BILKENT UNIVERSITY

ENGINEERING FACULTY

DEPARTMENT OF COMPUTER ENGINEERING

CS 491 2024 Fall Semester

Project Specifications Report

T2404 - Compedia

Supervisor: Uğur Doğrusöz

Innovation Expert: Emin Okutan

Project URL: https://bilkent-senior-project-group.github.io/compedia/

Yaşar Tatlıcıoğlu - 22003856
Serhat Yılmaz - 22002537
Bartu Albayrak - 22101640
Anıl Altuncu - 21901880
Ece Beyhan - 22003503

Specification Report
November 22, 2024

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

Table of Contents

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 lack of 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 problem 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 Description

Compedia aims to create a platform where companies can generate comprehensive profiles containing highly detailed and structured information, making these profiles easily searchable through natural language queries powered by advanced natural language processing (NLP) technology. The platform will enable businesses to include foundational details such as location, founding year, employee count, and more complex information like offered services, industrial specialties, and past collaborations. These complex data points will be processed and discoverable through state-of-the-art NLP models, including Large Language Models (LLMs) like GPT or BERT.

1.2 High Level System Architecture & Components of Proposed Solution

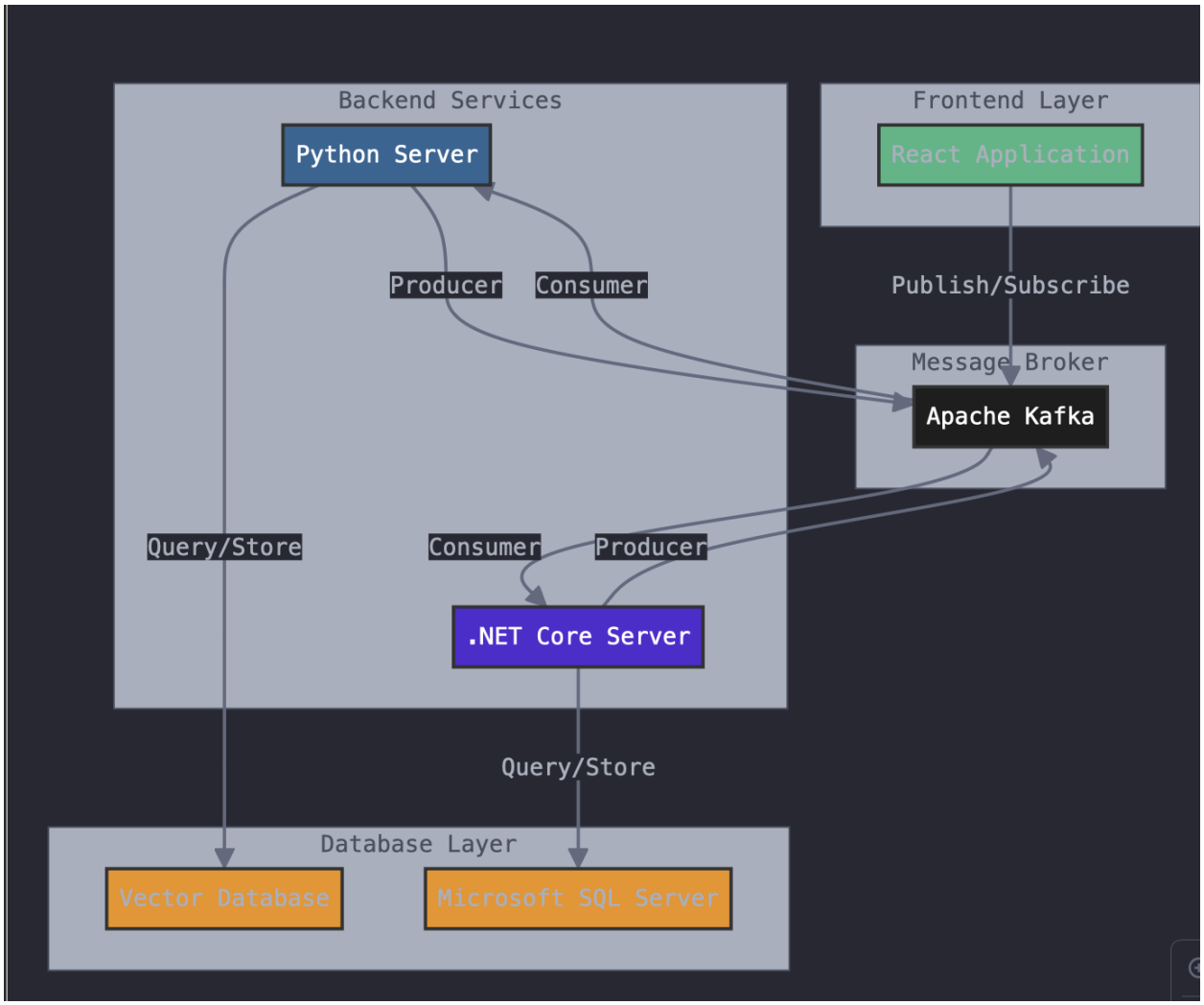

Fig. 1. Compedia High-Level System Architecture

For the frontend, we will use React to build a modular user interface and integrate Kafka for asynchronous communication between the React application and backend services. The application will follow a microservice architecture, using FastAPI (Python) and ASP.NET Core for backend services. FastAPI will handle REST endpoints and vector similarity searches using a vector database like Milvus, while ASP.NET Core will manage structured data with Microsoft SQL Server and provide authentication through .NET Identity. Kafka will serve as the message broker, facilitating seamless communication between microservices. Containerization will be

achieved using Docker, with Docker Compose for local development, and version control will be managed through Git.

1.3 Constraints

Constraints refers to the particular limits and circumstances that affect the Compedia project's growth and functioning. These constraints are categorized into three key areas: Implementation Constraints, Economic Constraints, and Ethical Constraints.

1.3.1. Implementation Constraints

- The Company-Hub project will use React for front-end development to ensure a responsive and user-friendly interface.
- For the backend, .NET Core will be used.
- MicrosoftSQL will serve as the primary relational database for basic company information (e.g., location, employee count, etc.), while 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.
- The project will integrate state-of-the-art LLMs like GPT or BERT for text processing tasks such as zero-shot classification and Named Entity Recognition (NER).
- The platform will be hosted on Azure.
- Development will be managed using Git and hosted on GitHub.
- NET Core Identity will be used to implement user authentication and role-based access control for secure management of platform users and permissions.
- Trello will be used for project task management and team collaboration.
- Axios will be utilized for managing API requests.
- Docker will be used to containerize the back-end services.

1.3.2. Economic Constraints

- The project relies on the use of open-source tools and frameworks, such as React, .NET Core, and MicrosoftSQL, to minimize software licensing costs.

- Hosting on Azure incurs initial setup and ongoing maintenance expenses.
- Utilizing pre-trained Large Language Models (LLMs), such as GPT or BERT, may incur costs for fine-tuning and training.
- Purchasing and maintaining a custom domain for the web platform and associated services will require a yearly or multi-year subscription, depending on the registrar chosen.

1.3.3. Ethical Constraints

- The platform will handle sensitive company data, such as business details, past collaborations, and user-generated content. Data privacy laws and local regulations are mandatory to ensure users' information is stored securely and used ethically.
- The Large Language Models (LLMs) used for natural language processing may inadvertently generate biased or inappropriate responses.
- Semantic search must not unintentionally favor certain types of companies or data.

1.4 Professional and Ethical Issues

- Transparency will be maintained in how user data is collected, stored, and used within the application.
- During the development of machine learning models, their behavior will be closely monitored to ensure they do not display bias or generate inappropriate content.
- Mechanisms such as user verification will be implemented to ensure trust and reliability on the platform.

1.5 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.

2. Design Requirements

2.1. Functional Requirements

2.1.1 User

- The user can register/login with email and password.
- The user can reset their password.
- The user can search and view user profiles.
- The user can search and view company 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 has all permissions such as view, edit, add user and remove permission from other users.
- The root user 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 can modify or remove permissions assigned to other users.

2.1.2 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 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)

2.2. Non-Functional Requirements

2.2.1. 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 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.

2.2.2. 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.

2.2.3. 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.

2.2.4. 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.

2.2.5. 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].

2.2.6 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.

3. Feasibility Discussions

3.1. Market & Competitive Analysis

From the perspective of stakeholders looking for companies, Compedia is a platform to find companies more effectively. The first thing that differentiates Compedia from other platforms that act as a company database is that Compedia offers a more intelligent way to understand the client's requests. The reason is that Compedia aims to include complex service descriptions and specialty categorizations from companies and acts as an Inference Engine to understand the total needs of the clients. Compedia uses LLM models and Semantic Search to recommend the companies that can best match the client's needs.

Some existing solutions to find companies:

- https://builtin.com/
- https://clutch.co/
- https://www.goodfirms.co/
- https://themanifest.com/
- https://www.ycombinator.com/

While these platforms act simply as a database of companies that is searchable with keywords, Compedia acts as an Inference Engine. It interprets a company's abilities, fields, portfolios, and credibility using intelligent data processing methods together with modern LLM models. Having these complex and intelligent descriptions of the companies extracted, our platform applies Semantic Search on the search queries to recommend the best matching companies in Compedia's database.

Another important aspect that differentiates Compedia from other platforms is that it sees the provider companies as the main stakeholders in its business plan. Compedia, as a platform, is built in order to respond to the provider companies' needs as much as it can. We aim to be a platform where providers can promote their business, increase their visibility, and benefit from Compedia to stay active in the markets in which they aim to find clients and make a profit. That purpose differentiates Compedia from other platforms as a business model.

3.2. Academic Analysis

Compedia's functionality relies heavily on state-of-the-art Natural Language Processing (NLP) techniques and Large Language Models (LLM). These techniques are planned to combine with the vector database and semantic search in order to provide customers with more efficient company search.

NLP is used to enhance the interaction between computers and humans in terms of the language. It aims to understand, read and decipher the human language. The complexities of human language are understood by NLP such as understanding context, language structure. The patterns in data are captured by NLP algorithms and unstructured data is turned into a format that can be understood by computers. Like NLP, LLMs are machine learning techniques that are used to understand human language and generate human-like text. The likelihood of a word appearing in the sentence is calculated according to the presence of words that come before it. Thus, computers can generate sentences and coherent data. NLP models constitute the basis of LLMs. LLMs are fed with a great amount of data so that they can learn the linguistic patterns. They are used in text generation required fields [2]. LLM will ease processing the complex queries and extracting the relevant results in Compedia.

Basic lexical matching is sometimes not enough for better recommendations. When searching a query in the search engine, the pure lexical matching will probably give lower efficient results compared to a matching according to the meaning. In that sense, semantic search is used to have better results thanks to the matching based on the meaning of the query. In semantic search, word embeddings are used and corpuses are embedded into a vector space. While dealing with a vast amount of data, instead of using vector space, vector databases are used to maintain efficiency. In that sense, semantic search and vector databases can be used together [3]. In terms of Compedia, vector databases are planned to be used in order to increase the scalability and efficiency of search operations. Also, embedding techniques are planned to be used in Compedia to represent company profiles and user queries which enables similarity computations for search functionality in the application.

4. Glossary

1. **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. [4]

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.

3. **BERT (Bidirectional Encoder Representations from Transformers)**
BERT language model is an open source machine learning framework for natural language processing [5].

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.

5. **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. [6]

6. **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. [7]

7. **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. [8]

8. **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. [9]

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

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

11. 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. [10]

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

13. Semantic Search
Semantic search is a search engine technology that interprets the meaning of words and phrases. [11]

14. 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. [12]

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

16. 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. [13]

17. Zero-shot Classification
Zero-shot classification models are large, pre-trained models that can classify images without being trained on a particular use case. [14]

5. References

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

[2] Kolena. "LLM vs. NLP: 6 Key Differences and Using Them Together." *Kolena*, https://www.kolena.com/guides/llm-vs-nlp-6-key-differences-and-using-them-together. Accessed 21 Nov. 2024.

[3] "Semantic Search with Vector Databases: An Overview." *KDnuggets*, https://www.kdnuggets.com/semantic-search-with-vector-databases. Accessed 21 Nov. 2024.

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

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

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

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

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

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

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

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

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

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

[14] 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.