Bilkent University
Department of Computer Engineering

Senior Design Project
T2404
Compedia

Detailed Design 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

10 March 2025

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.

Detailed Design 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.

1.1 Purpose of the system

Compedia aims to address the challenges businesses face in finding relevant partners and service providers in today's crowded marketplace. The system will serve as a comprehensive, reliable platform where businesses can discover and connect with potential partners based on their specific needs, regardless of company size or advertising budget.

The primary purpose of Compedia is to create a level playing field for businesses of all sizes through an AI-powered semantic search system. By implementing advanced search capabilities using GPT, Compedia will deliver highly relevant search results that match user-specific requirements, eliminating the problem of overwhelming and irrelevant search results that businesses currently encounter with traditional search engines.

Another key purpose of the system is to enhance the visibility of smaller businesses that often struggle to compete with established companies in traditional search environments. Compedia will provide equal opportunities for companies to showcase their services and capabilities through verified company profiles, ensuring that businesses are discovered based on relevance rather than advertising spending or search engine optimization tactics.

Compedia will also serve as a credibility verification platform. By implementing a root user approval system for company information and regular updates, the platform ensures that all company data is reliable and accurate. This verification process addresses the current issue of businesses making decisions based on potentially fake reviews or biased information, thereby improving the quality of business partnerships.

Additionally, the system aims to streamline the business decision-making process by providing a user-friendly, responsive interface accessible from any device with an internet connection. With features like a consistent search button in the topbar, intuitive navigation through a sidebar, and clear loading indicators for in-progress requests, Compedia will significantly reduce the time businesses spend searching for partners and increase the effectiveness of their decision-making process.

1.2 Design goals

1.2.1 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.

1.2.2 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.

1.2.3 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.

1.2.4 Interoperability

● The website will integrate seamlessly with GPT API and databases.

● The user that will create the company page can import files in standard formats (e.g. PDF) for company profile information.

● The website will be compatible with popular web browsers (e.g., Chrome, Firefox, Safari, Edge) and their latest two versions.

1.2.5 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].

1.2.6 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.

1.3 Definitions, acronyms, and abbreviations

AI: Artificial Intelligence
API: Application Programming Interface
BERT: a machine learning model for natural language processing
CSV: a simple file format used to store tabular data
GPT: a type of large language model for generating human-like text
JSON: JavaScript Object Notation
Kafka: An event streaming platform used for building real-time data pipelines
LLM: Large Language Model
Milvus: A vector database designed for similarity search and AI applications
REST API: An architectural approach for creating web services that uses HTTP methods
Root User: A user who creates and manages a specific company within the system
Semantic Search: A search technique that considers contextual meaning, not just keywords
Vector Database: A database that stores data as high-dimensional vectors and enables similarity search

1.4 Overview

Compedia is a comprehensive business discovery platform designed to transform how companies find and connect with relevant partners in today's crowded marketplace. Unlike traditional search engines that favor established businesses with larger advertising budgets, Compedia creates a level playing field where companies of all sizes can be discovered based on their actual relevance to specific business needs.

The platform utilizes advanced AI-powered semantic search capabilities through technologies like GPT and BERT to deliver highly targeted search results that match user-specific requirements. This approach eliminates the overwhelming number of irrelevant results typically encountered with conventional search methods, making the business partner discovery process significantly more efficient and effective.

Compedia implements a robust verification system where company information is reviewed and approved by root users, ensuring the reliability and accuracy of all data on the platform. This credibility verification process addresses the current challenges businesses face when making decisions based on potentially misleading reviews or biased information found elsewhere online.

The system features a user-friendly, responsive interface accessible from any device with an internet connection. With intuitive navigation elements like a consistent search button in the topbar and a sidebar for easy access to different sections, users can seamlessly explore the platform. Additionally, clear loading indicators for in-progress requests provide transparency throughout the user experience.

Overall, Compedia brings a revolutionary approach to business partner discovery by combining cutting-edge AI technology with a commitment to data reliability and user-centered design. By emphasizing relevance over advertising spending, providing equal visibility opportunities for businesses of all sizes, and ensuring information accuracy through verification processes, Compedia creates a more equitable, efficient, and trustworthy business ecosystem where meaningful partnerships can flourish based on genuine compatibility rather than marketing prominence.

2. Current software architecture

While detailed architectural information about Compedia's competitors like Built In, Clutch, GoodFirms, The Manifest, and Y Combinator's platform is not publicly available, analyzing their functionality and performance characteristics provides insights into common architectural patterns in the business discovery and B2B platform space.

Platforms like Clutch and GoodFirms appear to use microservice architectures to handle their extensive review systems, company listings, and search functionality. This architectural choice likely enables them to scale individual components independently based on usage patterns. For instance, Clutch's ability to handle thousands of detailed company profiles with complex filtering options suggests a robust distributed system with specialized services for search, user management, and content delivery.

Built In, which focuses on tech industry employment and company discovery, demonstrates advanced personalization capabilities that indicate sophisticated user profiling systems. Their implementation likely utilizes recommendation engines powered by machine learning models, similar to the AI-powered semantic search Compedia aims to implement. The responsive performance of Built In's platform, even when handling large volumes of company data and job listings, suggests an effective caching strategy and optimized database queries.

The Manifest, another competitor in this space, uses a content-heavy approach that would require efficient content management systems and potentially a headless CMS architecture to deliver a consistent experience across different devices. Their platform's search responsiveness indicates the implementation of search optimization techniques such as indexing and potentially vector-based search similar to what Compedia plans to implement with Milvus.

Y Combinator's platform, serving as both a startup discovery tool and accelerator program interface, seems like using a monolithic architecture with specialized modules for different functions. Their platform's stability and reliability suggest robust error handling and failover mechanisms that Compedia should consider when implementing critical features like company verification and semantic search.

From a front-end perspective, these competitors generally use modern JavaScript frameworks like React or Vue.js to deliver responsive user interfaces. They typically use REST APIs for backend communication, though some may have implemented GraphQL for more efficient data fetching patterns. The consistent performance of these platforms across different user loads suggests the implementation of load balancing, CDN integration for static content delivery, and potentially serverless computing for handling variable workloads.

For database implementations, these platforms likely use a combination of relational databases (PostgreSQL, MySQL) for structured company data and potentially NoSQL solutions (MongoDB, Elasticsearch) for handling unstructured content and powering search functionality. This hybrid database approach allows for both data integrity and search performance, considerations that will be important for Compedia's implementation of its vector database with Milvus.

Analyzing these architectural patterns informs Compedia's development strategy, particularly regarding the implementation of our AI-powered semantic search, scalability requirements, and the integration approach for our verification systems.

3. Proposed software architecture

3.1 Overview

Compedia is an AI-driven business discovery platform designed to help companies find and connect with relevant partners in a highly competitive market. Traditional search engines often prioritize larger

businesses with high advertising budgets, making it difficult for smaller companies to gain visibility. Compedia levels the field by utilizing semantic search powered by GPT and BERT, ensuring that search results are based on relevance.

The platform integrates a microservices-based architecture, enabling modularity, scalability, and high availability. Key components include an API Gateway for request routing, authentication and authorization services for secure access, and specialized services for handling company management, projects, and notifications. The backend leverages ASP.NET Core, MicrosoftSQL, and Kafka for event-driven communication, ensuring efficiency in data processing and user interactions.

As the Subsystem Decomposition outlines, Compedia follows a layered architecture where the user interacts with a responsive web interface, which in turn communicates with the backend microservices responsible for processing search queries, managing company data, and handling project collaborations. These microservices interact with PostgreSQL for structured data storage, Milvus for semantic search indexing, and S3 for media file storage. Additionally, Hardware/Software Mapping describes the infrastructure supporting the architecture, while Persistent Data Management and Access Control and Security ensure data integrity, user authentication, and system reliability.

3.2 Subsystem decomposition

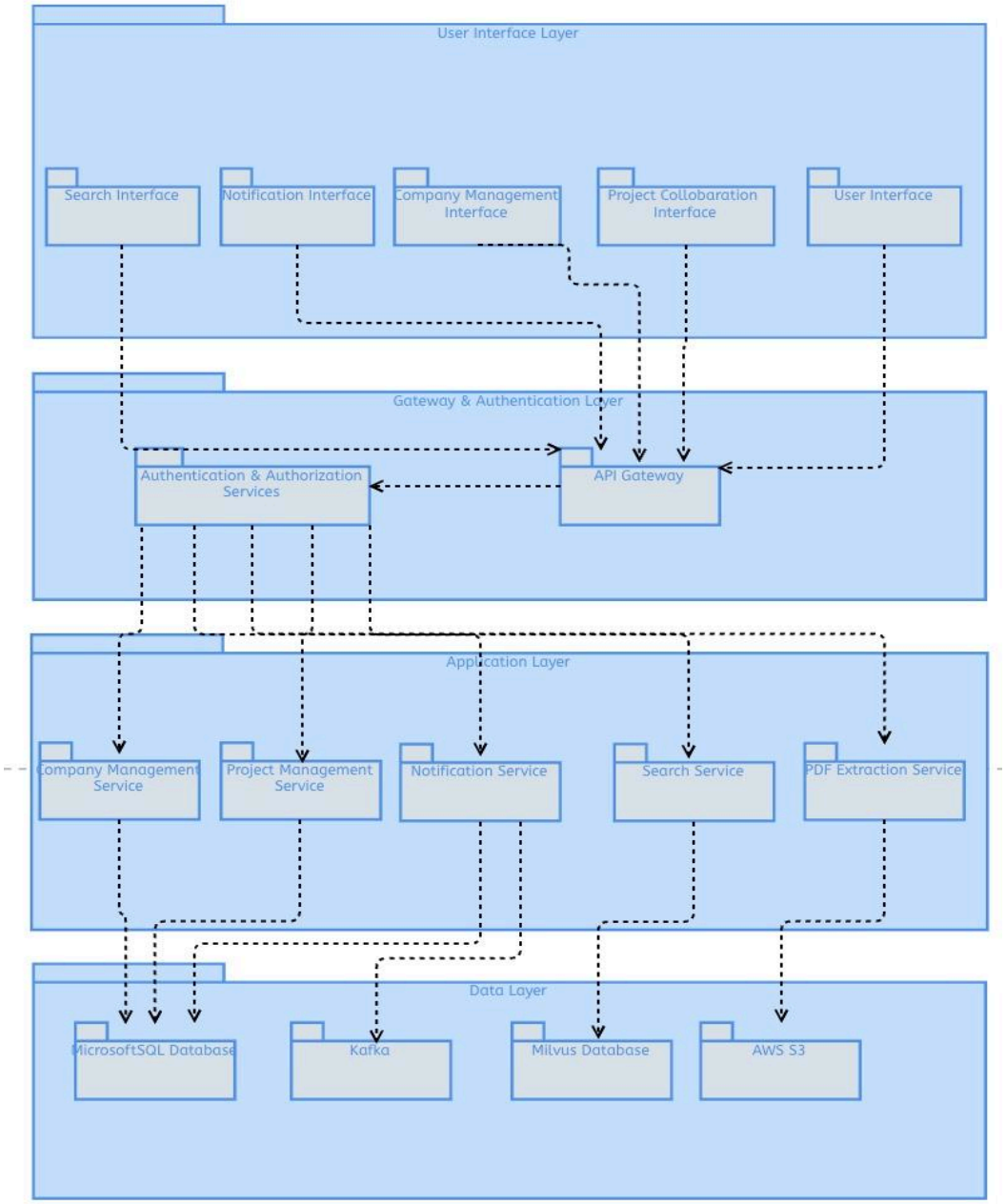

Figure 1: Subsystem Decomposition Diagram

We have designed a four-layer architecture, which consists of the User Interface Layer, Gateway and Authentication Layer, Application Layer, and Data Layer. Each layer is independent of the others and can only interact with the layer below it. This structure ensures a clear separation of concerns, making the system modular, scalable, and maintainable.

To enhance system security, we enforce role-based access control (RBAC) and authentication mechanisms through our Authentication & Authorization Service, which validates all incoming requests before they reach the backend. Additionally, for storing large-sized files such as PDFs and extracted documents, we utilize Azure Blob Storage, ensuring efficient file management and retrieval. The event-driven communication model, powered by Kafka, allows real-time updates and asynchronous messaging between microservices, improving overall system responsiveness and reliability.

3.3 Hardware/software mapping

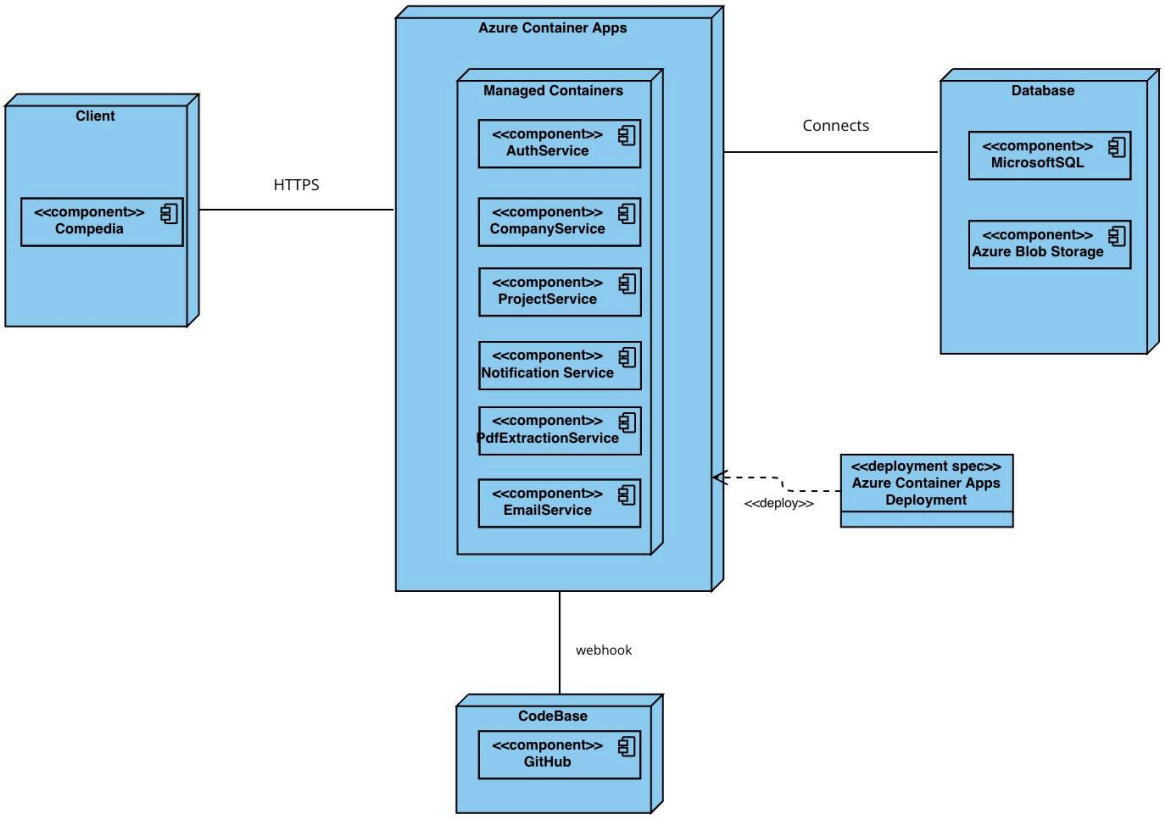

Figure 2: Deployment Diagram of Compedia

Compedia is designed to run as a fully web-based application, providing accessibility across desktop and mobile browsers. The client application will be built as a responsive web application.

All core business logic and data processing will reside on Microsoft Azure cloud servers. The backend microservices will be containerized and deployed on Azure Container Apps, eliminating the need for manual server management. API requests from the client application will be securely transmitted via

HTTPS to the backend services, which will be managed through Azure API Management. The API Gateway will efficiently route these requests to the appropriate microservices, ensuring smooth interactions between the frontend and backend.

For data storage, Compedia will rely on Microsoft SQL (Azure SQL Database) for structured business data, Milvus for AI-powered semantic search indexing, and Azure Blob Storage for handling file storage. To facilitate event-driven communication and notifications, Azure Event Hub will be employed to process real-time events asynchronously, ensuring efficient handling of system-wide updates and messages.

The deployment pipeline will be fully automated, with the codebase hosted on GitHub and deployment processes managed through Azure DevOps and GitHub Actions. This setup will ensure that any push to the main branch automatically triggers a new deployment, keeping the system continuously updated without manual intervention.

3.4 Persistent data management

To store structured application data, including company profiles, user information, project details, and collaboration records, Microsoft SQL (Azure SQL Database) will be used as the primary relational database. Microsoft SQL provides a highly reliable and scalable solution for managing structured business data while ensuring ACID compliance for transaction consistency.

For AI-powered semantic search and recommendation capabilities, Milvus will be used as the vector database. Milvus enables efficient storage and retrieval of high-dimensional embeddings for business discovery, allowing users to find relevant companies based on content similarity rather than simple keyword matching.

In addition to structured data, Compedia will require persistent storage for files such as company documents. These files will be stored in Azure Blob Storage, which provides scalable and durable object storage. To support real-time event-driven processing, Azure Event Hub will be used for managing system-wide notifications and asynchronous task execution. Azure Event Hub will allow microservices to efficiently communicate with each other without direct coupling, enabling a more scalable and resilient architecture.

3.5 Access control and security

We utilize JSON Web Tokens (JWTs) for managing access control. When a request reaches our API, we first verify whether a JWT has been provided by the user. If a token is present, we proceed to validate its authenticity. Once the user is successfully authenticated, we check their assigned role to ensure they have

the appropriate permissions for the requested action and confirm that they are only accessing or modifying their own data.

To enhance security, we do not store any confidential information in our database. Additionally, passwords are never stored in their original form; instead, they are securely encrypted. The plaintext version of a password is only accessible during the initial signup process and is never stored or displayed afterward.

4. Subsystem services

4.1 Client Subsystem

The client side of the project is a web application. The client subsystem consists of multiple pages, each with its own logic and user interface implementation. Navigation within the system is designed in a specific structure. Users can only access certain pages according to defined transitions from the page they are on. Once authenticated, users are directed to the Main Page, which serves as the central hub for navigation. From the Main Page, users can access key areas of the application.

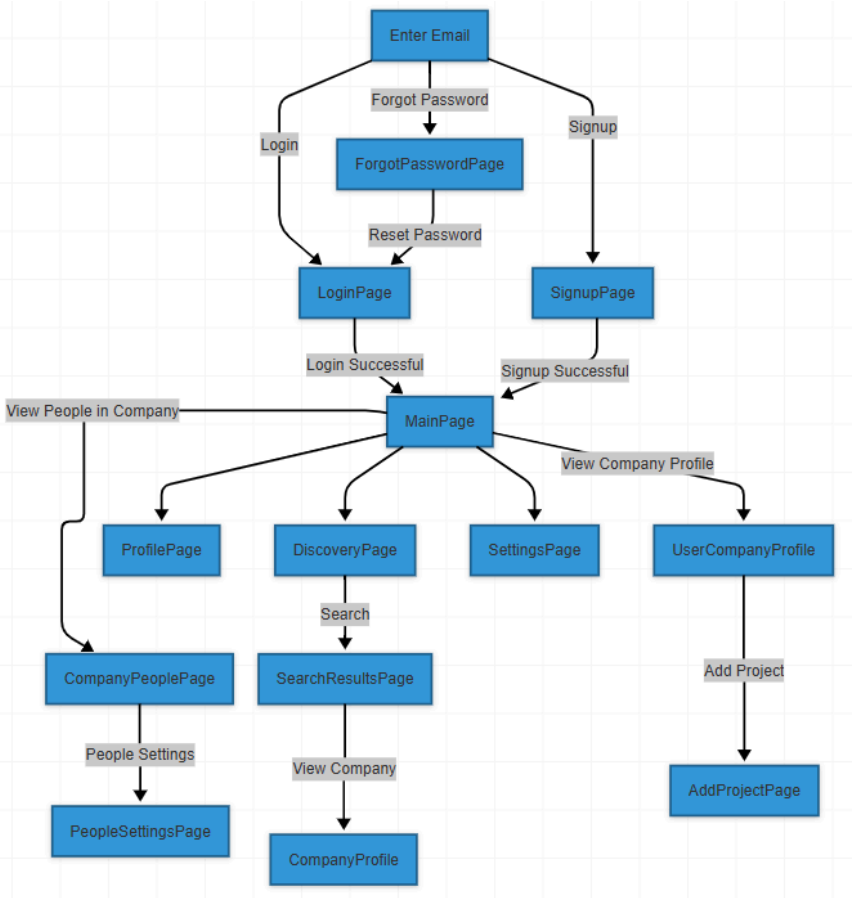

Figure 3: Client Subsystem

4.2 Backend Subsystem

The general structure of the backend is shown in the figure below. All incoming requests first reach the Gateway, which is responsible for routing the requests to the appropriate Microservice. The microservices process the requests and execute necessary database transactions.

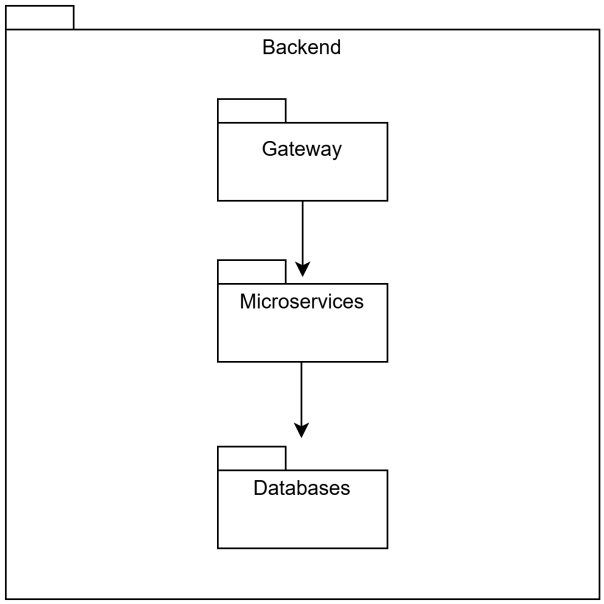

Figure 4: Backend structure

The next figure illustrates the internal structure of the Gateway. The Gateway first processes requests through its Authentication module, ensuring that only authenticated users can proceed. After successful authentication, the request is passed to the Router, which forwards it to the appropriate microservice.

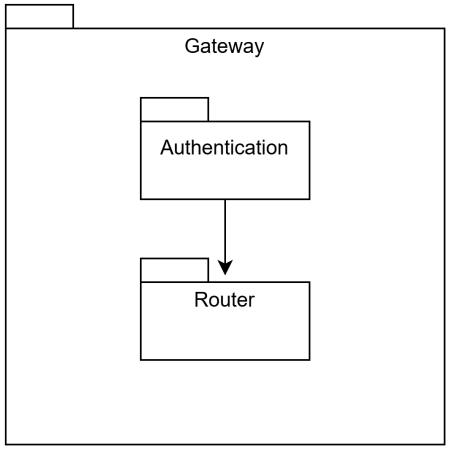

Figure 5: Gateway structure

Our system includes two microservices. The CompanyHubService microservice manages user authentication, company data, and projects. It has four main controllers:

- **AccountController**: Handles authentication via AuthService.
- **CompanyController**: Manages company-related actions through CompanyService.
- **ProjectController**: Handles project creation and updates through ProjectService.
- **UserController**: Manages user-related operations.

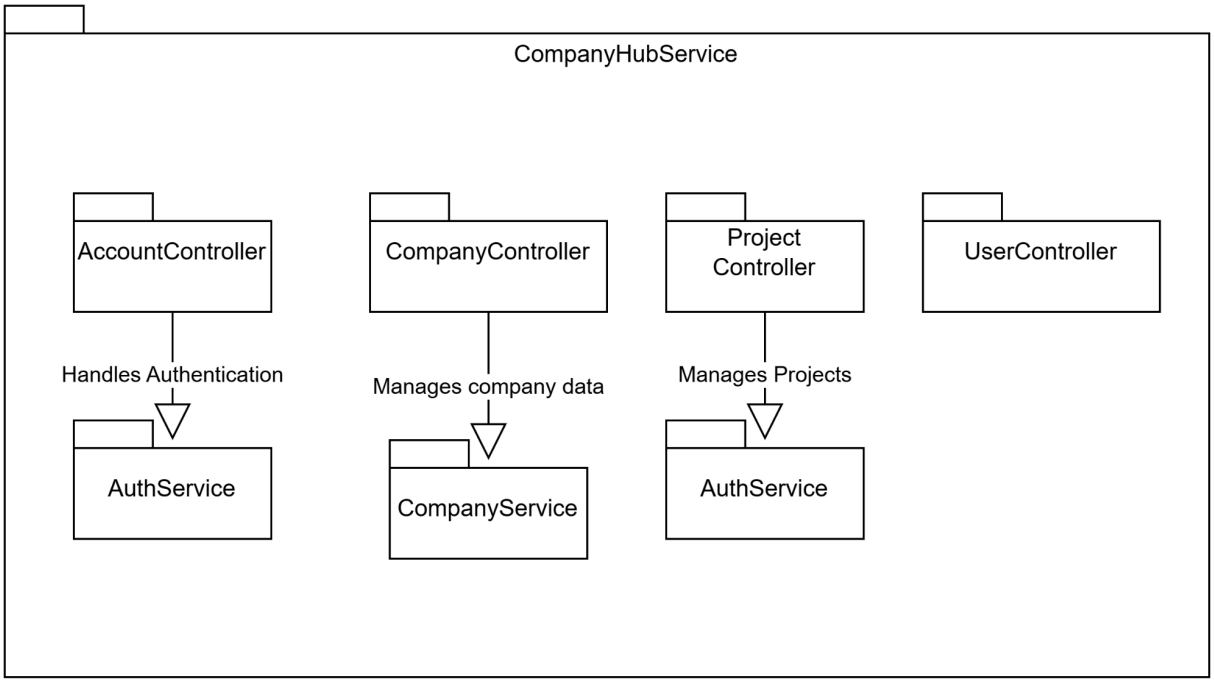

Figure 6: Company Hub Backend Service

The LLMService microservice is responsible for processing search queries, extracting structured data, and interacting with Milvus for company search.

- **Search Request Handler** processes the search query.
- **OpenAI API Client** is used for data extraction and embedding generation.
- **Milvus Database Handler** performs vector searches in **MilvusDB**.
- **GPT Extraction Module** extracts various structured fields from free-text user input.
- **OpenAI Embedding Model** converts extracted text into embeddings before querying Milvus.

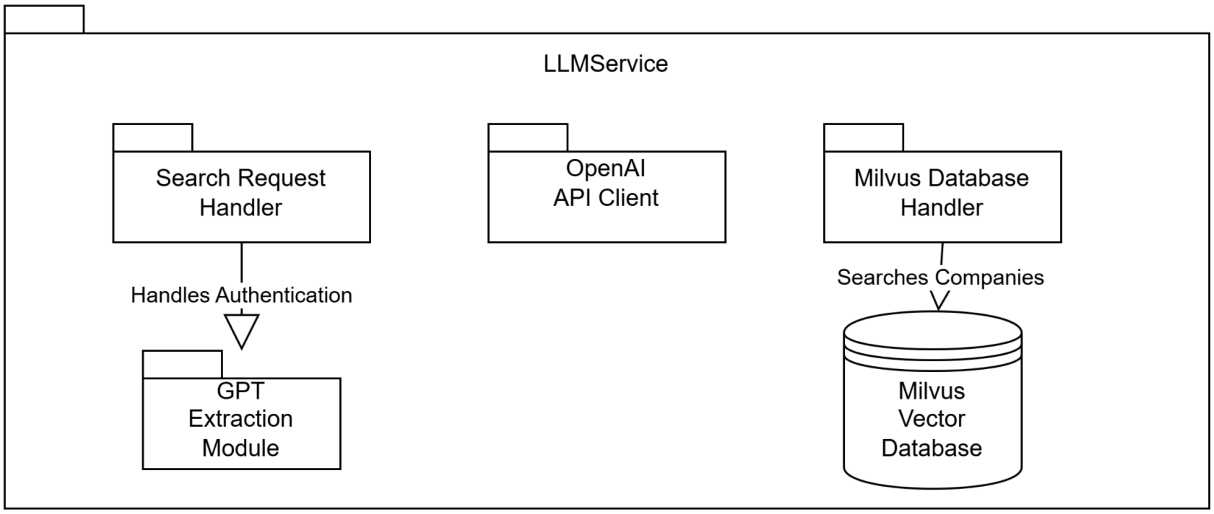

Figure 7: LLM Backend Service

5. Test Cases

Table 1: Test Cases of Compedia

Test ID	TC001	Category	Functional	Severity	Major
Requirement	When signing up, the password entered by the user should contain at least 6 characters including 1 uppercase, 1 lowercase and 1 symbol.				
Objective	Verify that the password entered by the user is valid				
Steps	1. Open the application's sign up page. 2. Try signing up with a password that does not include at least 6 characters 3. Try signing up with a password that does not include at least 1 uppercase letter 4. Try signing up with a password that does not include at least 1 lowercase letter 5. Try signing up with a password that does not include at least 1 symbol				
Expected	Invalid passwords should not be accepted and relevant fail notification should appear on the screen.				
Date-Result	TBD				

Test ID	TC002	Category	Functional	Severity	Major
Requirement	System must redirect unauthenticated access attempts to the login page.				
Objective	Verify unauthenticated access prevention				
Steps	1. Open the application in a browser (without logging in). 2. Directly enter URLs for authenticated pages into the browser's address bar 3. Observe the application's response to each request.				
Expected	User redirected to login page				
Date-Result	TBD				

Test ID	TC003	Category	Functional	Severity	Major
Requirement	System must respond within 5 seconds under normal usage				
Objective	Measure application response time under typical load				
Steps	1. Log in to the application. 2. Navigate through pages. 3. Perform actions. 4. Observe how long it takes for pages/actions to complete.				
Expected	Response within 5 seconds consistently				
Date-Result	TBD				

Test ID	TC004	Category	Functional	Severity	Major
Requirement	Users must log in successfully with valid credentials				
Objective	Verify successful login functionality				
Steps	1. Open login page 2. Enter valid credentials 3. Click on login				
Expected	User logged in successfully				
Date-Result	TBD				

Test ID	TC005	Category	Non-Functional	Severity	Minor
Requirement	Application must be compatible across all major browsers (Chrome, Firefox, Safari, Edge)				
Objective	Verify browser compatibility				
Steps	1. Open application in Chrome, Firefox, Safari, Edge 2. Perform basic actions				
Expected	Application functions correctly in all browsers				
Date-Result	TBD				

Test ID	TC006	Category	Functional	Severity	Major
Requirement	Logout action must effectively terminate the user's session				
Objective	Verify logout functionality				
Steps	1. Log into application 2. Perform logout 3. Attempt accessing authenticated pages				
Expected	Session terminated; user unable to access authenticated pages without re-login				
Date-Result	TBD				

Test ID	TC007	Category	Functional	Severity	Major
Requirement	System must enforce access control for protected areas				
Objective	Verify access control restrictions				
Steps	1. Open browser 2. Directly enter admin page URL				
Expected	Unauthorized pages inaccessible				
Date-Result	TBD				

Test ID	TC008	Category	Functional	Severity	Major
Requirement	Import functionality must accurately process valid PDF files for company profile creations made by user				
Objective	Verify accurate data import functionality				
Steps	1. Log into the application 2. Go to create company profile section 3. Upload a PDF file which contains relevant company information				
Expected	Data imported successfully and correctly reflected in the text fields of company creation page				
Date-Result	TBD				

Test ID	TC009	Category	Functional	Severity	Major
Requirement	Import functionality must reject invalid PDF files				
Objective	Verify handling of invalid data imports				
Steps	1. Log into application 2. Go to create company profile page 3. Attempt upload of incorrectly formatted or corrupted PDF file				
Expected	System rejects import and provides clear error message				
Date-Result	TBD				

Test ID	TC010	Category	Non-Functional	Severity	Critical
Requirement	Role-based permissions must be accurately enforced				
Objective	Validate accurate enforcement of role-based permissions				
Steps	1. Test system actions with varying roles and permissions				
Expected	Roles and permissions accurately enforced				
Date-Result	TBD				

Test ID	TC011	Category	Non-Functional	Severity	Major
Requirement	AI-based semantic search queries must return results within 5 seconds				
Objective	Measure performance of semantic search functionality				
Steps	1. Enter various semantic queries in search bar 2. Execute searches and measure response times				
Expected	Queries consistently completed within 5 seconds				
Date-Result	TBD				

Test ID	TC012	Category	Functional	Severity	Major
Requirement	Company profile updates must be correctly saved and be reflected.				
Objective	Verify company profile update functionality				
Steps	1. Login and navigate to existing company profile 2. Modify profile details (e.g., address, services) 3. Save changes				
Expected	Updates accurately saved and immediately visible				
Date-Result	TBD				

Test ID	TC013	Category	Functional	Severity	Critical
Requirement	Semantic search must return highly relevant company profiles				
Objective	Confirm accuracy of semantic search functionality				
Steps	1. Enter different semantic search queries 2. Review returned results				
Expected	Relevant company profiles returned matching semantic intent				
Date-Result	TBD				

Test ID	TC014	Category	Functional	Severity	Major
Requirement	User enters all the necessary information for the company profile creation				
Objective	Verify required fields are filled in the company profile creation				
Steps	1. Open company create page 2. Enter information into the text fields except the company name and try to submit the form 3. Enter information into the text fields except the foundation year and try to submit the form 4. Enter information into the text fields except the address and try to submit the form 5. Enter information into the text fields except the email and try to submit the form				
Expected	Company profile should not be not created and a relevant fail submit message should be reflected on the screen.				
Date-Result	TBD				

Test ID	TC015	Category	Non-Functional	Severity	Critical
Requirement	The failure of one microservice must not crash the entire system.				
Objective	Verify that system resilience and failover mechanisms work as intended.				
Steps	1. Simulate a failure in one microservice (e.g., the LLMService). 2. Monitor system behavior and error handling.				
Expected	Other services should remain operational, and the system should recover gracefully.				
Date-Result	TBD				

Test ID	TC016	Category	Non-Functional	Severity	Major
Requirement	All data must be transmitted securely using HTTPS.				
Objective	Ensure encryption of data in transit.				
Steps	1. Try to log in to the application with user credentials 2. Use network monitoring tools to inspect data transmissions between the client and server.				
Expected	All communications should be encrypted with HTTPS, ensuring secure data transfer.				
Date-Result	TBD				

Test ID	TC017	Category	Functional	Severity	Major
Requirement	JWT tokens must expire as intended and require re-authentication.				
Objective	Verify that expired tokens are rejected and proper re-authentication is enforced.				
Steps	1. Log in to obtain a valid token. 2. Allow the token to expire. 3. Attempt to access a protected resource.				
Expected	Access should be denied, prompting the user to log in again.				
Date-Result	TBD				

Test ID	TC018	Category	Functional	Severity	Major
Requirement	User sessions must automatically time out after a period of inactivity.				
Objective	Validate that sessions are terminated correctly after the timeout period.				
Steps	1. Log in and remain idle for a period exceeding the session timeout threshold. 2. Attempt to navigate to a protected page.				
Expected	The system should redirect the user to the login page due to session expiration.				
Date-Result	TBD				

Test ID	TC019	Category	Non-functional	Severity	Critical
Requirement	All user inputs must be sanitized to prevent SQL injection attacks.				
Objective	Ensure the application is secure against SQL injection.				
Steps	1. Input SQL injection code in login and search fields. 2. Observe the console messages				
Expected	The input should be sanitized, and no unauthorized access or data leakage should occur.				
Date-Result	TBD				

Test ID	TC020	Category	Non-functional	Severity	Major
Requirement	API endpoints should implement rate limiting to prevent abuse.				
Objective	Verify that excessive API calls are throttled or blocked.				
Steps	1. Log in to the application 2. Rapidly send multiple API requests in succession.				
Expected	Excessive requests should be throttled or rejected, with a rate-limit error message returned.				
Date-Result	TBD				

Test ID	TC021	Category	Non-functi onal	Severity	Major
Requirement	The interface must render appropriately on devices with different screen resolutions.				
Objective	Ensure proper layout and usability across desktops, laptops, and tablets.				
Steps	1. Open the application on devices with varying resolutions or use browser developer tools to simulate them.				
Expected	The layout should adapt without visual or functional issues.				
Date-Result	TBD				

Test ID	TC022	Category	Functional	Severity	Minor
Requirement	The sidebar and topbar navigation menus must provide smooth transitions.				
Objective	Verify that all navigation elements correctly direct the user.				
Steps	1. Click on each menu item in the sidebar 2. Click on each menu item in the topbar.				
Expected	The application should navigate smoothly to the intended pages.				
Date-Result	TBD				

Test ID	TC023	Category	Functional	Severity	Minor
Requirement	The search bar should be consistently available across all pages.				
Objective	Ensure the search feature is accessible at all times.				
Steps	1. Log in to the application 2. Navigate through various pages of the application.				
Expected	The search bar should remain visible and functional on every page.				
Date-Result	TBD				

Test ID	TC024	Category	Functional	Severity	Major
Requirement	Semantic search must tolerate minor typos or misspellings.				
Objective	Validate the robustness of the search algorithm.				
Steps	1. Enter search queries with minor misspellings.				
Expected	The returned results should remain relevant despite the typos.				
Date-Result	TBD				

Test ID	TC025	Category	Functional	Severity	Major
Requirement	The system must prevent the creation of duplicate company profiles.				
Objective	Enforce uniqueness constraints for company profiles.				
Steps	1. Log in to the application with a root user's credentials 2. Attempt to create a new company profile using an already existing company's information				
Expected	The system should reject the duplicate entry with a clear error message.				
Date-Result	TBD				

Test ID	TC026	Category	Functional	Severity	Major
Requirement	The root user must be able to delete a company profile he owns.				
Objective	Validate that deletion operations work correctly.				
Steps	1. Log in and select the existing company profile that you own. 2. Initiate the delete action.				
Expected	The profile should be removed from the system with a confirmation message.				
Date-Result	TBD				

Test ID	TC027	Category	Functional	Severity	Major
Requirement	The system must deliver real-time notifications when it is needed.				
Objective	Verify that critical events trigger immediate notifications.				
Steps	1. Perform an action that generates a notification (e.g., a project request is received).				
Expected	The notification should be delivered in real time to the appropriate user or company.				
Date-Result	TBD				

Test ID	TC028	Category	Functional	Severity	Major
Requirement	Users must have the ability to delete their accounts.				
Objective	Validate that the account deletion process works correctly.				
Steps	1. Log in and navigate to the account settings. 2. Initiate the account deletion process.				
Expected	The account should be removed, and a confirmation should be provided.				
Date-Result	TBD				

Test ID	TC029	Category	Functional	Severity	Major
Requirement	Users must be able to reset their forgotten passwords securely.				
Objective	Validate the password reset workflow.				
Steps	1. Use the “Forgot Password” link on the login page. 2. Follow the reset instructions received via email.				
Expected	The user should be able to reset their password and log in with the new credentials.				
Date-Result	TBD				

Test ID	TC030	Category	Functional	Severity	Major
Requirement	New user registrations must include an email verification step.				
Objective	Ensure that new users verify their email addresses before gaining full access.				
Steps	1. Register a new account. 2. Check for and follow the verification link sent via email.				
Expected	The account should be marked as verified upon completing the process.				
Date-Result	TBD				

Test ID	TC031	Category	Functional	Severity	Major
Requirement	After a password reset, existing sessions must be invalidated.				
Objective	After a password reset, existing sessions must be invalidated.				
Steps	1. Log in to the application 2. Reset the password 3. Attempt to continue using the old session.				
Expected	The old session should be terminated, requiring a new login.				
Date-Result	TBD				

Test ID	TC032	Category	Functional	Severity	Major
Requirement	Administrators must be able to change user roles.				
Objective	Validate role management and permission updates.				
Steps	1. Log in as an administrator. 2. Change the role of a selected user and verify updated permissions.				
Expected	The user's role should be updated and corresponding access rights should be adjusted.				
Date-Result	TBD				

Test ID	TC033	Category	Functional	Severity	Major
Requirement	Unverified users should not create company profiles				
Objective	Verify that only the verified users create company profiles.				
Steps	1. Log in the application with an unverified user's credentials 2. Open the company create page 3. Fill the form and attempt to create the company				
Expected	Company should not be created and an appropriate error message should be reflected				
Date-Result	TBD				

Test ID	TC034	Category	Functional	Severity	Major
Requirement	An email should be sent to the company whose profile has been created by a user.				
Objective	Ensure that the company is aware of that a user has created their account on the application				
Steps	1. Log in to the application with the credentials of a verified user 2. Navigate to the company create page and submit the form 3. Check the inbox of the company's email address whose profile has been created				
Expected	An email should be sent to the company's email address				
Date-Result	TBD				

Test ID	TC035	Category	Functional	Severity	Critical
Requirement	Only the verified companies can create projects.				
Objective	Verify that unverified companies cannot create projects				
Steps	1. Log in to the application with the credentials of an unverified company 2. Go to the create project page and attempt to create it.				
Expected	Project should not be created and an error message should be shown.				
Date-Result	TBD				

Test ID	TC036	Category	Functional	Severity	Major
Requirement	Only the root user of the company can add people to the company.				
Objective	Ensure that the company's people management is under control of authorized user(s).				
Steps	1. Log in to the application with the credentials of a normal user (not root user). 2. Go to the people page of the user's company 3. Click on add people button on the page and attempt to add people				
Expected	An error message should be shown that only the authorized users (root users) can do it and should not allow operation.				
Date-Result	TBD				

Test ID	TC037	Category	Functional	Severity	Major
Requirement	A user should be added to the company's people if he clicks on the invitation link sent to his email address by the root user.				
Objective	Ensure that users are added to the company's people after clicking on the invitation link.				
Steps	1. Send invitation link of a company to a person's email address 2. Click on the link sent via email 3. Check the company's people page				
Expected	The added person should appear on the people list of the company.				
Date-Result	TBD				

Test ID	TC038	Category	Functional	Severity	Critical
Requirement	A project created by a company should be approved by the partner company to make it visible on both companies' profile pages.				
Objective	Verify that only mutually accepted projects are visible.				
Steps	1. Create a project and add a partner company 2. Without the approval of the partner company, check both companies' profile pages				
Expected	The recently created project should not be visible on profile pages yet.				
Date-Result	TBD				

Test ID	TC039	Category	Functional	Severity	Critical
Requirement	Project completion should be approved by both companies to make it marked completed on profile pages of companies.				
Objective	Ensure that both companies accept the end of the project.				
Steps	1. Create a project and add a partner company 2. Mark it completed for one company 3. Check the both companies' profile pages				
Expected	The project should be seen as not completed yet on both profile pages.				
Date-Result	TBD				

Test ID	TC040	Category	Functional	Severity	Critical
Requirement	A project should be completed to access the rating system.				
Objective	Verify that the companies can rate their partners only if their project is completed.				
Steps	1. Create a project by adding partner company 2. Try to rate partner company				
Expected	Rate system should not be allowed and an error message should be shown.				
Date-Result	TBD				

Test ID	TC041	Category	Functional	Severity	Major
Requirement	A client company must be able to submit a new project and have it stored with a "Pending" status while notifying the appropriate provider company.				
Objective	Verify that the project submission process saves all details correctly and triggers a notification.				
Steps	1. Log in as a client company user with the proper "edit" permission. 2. Navigate to the "Create Project" section. 3. Enter all required project details and submit the form. 4. Check the project record in the database for a "Pending" status. 5. Verify that a notification is sent to the designated provider company.				
Expected	The project should be saved with a "Pending" status, and the provider company should be notified of the new project submission.				
Date-Result	TBD				

Test ID	TC042	Category	Functional	Severity	Major
Requirement	When a provider company accepts a project, the system must update the status to “Accepted” and notify the client company.				
Objective	Confirm that acceptance actions are correctly handled and reflected.				
Steps	1. Log in as a provider company user. 2. Access a project submitted by a client (currently “Pending”). 3. Accept the project using the provided action (e.g., click an “Accept” button). 4. Verify the project’s status updates to “Accepted” in the system. 5. Check that the client company receives a notification about the acceptance.				
Expected	The project status should be changed to “Accepted” and both companies should receive appropriate notifications.				
Date-Result	TBD				

Test ID	TC043	Category	Non-functional	Severity	Minor
Requirement	Login operations should complete within 2 seconds under normal conditions.				
Objective	Ensure that the application responds a login request within 2 seconds				
Steps	1. Log in with valid credentials multiple times under normal load. 2. Measure and record the response time for each login attempt.				
Expected	Each login should be completed in 2 seconds or less.				
Date-Result	TBD				

Test ID	TC044	Category	Non-functional	Severity	Minor
Requirement	Common UI elements (such as headers, footers, and navigation menus) must display uniformly across all pages.				
Objective	Ensure a better user experience in overall of the application				
Steps	1. Open multiple pages of the application (homepage, profile, search, etc.). 2. Visually verify that UI elements (fonts, colors, layout) remain consistent.				
Expected	All pages should show the same consistent design and layout for these elements.				
Date-Result	TBD				

Test ID	TC045	Category	Non-functional	Severity	Minor
Requirement	The application must display clear and user-friendly error messages for common issues (e.g., invalid login, form submission errors).				
Objective	Verify clear error messages are displayed to users				
Steps	1. Intentionally trigger errors by submitting invalid data (wrong password, incomplete forms). 2. Observe the error messages shown by the system.				
Expected	Error messages are simple, clear, and provide guidance on how to correct the issue.				
Date-Result	TBD				

Test ID	TC046	Category	Non-functional	Severity	Critical
Requirement	Sensitive data (e.g., user passwords, company details) must be stored encrypted in the database.				
Objective	Ensure that data stored in the database is encrypted to prevent unauthorized access.				
Steps	1. Register a new user and create a company profile containing sensitive data. 2. Access the database directly (with appropriate permissions) to inspect stored data. 3. Check that sensitive fields (e.g., passwords, email addresses) are not stored in plain text.				
Expected	Sensitive data should be stored in an encrypted format and direct database inspection should not reveal plain text information.				
Date-Result	TBD				

Test ID	TC047	Category	Functional	Severity	Critical
Requirement	Only admins can see the admin dashboard of the application.				
Objective	Verify that non-authorized people cannot see the sensitive information and do not have management permissions.				
Steps	1. Log in to the application with a non-admin account. 2. Try to reach the admin dashboard.				
Expected	The admin dashboard should not be reachable.				
Date-Result	TBD				

Test ID	TC048	Category	Functional	Severity	Major
Requirement	Only a project's companies' people can access project related documents.				
Objective	Verify that other than the relevant people, others do not have access to the document management system of a project.				
Steps	1. Go to a project's page in which other companies are involved other than yours. 2. Attempt to reach the documents of that project.				
Expected	The documents should not be reachable.				
Date-Result	TBD				

Test ID	TC049	Category	Functional	Severity	Major
Requirement	Companies should be able to import PDF files while creating projects.				
Objective	Verify that the companies can add PDF files to the system in addition to filling out the text fields manually in project creation.				
Steps	1. Go to the create project page. 2. Upload a PDF file 3. Check the text fields in the page				
Expected	The relevant text fields should be filled according to the PDF file.				
Date-Result	TBD				

Test ID	TC050	Category	Functional	Severity	Major
Requirement	File uploads must enforce a maximum file size limit while project creation.				
Objective	Validate that oversized files are not accepted.				
Steps	1. Go to create a project page. 2. Attempt to upload an oversized file.				
Expected	Upload should not be successful and a relevant error message should be displayed.				
Date-Result	TBD				

6. Consideration of Various Factors in Engineering Design

6.1 Constraints

6.1.1 Public Health

The Compedia platform features AI-powered semantic search and company information that could potentially include health-related businesses or services. To ensure responsible handling of health-related information, our platform must be designed with clear limitations and disclaimers. Any health-related company information displayed on Compedia should be clearly marked as informational only and not to be construed as medical advice. The semantic search should be constrained to avoid generating health recommendations or medical interpretations when users search for health-related businesses. Additionally, verification protocols for health industry companies will be more strict, requiring proper certification and credentials before approval by root users. All company profiles related to healthcare will include prominent disclaimers emphasizing that Compedia does not validate medical claims or endorse specific health treatments.

6.1.2 Public Safety

Public safety considerations are paramount in the design of Compedia's platform. All user data and company information are stored securely with robust encryption methods and we made comprehensive access control systems to prevent unauthorized access to sensitive company data. Our AI-powered search system undergoes rigorous testing to prevent potential misuse, such as searching for illegal services or products. We implement strict verification procedures for company information to ensure the legitimacy of businesses listed on our platform, protecting users from potential scams or fraudulent activities. Additionally, our root user approval system adds an extra layer of security by manually reviewing sensitive company information before it becomes publicly available. To further enhance safety, we will add content moderation tools to filter out inappropriate or harmful content in company descriptions and user interactions. All users must agree to our Terms of Service and Privacy Policy before registration, which clearly outlines acceptable use policies and data handling practices. In case of safety concerns, we provide easy-to-access reporting mechanisms for users to flag suspicious activities or potentially harmful content.

6.1.3 Global Factors

Global factors significantly influence Compedia's design as a business discovery platform intended for international use. Our system accommodates multiple languages and regional business terminology to serve users across different geographical locations. We've designed the platform with awareness of various international business regulations and standards, ensuring compliance with global commerce laws, including GDPR compliance for European users and similar regulations for other regions. The search functionality incorporates understanding of region-specific business practices and terminology, allowing for more accurate matching regardless of cultural or geographical context. Additionally, the platform supports

international date formats, time zones, and measurement systems to provide a seamless experience for users worldwide. We've also considered global digital infrastructure disparities, optimizing our platform to function effectively even in regions with limited internet bandwidth while maintaining core functionality across diverse technological environments.

6.1.4 Cultural Factors

Cultural factors play a crucial role in Compedia's design approach as we aim to create a platform that respects and adapts to diverse business cultures worldwide. Our semantic search algorithms are designed to understand and interpret culturally-specific business terminology and practices, ensuring relevant results regardless of cultural context. The platform's interface incorporates culturally neutral design elements while allowing for localization when appropriate. We've implemented rigorous content review processes to identify and prevent potentially offensive or culturally insensitive material in company descriptions and user interactions. Additionally, our verification system accommodates different business documentation standards across cultures, ensuring equitable verification opportunities for companies from various regions. The platform also supports multiple communication styles and business etiquette norms in how information is presented and how businesses can represent themselves, recognizing that business communication varies significantly across cultures.

6.1.5 Social Factors

Social considerations have a high impact on Compedia's development, centered on building trust through verified profiles, ethical AI implementation, and transparent business interactions. Our verification system creates accountability and reliability in the platform's information, addressing the social need for trustworthy business data. The AI models powering our semantic search adhere to ethical guidelines that prioritize fairness and prevent manipulation. The platform encourages transparent communication between businesses through standardized company profiles and clear verification indicators. These social trust mechanisms are fundamental to our value proposition and significantly shape our feature development priorities.

6.1.6 Environmental Factors

Environmental factors have been considered in Compedia's technical infrastructure and operational decisions. We've opted for cloud services that prioritize energy efficiency and use renewable energy sources where possible, minimizing the carbon footprint associated with our server operations. The AI models powering our semantic search are optimized for computational efficiency, reducing unnecessary processing and associated energy consumption. We implement intelligent caching strategies to minimize redundant requests and processing, further reducing energy usage. Our development and deployment processes follow green software engineering practices, including efficient code that minimizes resource utilization. Additionally, we've designed the platform to encourage digital business interactions and virtual partnerships, potentially reducing the need for physical travel and associated environmental impacts. The

company verification system includes optional fields for businesses to showcase their environmental credentials and sustainability practices, encouraging eco-conscious business partnerships.

6.1.7 Economical Factors

Economic factors significantly influence our technological choices and development strategy for Compedia. Budget constraints have guided our selection of open-source technologies and cloud services that offer scalable pricing models aligned with our growth projections. Our system architecture is designed for scalability, allowing infrastructure expenses to grow proportionally with user adoption and preventing premature over-investment. Database technologies were selected based on performance-to-cost ratios, with particular attention to Milvus vector database for its efficient scaling capabilities. We've also carefully evaluated the cost-benefit ratio of implementing advanced AI capabilities like semantic search, balancing the improved user experience against infrastructure expenses. Our development approach emphasizes modular architecture and incremental feature deployment, allowing us to allocate resources strategically and demonstrate value before significant investment in any particular component. We've also considered the economic sustainability of our business model, ensuring that our verification processes and platform features can be maintained without prohibitive operational costs. Additionally, we've designed the platform to provide value to businesses of all sizes, including consideration for smaller companies with limited budgets for business development and partnership discovery. This approach not only aligns with our mission but also expands our potential user base and revenue opportunities.

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

Table 2: Summary of Factors Affecting Compedia

6.2 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.

7. Teamwork Details

7.1 Contributing and functioning effectively on the team

Effective teamwork is essential for the success of any software development project. Our team consists of five senior Computer Science students, each specializing in different technical areas. Given our varying schedules, we make an effort to coordinate meetings efficiently and ensure that collaboration is both productive and convenient for all members.

To maintain an organized workflow, we conduct weekly meetings -both in-person and via Zoom- to review progress, discuss challenges, and plan upcoming tasks. Additionally, extra meetings are scheduled whenever necessary to work on specific issues or collaborative tasks. Other than general team meetings, smaller groups occasionally meet to work on focused tasks that require intensive collaboration.

For project management, we utilize industry-standard tools, including Trello for task tracking, GitHub for version control, WhatsApp for instant communication, and Google Docs for shared documentation. These tools enable us to stay aligned and streamline our development process effectively.

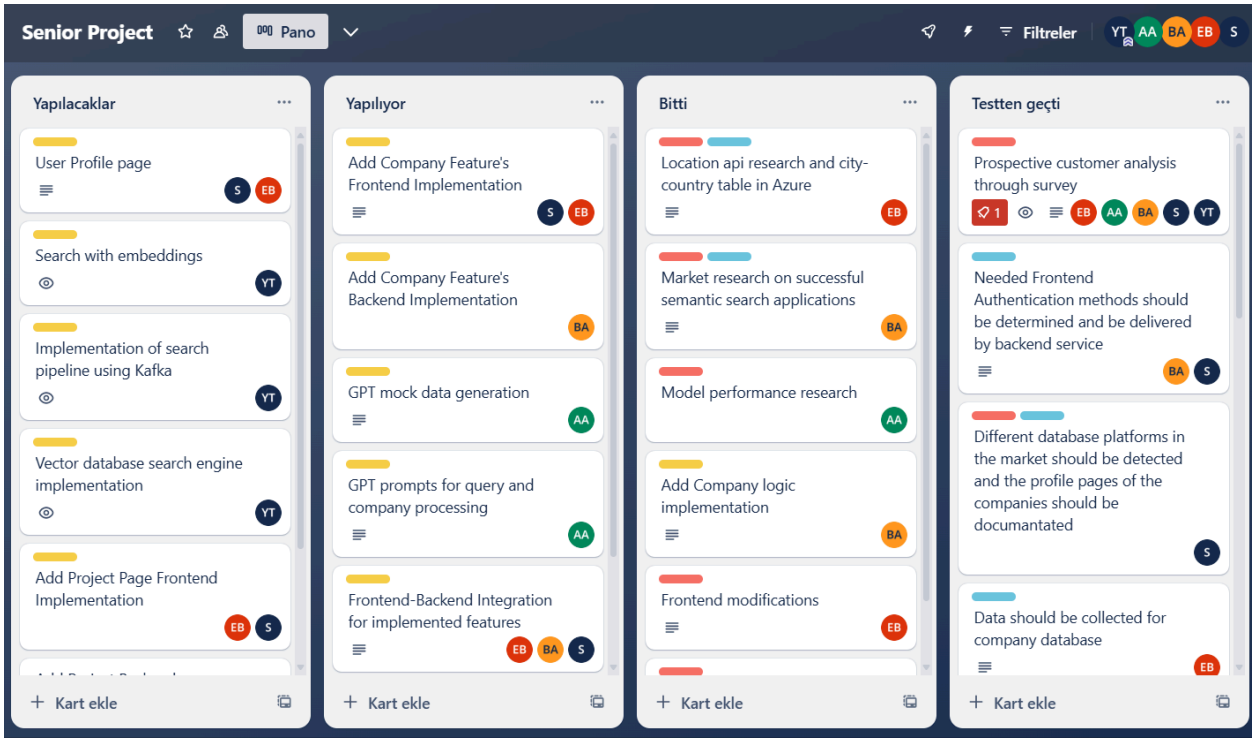

Figure 8: Trello Dashboard

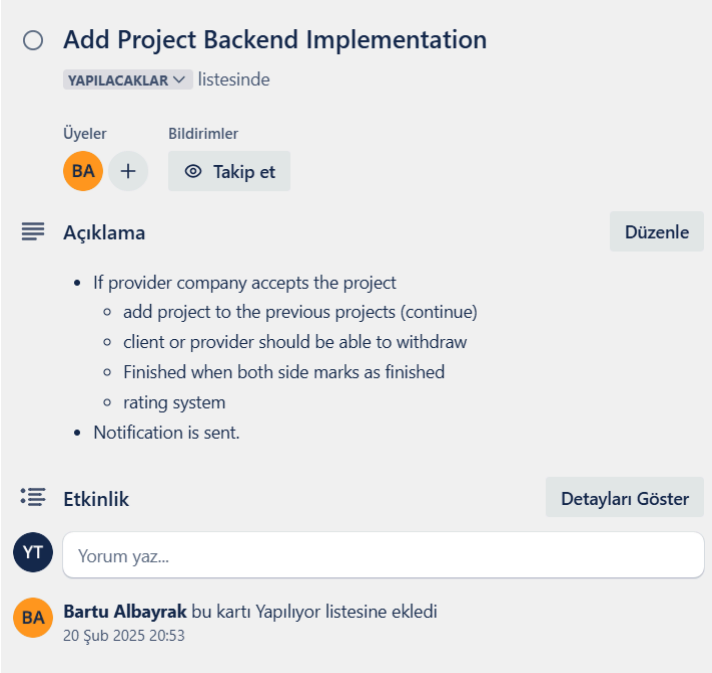

Figure 9: A Trello task structure

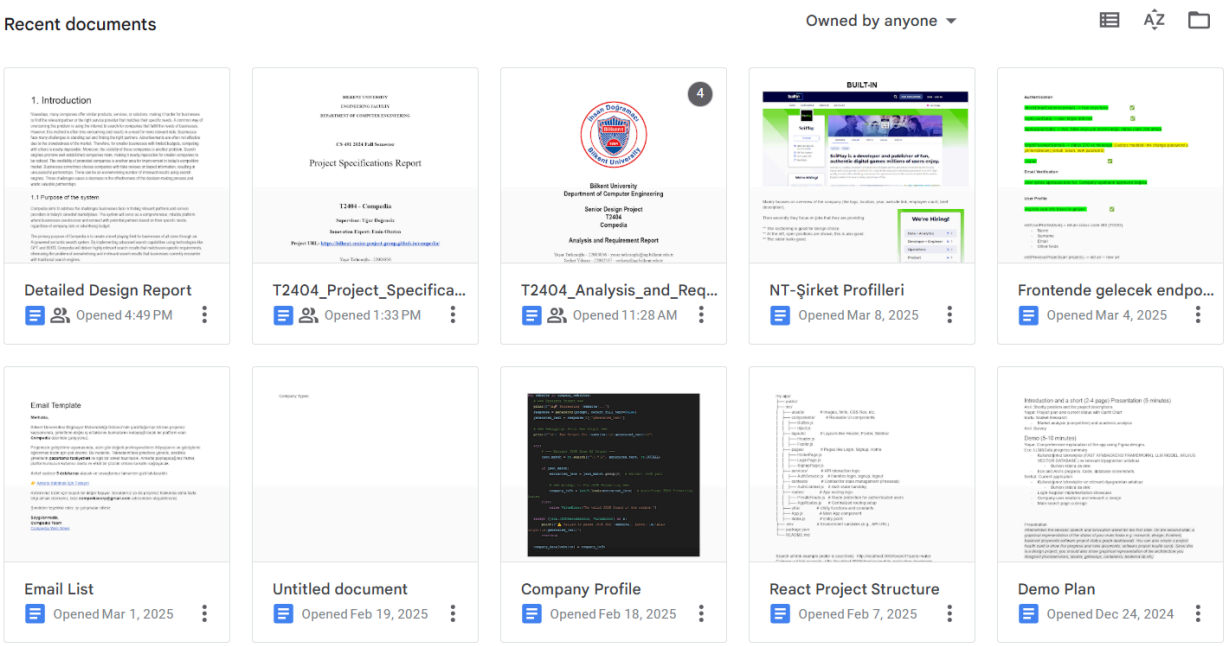

Figure 10: Google Docs collaboration space

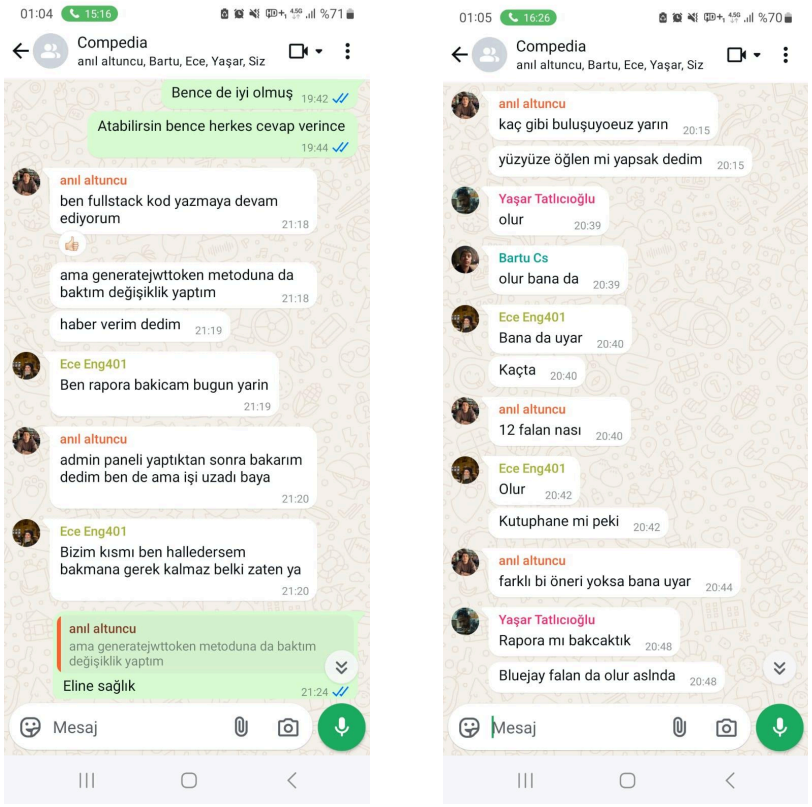

Figure 11: Communication instances through WhatsApp

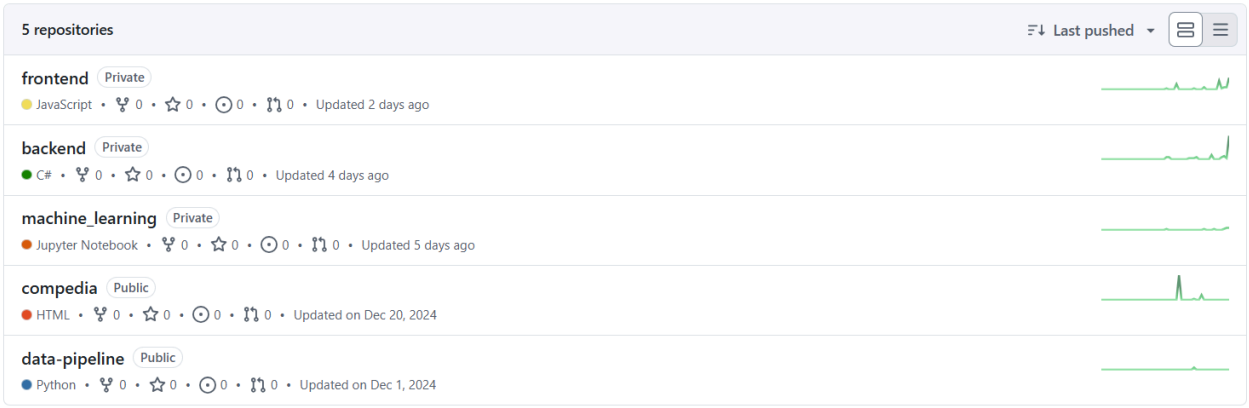

Figure 12: Project repositories and commit frequencies

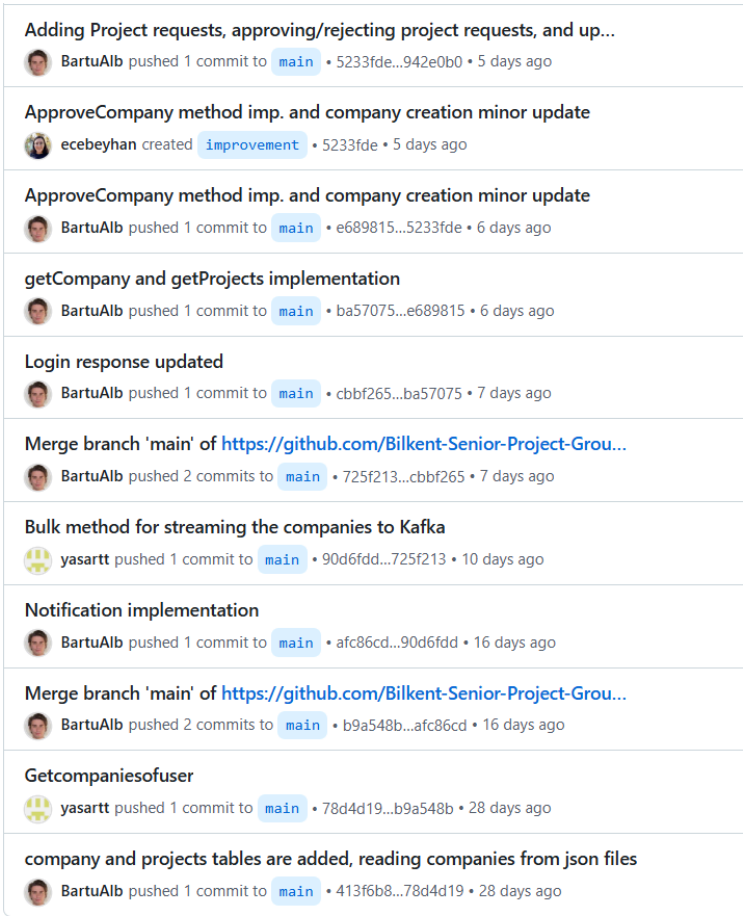

Figure 13: Backend repository commit names

7.2 Helping creating a collaborative and inclusive environment

Fostering a collaborative and inclusive environment is a priority within our team. We achieve this through open communication, transparent task management, and constructive feedback mechanisms.

We use Trello and GitHub for collaboration, ensuring that all tasks are visible and assigned fairly. This transparency encourages active participation from all members. Additionally, we engage in ongoing discussions via WhatsApp and Google Docs, where we make key decisions and develop problem-solving approaches.

To have an inclusive environment, every team member is encouraged by the others to share their perspectives and ideas, particularly during meetings and code reviews. We also ensure that workload distribution considers each member's expertise and availability, allowing for flexibility while being productive.

Our collaborative process is further strengthened through reviews of work done, where team members provide constructive feedback to stay integrated and maintain best practices. This not only enhances our technical work but also creates a culture of shared learning and mutual support.

7.3 Taking lead role and sharing leadership on the team

Leadership on our team is not centralized, but rather distributed among members based on expertise and responsibilities. Each team member takes responsibility for specific aspects of the project and ensures that all critical components receive dedicated attention and leadership.

The leadership responsibilities within the project are as follows:

- **Yaşar Tatlıcıoğlu**: Leads the microservices architecture, vector database implementation, and cloud computing efforts, ensuring robust infrastructure and efficient data storage.
- **Anıl Altuncu**: Leads the implementation of LLM features, prompt engineering, and data management strategies, making informed decisions about AI integration.
- **Serhat Yılmaz**: Takes the lead in frontend development and API integration, ensuring seamless communication between backend and frontend components while prioritizing security.
- **Bartu Albayrak**: Leads backend engineering, MicrosoftSQL implementation, and security measures, contributing to API integration and database efficiency.
- **Ece Beyhan**: Leads UI design and frontend development while contributing to data management strategies to enhance user experience.

8. 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. [2]

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.

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.

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. [3]

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. [4]

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. [5]

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. [6]

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. [7]

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. [8]

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. [9]

Zero-shot Classification
Zero-shot classification models are large, pre-trained models that can classify images without being trained on a particular use case. [10]

9. References

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

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

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

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

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

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

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

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

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

[10] 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.