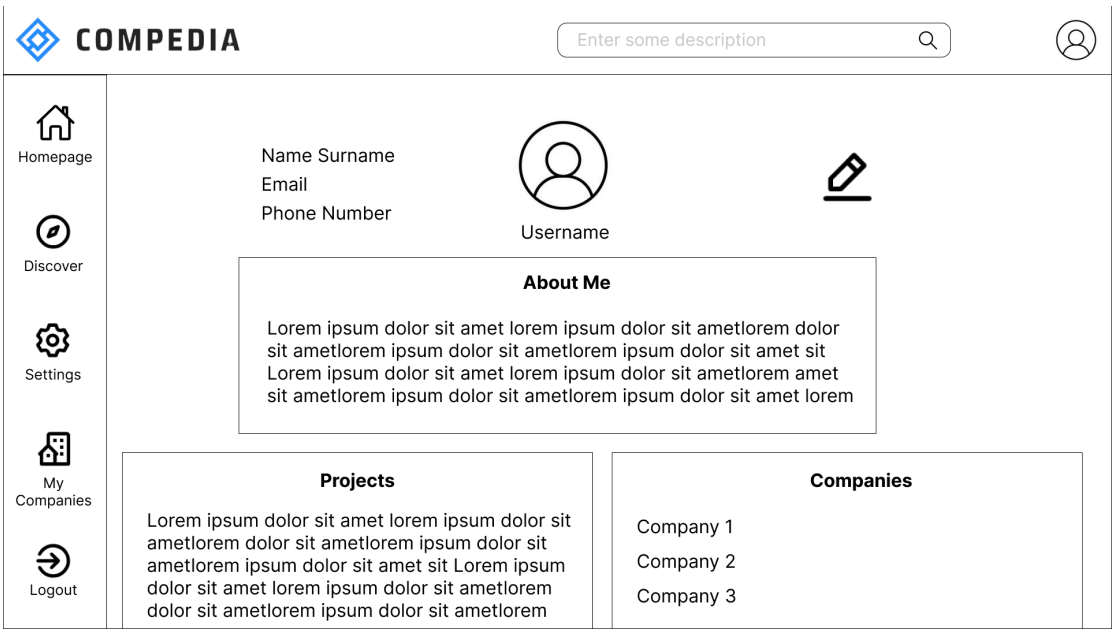


Fig. 19 User Profile Page

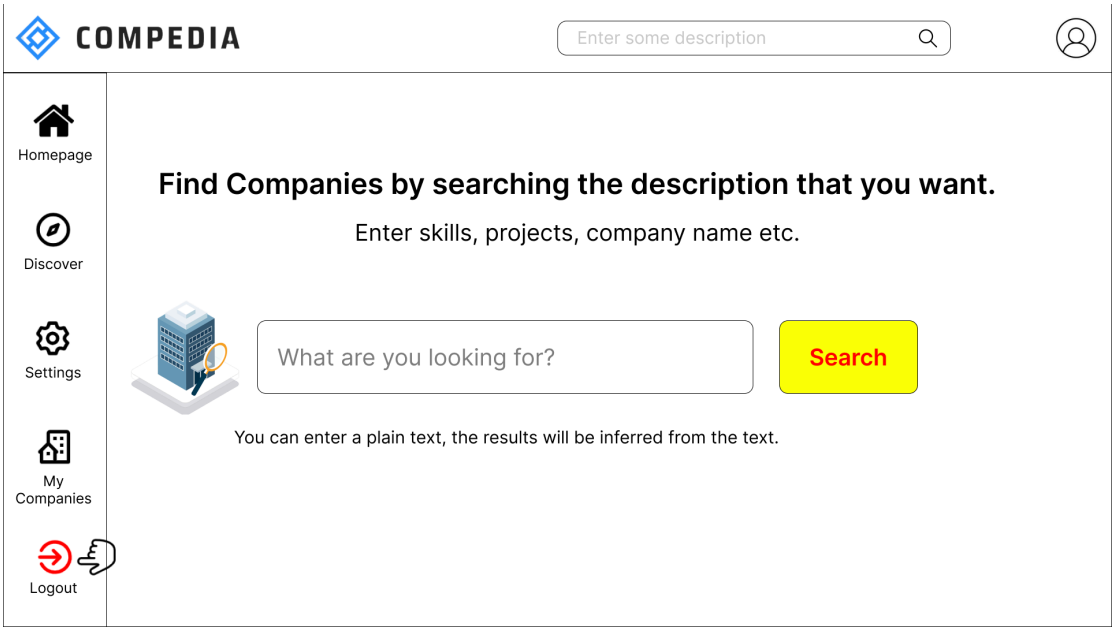


Fig. 20 Logout

### 3.5.5.2. Login/Sign Up

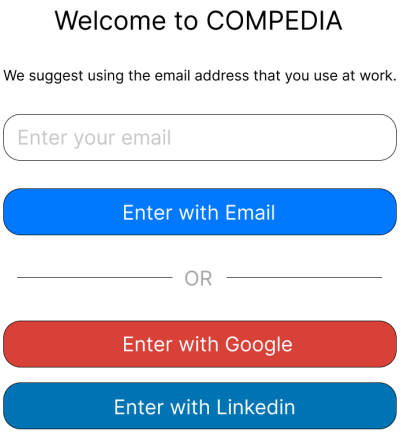


Fig. 21 Sign in Page

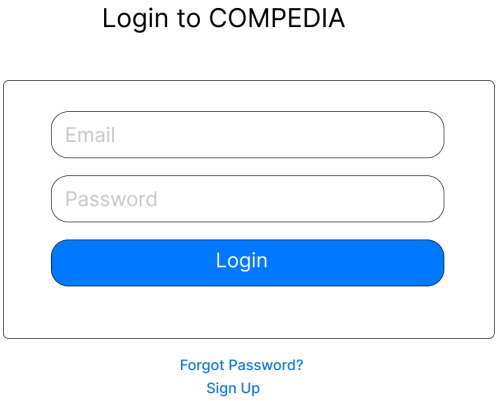


Fig. 22 Login Page

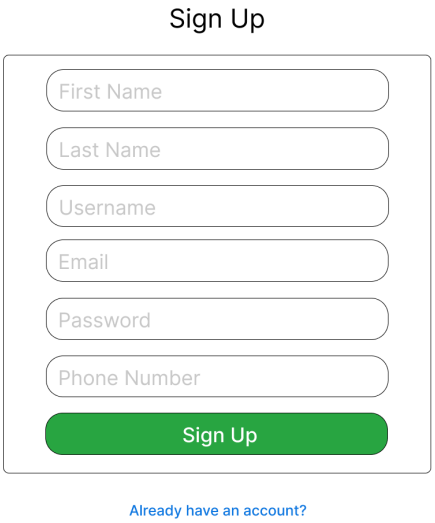


Fig. 23 Sign Up Page

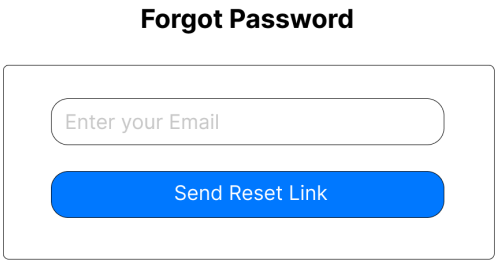


Fig. 24 Forgot Password Page

### 3.5.5.3. Unregistered User

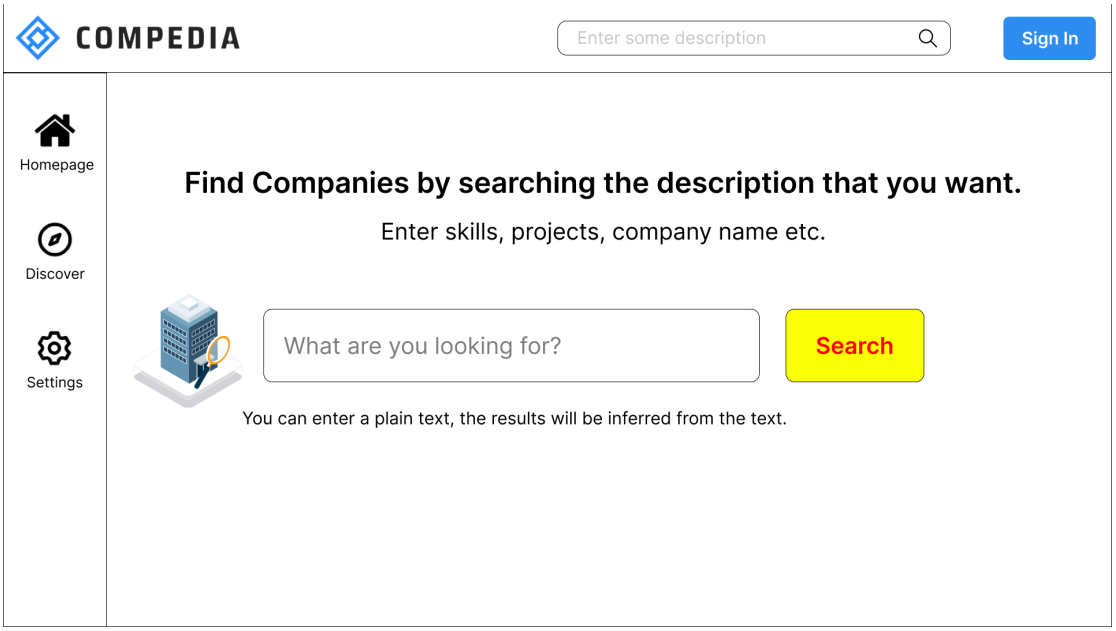


Fig. 25 Home Page

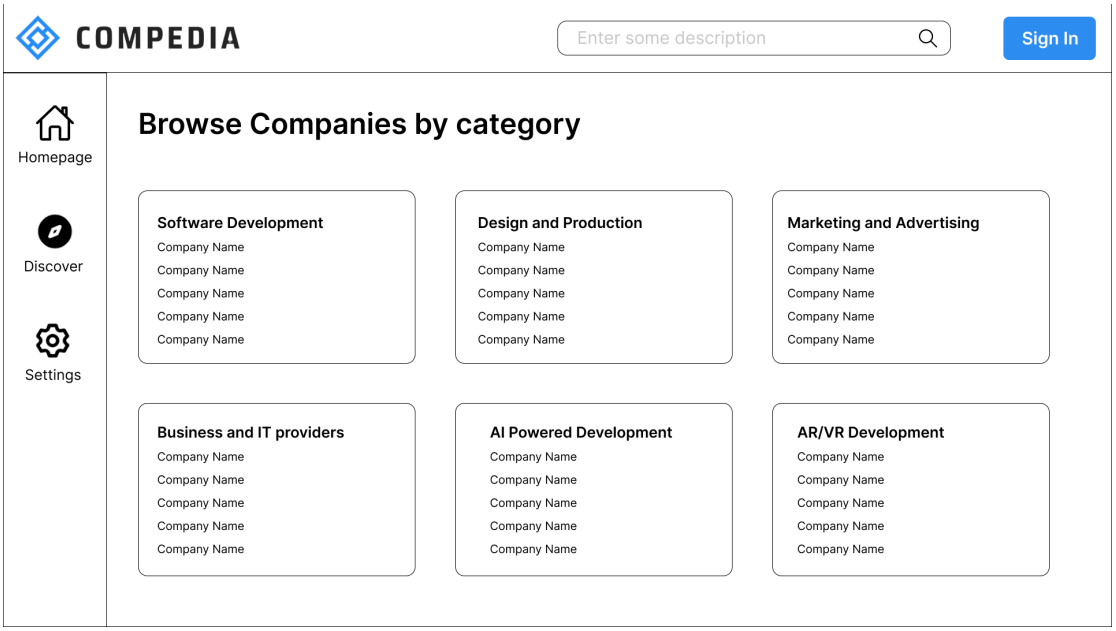


Fig. 26 Discover Page

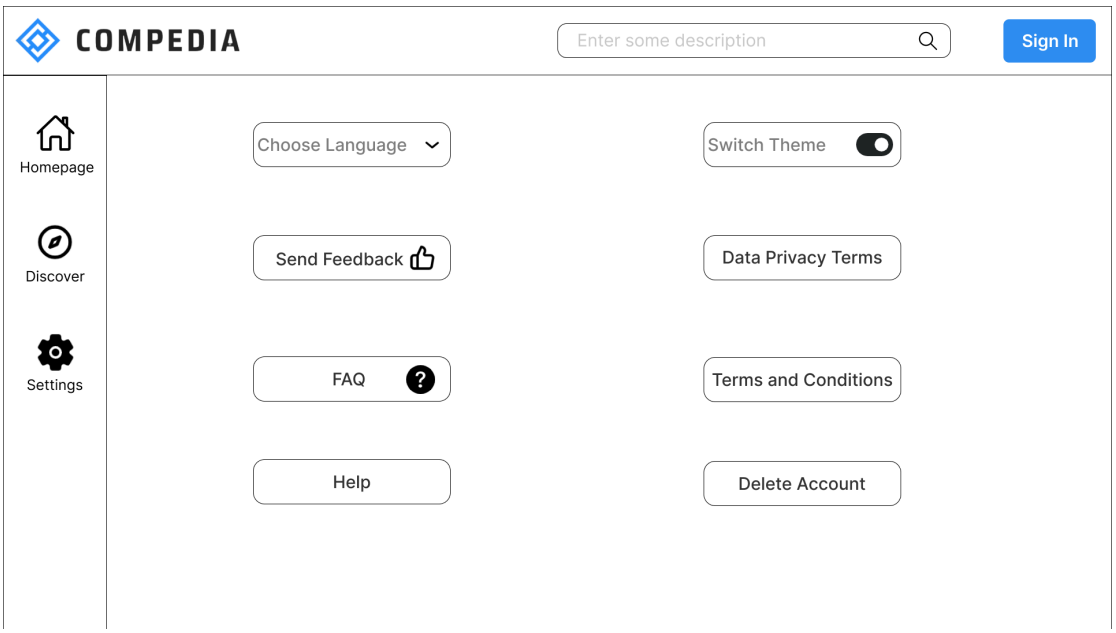


Fig. 27 Settings Page

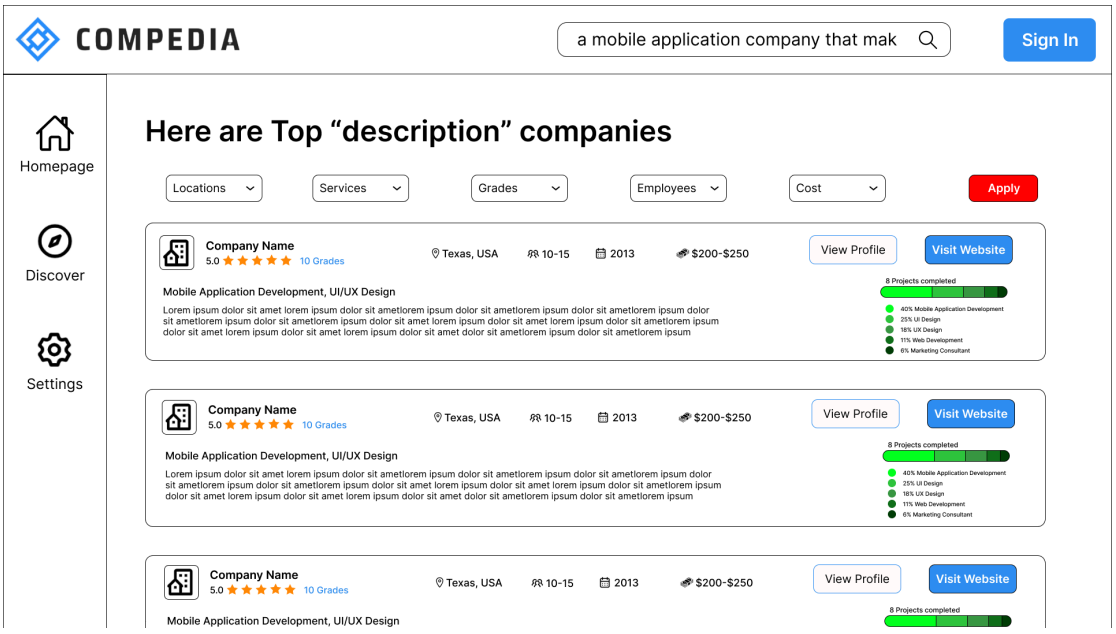


Fig. 28 Search Page

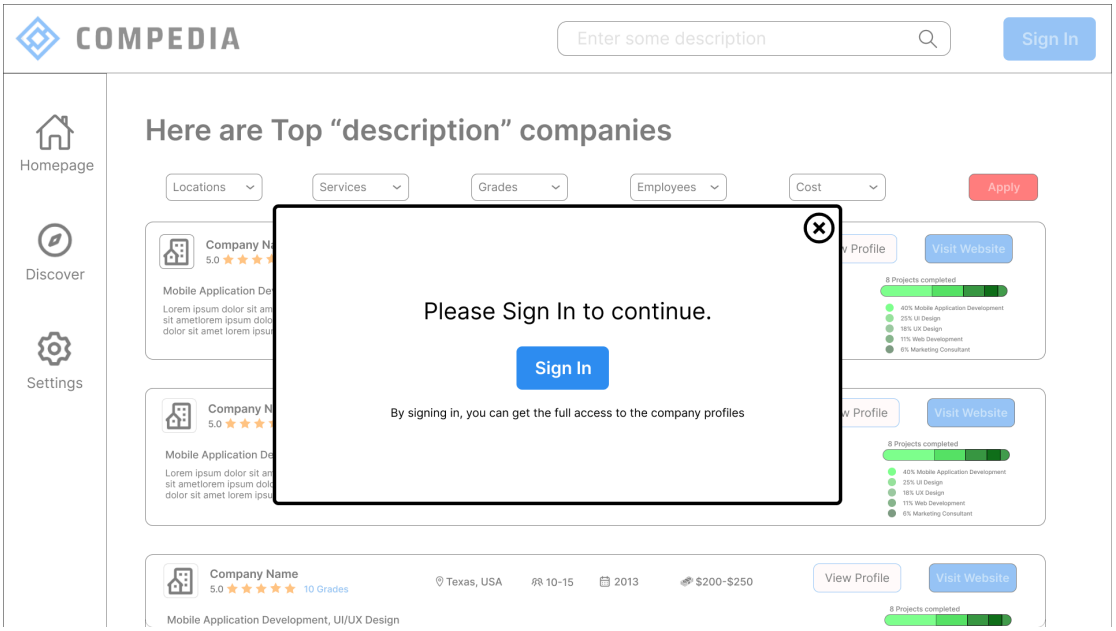


Fig. 29 Sign in Pop up

# 4 Other Analysis Elements

## 4.1 Consideration of Various Factors in Engineering Design

### 4.1.1 Constraints

The technical, practical, and external constraints affect the development and implementation of Compedia. The project will be implemented to meet the design goals with the accordance of constraints.

**Accessibility**

- The platform aims to be globally accessible to users by providing responsive and user-friendly web design.
- It is accessible on all major browsers (Chrome, Firefox, Safari, Edge) and devices with stable internet connectivity.

**Aesthetics**

- Simplicity and professionalism are prioritized by implementing the user interface and the platform has a business-oriented nature.
- An intuitive and clean design guarantees smooth navigation and reinforces trust in the platform’s credibility.

**Codes and Standards**

- Development adheres to REST API Design Guidelines and UML 2.5.1 for system modeling.
- Security measures are implemented in accordance with GDPR and local data privacy regulations.
- Peer-reviewed commits and code conventions ensure maintainability and scalability.

**Cost**

- The development costs are reduced thanks to open-source technologies (e.g., React, FastAPI).
- Azure cloud services optimize hosting and computational costs.
- Initial funding covers the custom domain, while maintenance costs include database and LLM processing fees.

**Ergonomics and Usability**

- Navigation is streamlined with search features always accessible via the topbar and homepage.
- WCAG (Web Content Accessibility Guidelines) is considered when choosing the UI components to ensure the accessibility.

**Interoperability**

- Milvus and Microsoft SQL Server are integrated with the backend for vector searches and relational data storage.
- To interact with the external AI services such as GPT and BERT, APIs are designed accordingly.
- Users can import standard data formats (e.g., CSV, JSON) for company profile creation.

**Legal Considerations**

- User and company data is stored and processed according to GDPR, data privacy laws.
- Policies ensure transparency about how data is used.

**Sustainability**

- By optimizing LLM usage and relying on cloud infrastructure, energy-efficient processing is achieved and redundant computations are minimized.
- Efficient handling of large-scale vector data is ensured thanks to Milvus which also reduces the storage usage.

**Schedule**

- A strict timeline is followed and milestones are set for development phases, including MVP delivery and public beta testing.
- Agile methodologies (Scrum) are used to adapt schedules dynamically based on progress and feedback.

**Minimum Hardware and Software Specifications**

- A modern web browser and internet access are required.

- Backend infrastructure operates on Azure with containerized services (via Docker) for scalability and efficiency.

Table 1: Summary of Factors Affecting Compedia

| Factor | Impact on Design and Development | Effect Level |
|---|---|---|
| Public Health | Secure data handling is ensured to avoid misuse of sensitive information that could indirectly affect public health. | Medium |
| Public Safety | Robust encryption and access control are implemented to prevent unauthorized access to sensitive company data. | High |
| Public Welfare | Business matchmaking efficiency is enhanced, promoting economic opportunities for small and large companies alike. | High |
| Global Factors | International data privacy laws (e.g., GDPR) are followed and a multilingual user base for global accessibility is supported. | High |
| Cultural Factors | Avoids bias in recommendations by ensuring neutrality and respecting diverse business and cultural contexts. | Medium |
| Social Factors | Verified profiles, ethical AI use, and fostering transparency in business interactions build trust. | High |
| Environmental Factors | By using energy-efficient cloud hosting and optimized AI model deployments, environmental impact is reduced. | Low |
| Economic Factors | Open-source tools and scalable architecture are employed which ensure affordability and sustainability. | High |

### 4.1.2 Standards

We use Git as a version control system. We use Trello to track the assignments that are given to each group member. Moreover, in every two weeks, we arrange meetings in order to discuss recent problems, provide solutions to problems and make assignments to group members. Our project adheres to the following standards to ensure clarity, consistency, and quality. UML 2.5.1 will be used for system modeling to represent use cases, workflows, and design architecture effectively. REST API Design Guidelines will ensure the interoperability and scalability in API development. These standards enhance the project's development and usability, ensuring adherence to industry practices.

## 4.2 Risks and Alternatives

Table 2: Risks

| Risks | Likelihood | Effect on the Project | B plan summary |
|---|---|---|---|
| High competition in the market | High | The platform might struggle to attract users due to the presence of similar solutions and established players in the market. | Focus on niche industries, provide highly tailored solutions, and leverage advanced features like semantic search for differentiation. |
| Lack of trust in data credibility | Medium | Users may be reluctant to rely on the platform due to concerns about fake reviews or biased company data. | Implement a robust verification system, including cross-checking data sources and enabling user feedback on data accuracy. |
| Difficulty in onboarding smaller companies | High | Smaller companies with limited budgets may not adopt the platform due to concerns about affordability or visibility. | Offer a freemium model to attract small businesses and provide cost-effective promotion packages tailored for their needs. |
| User reliance on existing search engines | Medium | Potential users may prefer traditional search engines despite their limitations, reducing platform adoption. | Emphasize the platform’s unique value proposition through marketing campaigns and educate users about the benefits of semantic search. |
| Data privacy and security concerns | Low | Businesses may hesitate to provide detailed information due to concerns about data misuse or breaches. | Implement strict data security measures, comply with GDPR and similar regulations, and clearly communicate these measures to users. |

### 4.3 Project Plan

Table 3: List of work packages

| WP# | Work package title | Leader | Members involved |
|---|---|---|---|
| WP1 | Azure Database setup and Core Database Design | Yaşar Tatlıcıoğlu | Ece Beyhan |
| WP2 | Market research on successful semantic search examples | Bartu Albayrak | - |
| WP3 | Market research competitive analysis | Yaşar Tatlıcıoğlu | Serhat Yılmaz, Anıl Altuncu |
| WP4 | Github setup and configuration | Ece Beyhan | - |
| WP5 | Trello setup | Yaşar Tatlıcıoğlu | - |
| WP6 | Frontend Core design and implementation | Serhat Yılmaz | - |
| WP7 | Backend Core design and implementation | Bartu Albayrak | Yaşar Tatlıcıoğlu |
| WP8 | Company profile information structure research | Anıl Altuncu | Yaşar Tatlıcıoğlu |
| WP9 | Creating methodology for free-text processing | Anıl Altuncu | - |
| WP10 | Company database research | Ece Beyhan | - |
| WP11 | Conducting survey with potential customers | Anıl Altuncu | Yaşar Tatlıcıoğlu, Serhat Yılmaz, Ece Beyhan, Bartu Albayrak |
| WP12 | Backend authentication-authorization implementation | Bartu Albayrak | Yaşar Tatlıcıoğlu |
| WP13 | Frontend-Backend Integration | Serhat Yılmaz | Bartu Albayrak |
| WP14 | Location API research | Ece Beyhan | - |
| WP15 | LLM processing backend implementation | Anıl Altuncu | Yaşar Tatlıcıoğlu |
| WP16 | Vector Database Setup | Yaşar Tatlıcıoğlu | - |
| WP17 | Company profile responsive frontend design | Serhat Yılmaz | - |
| WP18 | LLM processing - Vector Database integration | Anıl Altuncu | Yaşar Tatlıcıoğlu |
| WP19 | Microservice Architecture Implementation | Ece Beyhan | Yaşar Tatlıcıoğlu |
| WP20 | Frontend app and backend services cloud deployment | Yaşar Tatlıcıoğlu | Ece Beyhan |

| **WP 1:** Azure Database setup and Core Database Design | | | |
|---|---|---|---|
| **Start date:** 01.10.2024 **End date:** 07.10.2024 | | | |
| **Leader:** | *Yaşar Tatlıcıoğlu* | **Members involved:** | *Ece Beyhan* |
| **Objectives:** *Configure an Azure managed Microsoft SQL Database to support the application's relational data needs* | | | |
| **Tasks:** | | | |

| |
|---|
| ***Task 1.1 Cloud Deployment*****:** *Provision Azure SQL instance.*<br>***Task 1.2 Cloud Deployment:*** *Set up schemas and tables for the database.*<br>***Task 1.3 Obtain Connection Information:*** *Connection data such as string id, root username and password.* |
| **Deliverables**<br>***D1.1:*** *Fully operational Azure database with secure access.* |

| **WP 2:** Market research on successful semantic search examples | | | |
|---|---|---|---|
| **Start date:** 01.10.2024 **End date:** 15.10.2024 | | | |
| **Leader:** | *Bartu Albayrak* | **Members involved:** | - |
| **Objectives:** *Identify best practices in semantic search systems by analyzing successful implementations in industries like e-commerce, recruitment, and knowledge management. The focus is on understanding the technology, features, and challenges of these systems.* | | | |
| **Tasks:**<br>***Task 2.1 Market Analysis*** **:** *Analyze top semantic search solutions*<br>***Task 2.2 Identification of common facts*** **:** *Document key features, benefits, and challenges.* | | | |
| **Deliverables**<br>***D2.1:*** *Report summarizing examples, insights, and takeaways.* | | | |

| **WP 3:** Market research competitive analysis | | | |
|---|---|---|---|
| **Start date:** 01.10.2024 **End date:** 01.11.2024 | | | |
| **Leader:** | *Yaşar Tatlıcıoğlu* | **Members involved:** | *Serhat Yılmaz, Anıl Altuncu* |
| **Objectives:** *Conduct a competitive analysis to understand the offerings, strengths, and weaknesses of similar platforms in the market. This will help us identify unique value propositions and areas for differentiation.* | | | |
| **Tasks:**<br>***Task 2.1 Finding competitors*** **:** *Identify competitors.*<br>***Task 2.2 Competitor analysis*****:** *Analyze features, pricing, and target markets.*<br>***Task 2.3 Market Summarization****: Compare strengths and weaknesses.* | | | |
| **Deliverables**<br>***D2.1:*** *Competitive analysis document.* | | | |

| **WP 4:** Github setup and configuration | | | |
|---|---|---|---|
| **Start date:** 01.10.2024 **End date:** 02.10.2024 | | | |
| **Leader:** | *Ece Beyhan* | **Members involved:** | - |
| **Objectives:** *Create and configure collaboration tools for development* | | | |
| **Tasks:**<br>***Task 2.1 Github*****:** *Set up Github organization and repositories.* | | | |
| **Deliverables**<br>***D2.1:*** Operational GitHub organization and repositories | | | |

| **WP 5:** Trello setup | | | |
|---|---|---|---|
| **Start date:** 01.10.2024 **End date:** 02.10.2024 | | | |
| **Leader:** | *Yaşar Tatlıcıoğlu* | **Members involved:** | - |

| |
|---|
| **Objectives:** *Create and configure project management tools* |
| **Tasks:**<br>***Task 2.1 Trello*** **:** *Create Trello board for task tracking* |
| **Deliverables**<br>***D2.1:*** *Operational Trello board with initial tasks* |

| | | | |
|---|---|---|---|
| **WP 6:** Frontend core design and implementation | | | |
| **Start date:** 01.11.2024 **End date:** 16.12.2024 | | | |
| **Leader:** | *Serhat Yılmaz* | **Members involved:** | - |
| **Objectives:** *Build the core frontend structure, focusing on usability and responsive design. This includes creating reusable components and implementing a visually appealing interface.* | | | |
| **Tasks:**<br>***Task 2.1 Mockups*** **:** *Design mockups for UI/UX*<br>***Task 2.2 Skeletons in React.js*** **:** *Develop core components in React.js* | | | |
| **Deliverables**<br>***D2.1:*** *Functional frontend with core components implemented.* | | | |

| | | | |
|---|---|---|---|
| **WP 7:** Backend core design and implementation | | | |
| **Start date:** 01.11.2024 **End date:** 01.01.2025 | | | |
| **Leader:** | *Bartu Albayrak* | **Members involved:** | *Yaşar Tatlıcıoğlu* |
| **Objectives:** *Develop a scalable and secure backend to support core application functionalities. This includes designing APIs and implementing services for data handling and business logic.* | | | |
| **Tasks:**<br>***Task 2.1 API design*****:** *Design RESTful APIs.*<br>***Task 2.2 MVC Implementation*****:** *Implement core services using .NET Core* | | | |
| **Deliverables**<br>***D2.1:*** *API documentation and functioning backend.* | | | |

| | | | |
|---|---|---|---|
| **WP 8:** Company profile information structure research | | | |
| **Start date:** 11.11.2024 **End date:** 02.12.2024 | | | |
| **Leader:** | *Anıl Altuncu* | **Members involved:** | *Yaşar Tatlıcıoğlu* |
| **Objectives:** *Research and define the optimal structure for storing company profiles, including key fields like services, products, and location. The structure should enable efficient search and retrieval for various use cases.* | | | |
| **Tasks:**<br>***Task 2.1 Market research*****:** *Research common data fields for company profiles.*<br>***Task 2.2 Designing a schema*****:** *Define a schema for profile data storage.* | | | |
| **Deliverables**<br>***D2.1:*** *Company profile schema documentation.* | | | |

| | | | |
|---|---|---|---|
| **WP 9:** Creating methodology for free-text processing | | | |
| **Start date:** 02.12.2024 **End date:** 01.01.2025 | | | |
| **Leader:** | *Anıl Altuncu* | **Members involved:** | - |

| |
|---|
| **Objectives:** *Develop a methodology for processing unstructured text into structured data, focusing on semantic understanding. This includes leveraging NLP techniques and LLMs for parsing and normalization.* |
| **Tasks:**<br>***Task 2.1 Research to determine the state-of-the-art*:** *Research techniques for text parsing and semantic understanding.*<br>***Task 2.2 Implementation Plan* :** *Design a preprocessing pipeline.* |
| **Deliverables**<br>***D2.1:** Documentation on free-text processing methodology.* |

| **WP 10:** Company database research | | | |
|---|---|---|---|
| **Start date:** 01.11.2024 **End date:** 01.01.2025 | | | |
| **Leader:** | *Ece Beyhan* | **Members involved:** | - |
| **Objectives:** *Finding useful company datasets. Especially the datasets that include related data such as specialties, products will be collected.* | | | |
| **Tasks:**<br>***Task 2.1 Search for Dataset*:** *Find company datasets in Kaggle or other 3rd party options.*<br>***Task 2.2 Documentation* :** *Document the collected datasets with their schema* | | | |
| **Deliverables**<br>***D2.1:** Database tables with company datasets*<br>***D2.2**: Documentation for collected datasets.* | | | |

| **WP 11:** Conducting survey with potential customers | | | |
|---|---|---|---|
| **Start date:** 01.12.2024 **End date:** 22.12.2024 | | | |
| **Leader:** | *Anıl Altuncu* | **Members involved:** | Yaşar Tatlıcıoğlu, Serhat Yılmaz, Ece Beyhan, Bartu Albayrak |
| **Objectives:** *Gather insights from potential users to validate assumptions about the product's features and functionality. The survey will help identify user needs, preferences, and pain points. Results will shape the product development roadmap.* | | | |
| **Tasks:**<br>***Task 2.1 Composing a survey* :** *Design survey questions.*<br>***Task 2.2 Collect responses* :** *Distribute surveys and collect responses.* | | | |
| **Deliverables**<br>***D2.1:** Survey report with analysis and recommendations.* | | | |

| **WP 12:** Backend authentication-authorization implementation | | | |
|---|---|---|---|
| **Start date:** 11.11.2024 **End date:** 13.01.2025 | | | |
| **Leader:** | *Bartu Albayrak* | **Members involved:** | Yaşar Tatlıcıoğlu |
| **Objectives:** *Secure the application by implementing robust authentication and authorization mechanisms. This ensures that user access is restricted based on roles and permissions.* | | | |
| **Tasks:**<br>***Task 2.1 Configure .net core identity* :** *Configure .net core identity based authentication.*<br>***Task 2.2 User access implementation* :** *Implement claim-based access control* | | | |
| **Deliverables** | | | |

| |
|---|
| ***D2.1:*** *Secure authentication and authorization system.* |

| **WP 13:** Frontend-Backend Continuous Integration | | | |
|---|---|---|---|
| **Start date:** 16.12.2024 **End date:** 14.04.2025 | | | |
| **Leader:** | *Serhat Yılmaz* | **Members involved:** | Bartu Albayrak |
| **Objectives:** *Continuous frontend development that is backed up with API development to acquire optimal and functioning apps with efficient data transactions.* | | | |
| **Tasks:**<br>***Task 2.1 App Design Needs*** **:** *Determination of needed methods and their structures*<br>***Task 2.2 API development*** **:** *API development to respond the needs* | | | |
| **Deliverables**<br>***D2.1:*** *Continuous effective web application development* | | | |

| **WP 14:** Location API research | | | |
|---|---|---|---|
| **Start date:** 03.02.2025 **End date:** 17.02.2025 | | | |
| **Leader:** | *Ece Beyhan* | **Members involved:** | - |
| **Objectives:** *Identify a reliable API for location-based services to enable features like filtering by city or region. The API should support scalable and accurate location data processing.* | | | |
| **Tasks:**<br>***Task 2.1 Research for API*** **:** *Research available location APIs*<br>***Task 2.2 Integration to the system*** **:** *Test API integration with sample data.* | | | |
| **Deliverables**<br>***D2.1:*** *API recommendation and integration plan.* | | | |

| **WP 15:** LLM processing backend implementation | | | |
|---|---|---|---|
| **Start date:** 03.02.2025 **End date:** 21.04.2025 | | | |
| **Leader:** | *Anıl Altuncu* | **Members involved:** | *Yaşar Tatlıcıoğlu* |
| **Objectives:** *Build a backend service to process free-text inputs using large language models. This includes designing interfaces for model interactions and optimizing performance for frequent queries.* | | | |
| **Tasks:**<br>***Task 2.1 API design*** **:** *Design LLM API interface.*<br>***Task 2.2 Backend Implementation*** **:** *Implement request-handling logic.* | | | |
| **Deliverables**<br>***D2.1:*** *Functioning LLM processing backend.* | | | |

| **WP 16:** Vector Database Setup | | | |
|---|---|---|---|
| **Start date:** 01.01.2025 **End date:** 03.02.2025 | | | |
| **Leader:** | *Yaşar Tatlıcıoğlu* | **Members involved:** | - |
| **Objectives:** *Configure and integrate a vector database to support semantic search and recommendation features. The setup will ensure efficient storage and retrieval of embeddings.* | | | |
| **Tasks:**<br>***Task 2.1 Setting up Milvus*** **:** *Install and configure Milvus.* | | | |

| |
|---|
| ***Task 2.2 Integration with the system*****:** *Integrate database with LLM processing backend.* |
| **Deliverables**<br>***D2.1:*** *Operational vector database.* |

| | | | |
|---|---|---|---|
| **WP 17:** Company profile responsive interface design | | | |
| **Start date:** 03.02.2025 **End date:** 03.03.2025 | | | |
| **Leader:** | *Serhat Yılmaz* | **Members involved:** | - |
| **Objectives:** *Design a responsive and intuitive interface for managing and displaying company profiles. The interface will ensure accessibility and usability across devices. This design will align with the overall application theme.* | | | |
| **Tasks:**<br>***Task 2.1 UI Design*** **:** *Develop UI components.*<br>***Task 2.2 Test responsiveness*** **:** *Ensure responsiveness across devices.* | | | |
| **Deliverables**<br>***D2.1:*** *Responsive frontend for company profiles.* | | | |

| | | | |
|---|---|---|---|
| **WP 18:** LLM processing - Vector Database integration | | | |
| **Start date:** 03.03.2025 **End date:** 10.03.2025 | | | |
| **Leader:** | *Anıl Altuncu* | **Members involved:** | Yaşar Tatlıcıoğlu |
| **Objectives:** *Integrate LLM outputs with the vector database for seamless search and recommendation functionality. This includes creating pipelines for storing and querying embeddings.* | | | |
| **Tasks:**<br>***Task 2.1 Design the transactions*** **:** *Design data flow between LLM and vector database.*<br>***Task 2.2 Implementation*** **:** *Implement query and storage logic.* | | | |
| **Deliverables**<br>***D2.1:*** *Fully integrated LLM and vector database.* | | | |

| | | | |
|---|---|---|---|
| **WP 19:** Microservice Architecture Implementation | | | |
| **Start date:** 24.03.2025 **End date:** 21.04.2025 | | | |
| **Leader:** | *Ece Beyhan* | **Members involved:** | Yaşar Tatlıcıoğlu |
| **Objectives:** *Implement a microservices-based architecture to ensure scalability and modularity of the application. Each microservice will handle distinct functionalities, promoting independence and flexibility. This will prepare the app for future growth.* | | | |
| **Tasks:**<br>***Task 2.1 Microservice topic design*****:** *Design topics for microservices*<br>***Task 2.2 Testing*****:** *Deploy microservices and test interactions.* | | | |
| **Deliverables**<br>***D2.1:*** *Microservice-based application structure.* | | | |

| | | | |
|---|---|---|---|
| **WP 20:** Frontend app and backend services cloud deployment | | | |
| **Start date:** 14.04.2025 **End date:** 28.04.2025 | | | |
| **Leader:** | *Yaşar Tatlıcıoğlu* | **Members involved:** | Ece Beyhan |

| |
|---|
| **Objectives:** *Deploy the frontend application and backend services on a cloud platform for global accessibility. This includes optimizing deployment configurations for performance and cost-effectiveness.* |
| **Tasks:**<br>***Task 2.1 Azure configuration*:** *Configure Azure cloud hosting*<br>***Task 2.2 Deployment*:** *Deploy frontend and backend services.* |
| **Deliverables**<br>***D2.1:*** *Deployed application accessible online* |

### 4.3.1 Gantt Chart

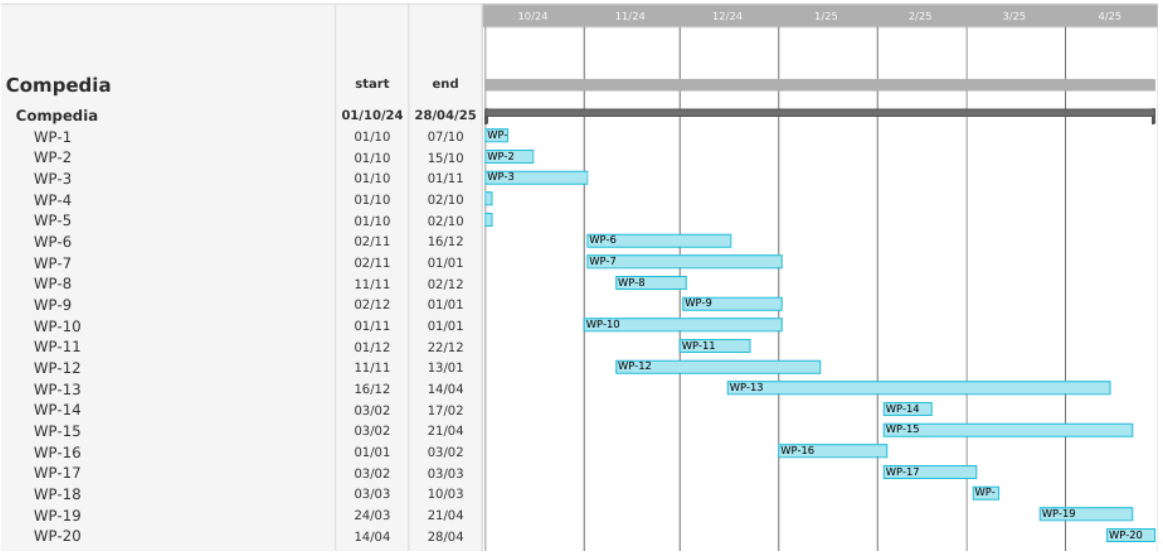


Fig. 30 Compedia Gantt Chart

## 4.4 Ensuring Proper Teamwork

With the labor of a team consisting of 5 people, ensuring proper teamwork plays a huge role in achieving success. To ensure productivity and efficiency, having a stable communication channel and properly using that channel is important. Using this channel properly is crucial in pursuance of the improvements of the project. A healthy development phase occurs with an equal workload for each group member. To ensure equality in contribution, we decided to divide the work according to the familiarity of group members. However, a trusting environment where if someone needs a hand, others will help is successfully created. The general workloads are provided below:

Serhat Yılmaz - Frontend Development

Bartu Albayrak - Backend Development

Ece Beyhan - DevOps and Infrastructure

Yaşar Tatlıcıoğlu - Vector Database Management

Anıl Altuncu - LLM Integration and NLP Tasks

As stated earlier, to ensure smooth collaboration, clear communication channels were established. A popular message application WhatsApp is used for scheduling meetings and sharing updates, while Discord and Zoom serve as the platforms for online meetings, which are scheduled as needed with relevant team members. Every two weeks, face-to-face meetings are held with all members whenever appropriate to review the progress of the project, assign tasks, and address implementation details. Task tracking is managed through Trello, which enables the team to monitor tasks at various stages such as To Do, In Progress, Done, and Succeeded. This ensures that each member has a clear understanding of their duties and can stay on track throughout the sprint.

## 4.5 Ethics and Professional Responsibilities

To ensure compliance with GDPR and other relevant privacy laws, the platform will allow users to view, edit, and delete their personal data, providing them with full control over their information. User verification mechanisms will be implemented to create a safe and trustworthy environment, ensuring that only authorized individuals can access sensitive features. Additionally, identity checks will be enforced for users interacting with critical areas such as claims and projects, actively preventing misuse and promoting a secure and reliable platform experience.

## 4.6 Planning for New Knowledge and Learning Strategies

While developing our project, we will make use of semantic search algorithms and not all team members have knowledge about semantic search. In order to get a better grasp of the subject we will be utilized from the online courses and also benefited from the YouTube playlists we have prepared. Moreover, knowledge of configuring and managing Milvus may be limited. Again we will take advantage of the online sources to get a better understanding of it.

# 5 Glossary

**AI-powered Search**
AI-powered search engines are designed to make searching smarter and more efficient, leveraging the power of artificial intelligence to deliver highly relevant results. [3]

**Azure**
A cloud computing service created by Microsoft that provides infrastructure, platform, and software solutions to support applications and services, including hosting, database management, and AI integration.

**BERT (Bidirectional Encoder Representations from Transformers)**
BERT language model is an open source machine learning framework for natural language processing [4].

**Compedia**
A proposed platform aimed at creating comprehensive, structured, and searchable profiles of companies using advanced NLP models and semantic search techniques to match user needs with businesses.

**Docker**
Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. [5]

**Embeddings**
Embeddings are numerical representations of real-world objects that machine learning (ML) and artificial intelligence (AI) systems use to understand complex knowledge domains like humans do. [6]

**Kafka**
Kafka is primarily used to build real-time streaming data pipelines and applications that adapt to the data streams. It combines messaging, storage, and stream processing to allow storage and analysis of both historical and real-time data. [7]

**LLM (Large Language Model)**
Large language models (LLMs) are machine learning models that can comprehend and generate human language text. They work by analyzing massive data sets of language. [8]

**MicrosoftSQL**
A relational database management system by Microsoft, used to store structured data such as basic company details in Compedia.

**Milvus**
An open-source vector database optimized for processing large-scale vector data and supporting semantic search operations efficiently.

**Natural Language Processing (NLP)**
Natural language processing (NLP) is a subfield of computer science and artificial intelligence (AI) that uses machine learning to enable computers to understand and communicate with human language. [9]

**React**
A JavaScript library for building user interfaces, particularly for single-page applications, ensuring a responsive and interactive user experience.

**Semantic Search**
Semantic search is a search engine technology that interprets the meaning of words and phrases. [10]

**State-of-the-Art (SOTA)**
The state of the art (SOTA or SotA, sometimes cutting edge, leading edge, or bleeding edge) refers to the highest level of general development, as of a device, technique, or scientific field achieved at a particular time. [11]

**Trello**
A task and project management tool used for team collaboration and organizing workflows visually via boards, lists, and cards.

**Vector Database**
A vector database, vector store or vector search engine is a database that can store vectors (fixed-length lists of numbers) along with other data items. [12]

**Zero-shot Classification**
Zero-shot classification models are large, pre-trained models that can classify images without being trained on a particular use case. [13]

## 6 References


[1] "Milvus: The High-Performance Vector Database Built for Scale." *Milvus*, https://milvus.io. Accessed 21 Nov. 2024.

[2] Object-Oriented Software Engineering, Using UML, Patterns, and Java, 2nd Edition, by Bernd Bruegge and Allen H. Dutoit, Prentice-Hall, 2004, ISBN: 0-13-047110-0.

[3] Sitecore. “What Is AI Search?” *Sitecore*, www.sitecore.com/explore/topics/artificial-intelligence/what-is-ai-search. Accessed 21 Nov. 2024.

[4] Lutkevich, Ben. "What Is the BERT Language Model?" *TechTarget*, 15 Nov. 2024, www.techtarget.com/searchenterpriseai/definition/BERT-language-model. Accessed 21 Nov. 2024.

[5] Sitecore. “What Is AI Search?” *Sitecore*, www.sitecore.com/explore/topics/artificial-intelligence/what-is-ai-search. Accessed 21 Nov. 2024.

[6] "Embeddings in Machine Learning." *Amazon Web Services (AWS)*, https://aws.amazon.com/what-is/embeddings-in-machine-learning/. Accessed 21 Nov. 2024.

[7] "What Is Apache Kafka?" Amazon Web Services (AWS), https://aws.amazon.com/what-is/apache-kafka/. Accessed 21 Nov. 2024.

[8] "What Is a Large Language Model?" *Cloudflare*, www.cloudflare.com/learning/ai/what-is-large-language-model/. Accessed 21 Nov. 2024.

[9] IBM. "Natural Language Processing (NLP) Solutions." *IBM*, www.ibm.com/natural-language-processing. Accessed 21 Nov. 2024.

[10] Elastic. “Semantic Search.” *Elastic Documentation*, Elastic, https://www.elastic.co/guide/en/serverless/current/elasticsearch-reference-semantic-search.html. Accessed 21 Nov. 2024.

[11] Wikipedia contributors. “State of the Art.” *Wikipedia*, 20 Nov. 2024, https://en.wikipedia.org/wiki/State_of_the_art. Accessed 21 Nov. 2024.

[12] Wikipedia contributors. “Vector Database.” *Wikipedia*, 20 Nov. 2024, https://en.wikipedia.org/wiki/Vector_database. Accessed 21 Nov. 2024.

[13] Gallagher, James. "What Is Zero-Shot Classification?" *Roboflow Blog*, 15 Nov. 2023, https://blog.roboflow.com/what-is-zero-shot-classification/%20model%20developed%20by%20OpenAI). Accessed 21 Nov. 2024.