



Software Architecture Description

afetbilgi.com

Group 1

Emirhan Oğul 2580777
Mehmet Rüçhan Yavuzdemir 2522159

31.05.2023

Table of Contents

1	Introduction	1
1.1	Purpose and objectives of afetbilgi.com	1
1.2	Scope	1
1.3	Stakeholders and their concerns	1
2	References	2
3	Glossary	3
4	Architectural Views	4
4.1	Context View	4
4.1.1	Stakeholders' uses of this view	4
4.1.2	Context Diagram	4
4.1.3	External Interfaces	6
4.1.4	Interaction scenarios	8
4.2	Functional View	9
4.2.1	Stakeholders' uses of this view	9
4.2.2	Component Diagram	10
4.2.3	Internal Interfaces	11
4.2.4	Interaction Patterns	13
4.3	Information View	15
4.3.1	Stakeholders' uses of this view	15
4.3.2	Database Class Diagram	16
4.3.3	Operations on Data	16
4.4	Deployment View	19
4.4.1	Stakeholders' uses of this view	19
4.4.2	Deployment Diagram	19
4.5	Design Rationale	20
4.5.1	Context View	20
4.5.2	Functional View	20
4.5.3	Information View	20
4.5.4	Deployment View	20
5	Architectural Views for Suggestions to Improve the Existing System	21
5.1	Context View	21
5.1.1	Stakeholders' uses of this view	21
5.1.2	Context Diagram	21
5.1.3	External Interfaces	22

5.1.4	Interaction scenarios	24
5.2	Functional View	24
5.2.1	Stakeholders' uses of this view	24
5.2.2	Component Diagram	25
5.2.3	Internal Interfaces	26
5.2.4	Interaction Patterns	28
5.3	Information View	28
5.3.1	Stakeholders' uses of this view	28
5.3.2	Database Class Diagram	29
5.3.3	Operations on Data	29
5.4	Deployment View	31
5.4.1	Stakeholders' uses of this view	31
5.4.2	Deployment Diagram	32
5.5	Design Rationale	33
5.5.1	Context View	33
5.5.2	Functional View	33
5.5.3	Information View	33
5.5.4	Deployment View	33

List of Figures

1	Context Diagram	4
2	External Interfaces Diagram	6
3	View Information Details Activity Diagram	8
4	Add Verified Information Activity Diagram	9
5	Component Diagram	10
6	Internal Interfaces Diagram	11
7	Serialize Sequence Diagram	13
8	Convert Bytes To PDF Sequence Diagram	14
9	Convert Bytes To Map Location Sequence Diagram	15
10	Database Class Diagram	16
11	Deployment Diagram	19
12	Improved Context Diagram	21
13	Improved External Interfaces Diagram	22
14	Translation API Activity Diagram	24
15	Improved Component Diagram	25
16	Improved Interface Class Diagram	26
17	Translation Sequence Diagram	28
18	Improved Database Class Diagram	29
19	Improved Deployment Diagram	32

List of Tables

1	Glossary	3
2	External Interface Operation Descriptions	7
3	Internal Interface Operation Descriptions	12
4	CRUD Operations	18
5	Improved External Interface Operation Descriptions	23
6	Improved Internal Interface Operation Descriptions	27
7	Improved CRUD Operations	31

Chapter 1

Introduction

1.1 Purpose and objectives of afetbilgi.com

The purpose of this system is to provide reliable and validated information in the fight against the Pazarcık Earthquake on 6 February 2023, to both disaster victims and those who want to help. It aims to prevent the spread of misinformation thanks to its completely human-made database which is accurate, user-friendly, fast, and simple.

1.2 Scope

The main scope of afetbilgi.com is that view reliable information split into various interfaces like topic selection interface, information list, Google Maps, and PDFs.

1.3 Stakeholders and their concerns

There are 5 main stakeholders of afetbilgi.com, which are suppliers, admins, users, government, and developers.

- Suppliers are the companies or organizations that want to help people in various ways, such as temporary accommodation places, food distribution centers, gas stations, mobile toilets, and so on. Their concern is receiving up-to-date information, i.e., providing support to a place that already has not been supported.
- Admins are the people who organize and update the information that is present in the current human-made database. Their concern is automating the tasks and receiving less manual work.
- Users can be divided into two categories:
 - Users who try to help others, such as doctors, winch operators, veterinarians, and so on. Their concern is a website that enables them to help people in need.
 - Users who try to find help on the site, i.e., people who were affected by the earthquake. Their concern is a website that acts as a means of help center.
- Government organizations such as AFAD are making use of this website for finding missing people, reported by trusted sources. Their concern is the legal issues of the website.
- Developers are highly active in GitHub, they give full support to the open source project on GitHub. In the early stages, they work almost full-time for the website. Their concern is developing the website in the fastest and the most efficient way.

Chapter 2

References

This document is written with respect to IEEE 42010 standard, using the source below: "ISO/IEC/IEEE Systems and software engineering – Architecture description," in ISO/IEC/IEEE 42010:2022(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000), vol., no., pp.1-46, 1 Dec. 2011, doi: 10.1109/IEEESTD.2011.6129467

Chapter 3

Glossary

Term	Definition
API	Application Programming Interface
CDN	Content Delivery Network
AWS	Amazon Web Services
PDF	Portable Document Format
HTTPS	Hypertext Transfer Protocol Secure
TCP	Transmission Control Protocol

Table 1: Glossary

Chapter 4

Architectural Views

4.1 Context View

4.1.1 Stakeholders' uses of this view

There are 5 main stakeholders of afetbilgi.com, which are suppliers, admins, users, government, and developers. In this viewpoint, quite a few materials are used to explain the context of the system, such as, a context diagram, an external interfaces diagram, and an interaction scenarios diagram. Stakeholders may make use of this viewpoint in order to understand different perspectives and interactions between the website and any kind of users, such as, earthquake victims, people who try to help and organizations who try to supply goods to people in need. Furthermore, the impact of the website on earthquake victims and how external entities are used will be discussed from a context view.

4.1.2 Context Diagram

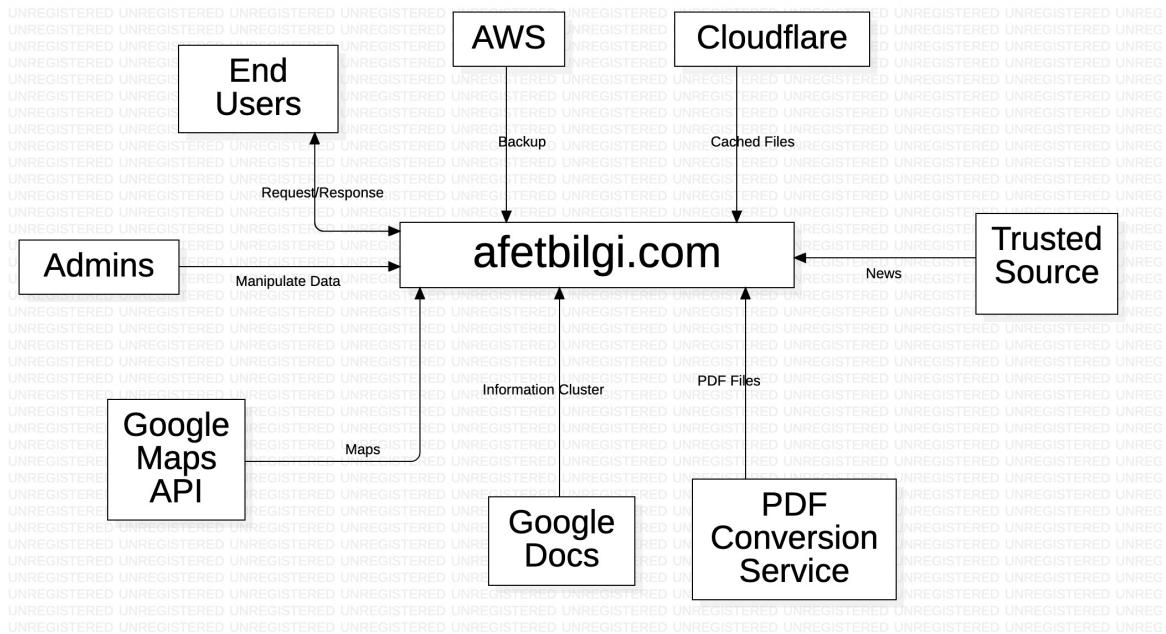


Figure 1: Context Diagram

- **Admins:** They approve/disapprove the data coming from trusted sources.
- **End Users:** They have interactions in both ways, either they can submit information or request trusted news from the human-made database.
- **AWS:** This service provides an update and backup functionality for CDN and database.
- **Cloudflare:** It provides content delivery networks for static data of the underlying services such as CDN, Google Maps API, and PDF Conversion Interface with the help of caching.
- **Trusted Source:** They are the sources trusted by the majority of the population, and they provide reliable news to the system.
- **PDF Conversion Service:** This API is used to convert the selected address' information list into a PDF document which is easy to download and share. The created PDF will include all the information about the selected address.
- **Google Docs:** This API keeps track of the relevant details of the general needs, important resources, and health services, such as phone numbers, addresses, and organization names in a structured, reachable, and easy-to-understand way.
- **Google Maps API:** This API is used to search and filter useful locations such as accommodation, food distribution centers, active hospitals, veterinarians, gas stations, evacuation points, and emergency gathering places in Turkey's map. The users should consent to share their location with the API.

4.1.3 External Interfaces

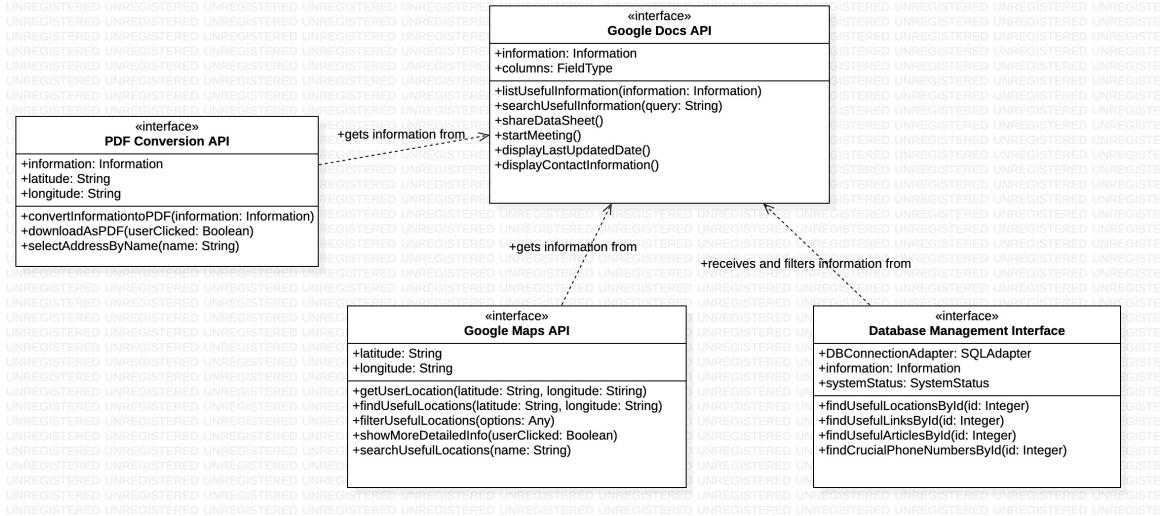


Figure 2: External Interfaces Diagram

afetbilgi.com has several external interfaces, such as, PDF Conversion API, Google Docs API, Google Maps API, and Database Management Interface. The operations given in the diagram can be summarized as:

- An information from a source is entered to the Google Docs API, and then it will be marked as trusted or non-trusted by the Database Management Interface and will be sent back to the Google Docs API.
- PDF Conversion API will use the information coming from Google Docs API to create a PDF containing all the appropriate information.
- Google Maps API will use the information coming from Google Docs API to create a fully functional map which can be navigated through freely and has functionalities like filtering, searching, zooming in, zooming out.

Operation	Description
convertInformationtoPDF	Takes information as a parameter, and it converts it to PDF.
downloadAsPDF	Triggered by the user clicking on the website button, and it downloads the PDF.
selectAddressByName	Takes a string parameter drop-down menu, queries over the human-made database, and returns the related information.
listUsefulInformation	Takes information as a parameter, and it returns a useful information table.
searchUsefulInformation	Takes a string parameter from drop-down menu, and gets the useful information regarding that address.
shareDataSheet	Generates a link for the share purposes, anyone with that link can access to the document if they have permission.
startMeeting	Starts a Google Meet in a short and practical way.
displayLastUpdatedDate	Displays the last updated date of the document.
displayContactInformation	Displays the contact information of the trusted sources.
getUserLocation	Gets user's current location if they have granted consent.
findUsefulLocations	Finds all the useful locations given the user input.
filterUsefulLocations	Filters all the useful locations based on the filters user has provided.
showMoreDetailedInfo	Redirects user to the more detailed version of the Google Maps Interface.
searchUsefulLocations	Searches for the useful locations based on the text user has provided.
findUsefulLocationsById	Returns the useful locations from the database, by their ID.
findUsefulLinksById	Returns the useful links from the database, by their ID.
findUsefulArticlesById	Returns the useful articles from the database, by their ID.
findCrucialPhoneNumbersById	Returns the crucial phone numbers from the database, by their ID.

Table 2: External Interface Operation Descriptions

4.1.4 Interaction scenarios

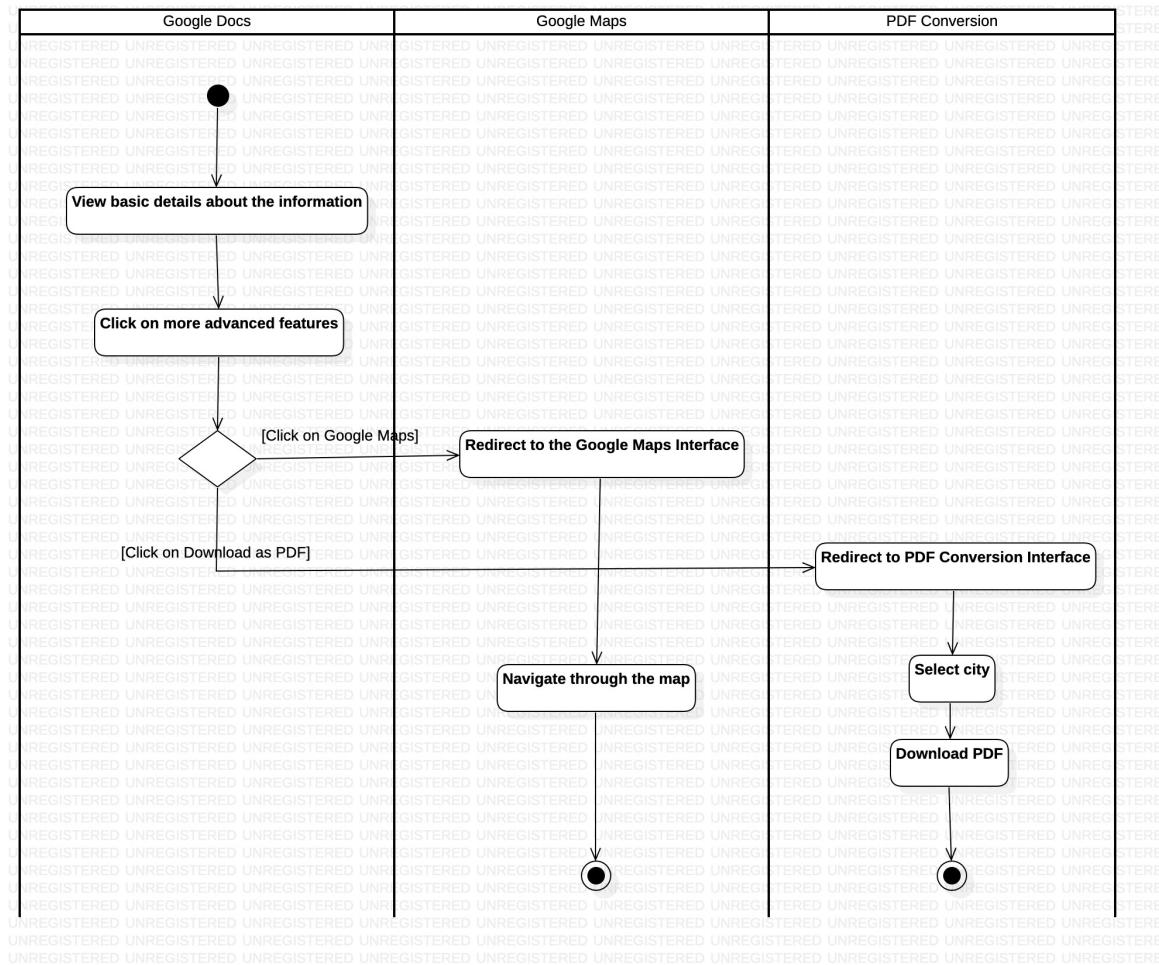


Figure 3: View Information Details Activity Diagram

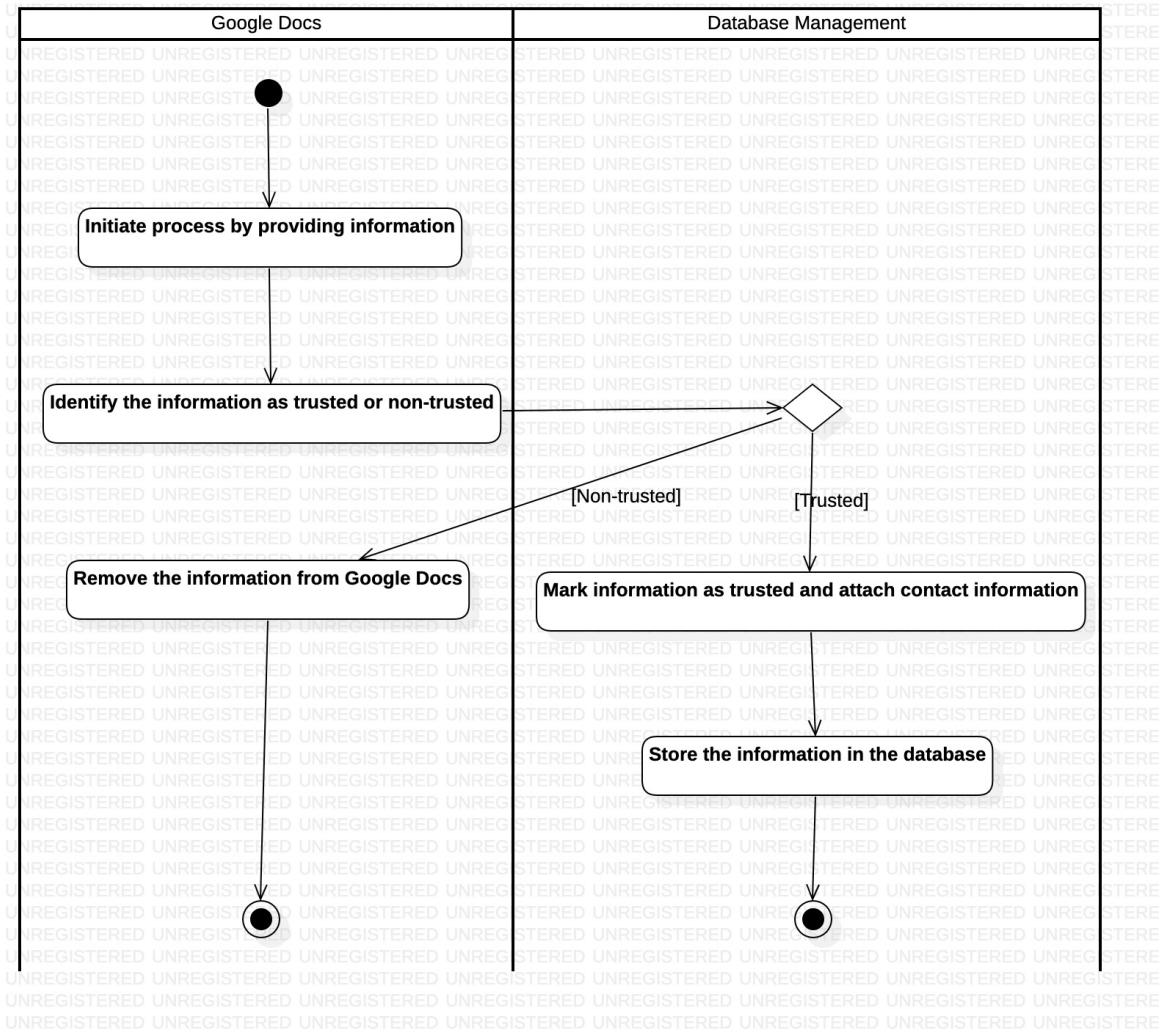


Figure 4: Add Verified Information Activity Diagram

4.2 Functional View

4.2.1 Stakeholders' uses of this view

There are 5 main stakeholders of afetbilgi.com, which are suppliers, admins, users, government, and developers. Generally, users, the government, and the suppliers will use this view to grasp the limits and capabilities of the website. The developers and admins may use this view to understand the functional design implemented in the website. The government may use this view to understand the legal issues of the website.

4.2.2 Component Diagram

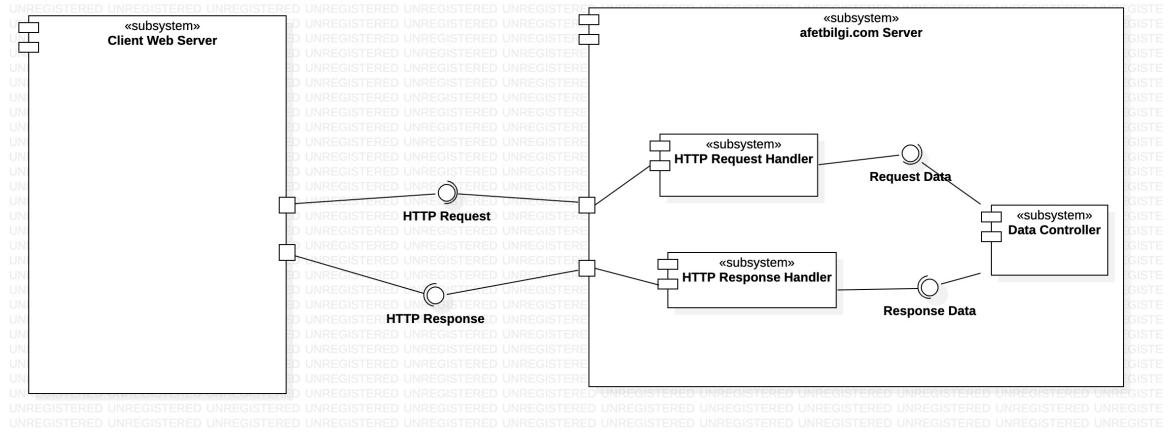


Figure 5: Component Diagram

- There are 5 subsystems in afetbilgi.com's infrastructure, namely Client Web Server, afetbilgi.com Server, HTTP Request Handler, HTTP Response Handler, and Data Controller.
- afetbilgi.com Server consists of an HTTP Request Handler, HTTP Response Handler, and Data Controller.
- afetbilgi.com Server has a Data Controller which lets the server handle the information traffic between the database and the server.
- afetbilgi.com provides an HTTP Response interface to the Client Web Server, and it requires an HTTP Request interface from the Client Web Server.
- Client Web Server acts as a communicator between the client (earthquake victims) and the afetbilgi.com servers by sending HTTP requests. It provides an HTTP Request interface and requires an HTTP Response interface.
- Data Controller provides a Response Data interface that returns the appropriate information to the HTTP Response Handler.
- Data Controller requires a Request Data interface which passes an HTTP Request to the Data Controller specifying the requirements of the Client Web Server.
- HTTP Request Handler provides a Request Data interface, and HTTP Response Handler requires a Response Data interface.

4.2.3 Internal Interfaces

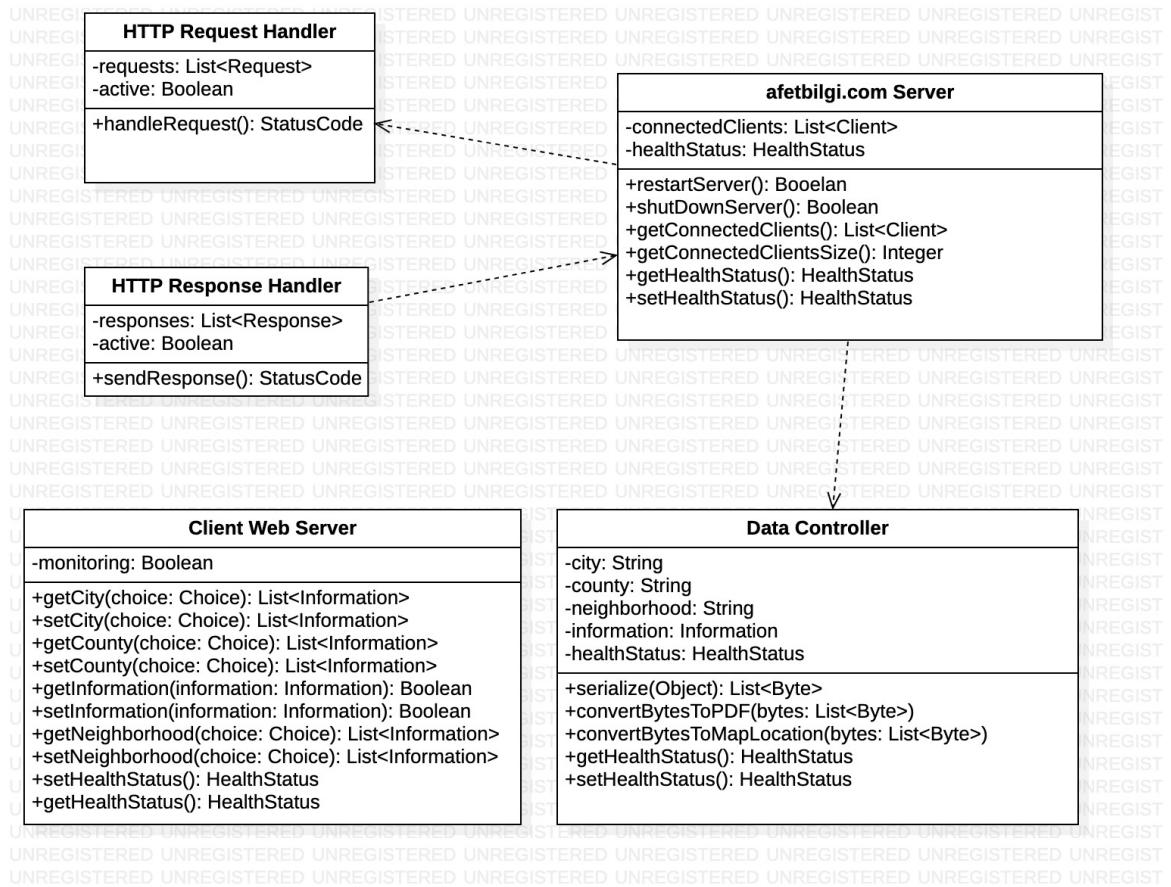


Figure 6: Internal Interfaces Diagram

Operation	Description
handleRequest	Handles requests based on given headers and body.
sendResponse	Creates a response and sends it to the receiver.
restartServer	Restarts afetbilgi.com with internal configurations.
shutDownServer	Shuts down the server.
getConnectedClients	Returns the list of connected clients.

getConnectedClientsSize	Returns the number of connected clients.
afetbilgi::getHealthStatus	Returns the health status of afetbilgi.com servers.
afetbilgi::setHealthStatus	Sets the health status of afetbilgi.com servers.
getCity	Returns the city that the user has provided.
setCity	Sets the city according to the user's choices.
getCounty	Returns the county that the user has provided.
setCounty	Sets the county based on the user's choices.
getInformation	Returns the information from the database based on user choices.
setInformation	Adds information to the database based on the information trusted sources have provided.
getNeighborhood	Returns the neighborhood that the user has chosen.
setNeighborhood	Sets the neighborhood based on user's choices.
ClientWebServer::setHealthStatus	Sets the health status of Client Web Server
ClientWebServer::getHealthStatus	Returns the health status of Client Web Server
serialize	Serializes the given object into bytes and returns that byte array.
convertBytesToPDF	Converts the bytes present on the Data Controller to the desired PDF output.
convertBytesToMapLocation	Converts the bytes present on the Data Controller to the desired map location.
DataController::getHealthStatus	Returns the health status of Data Controller.
DataController::setHealthStatus	Sets the health status of the Data Controller.

Table 3: Internal Interface Operation Descriptions

4.2.4 Interaction Patterns

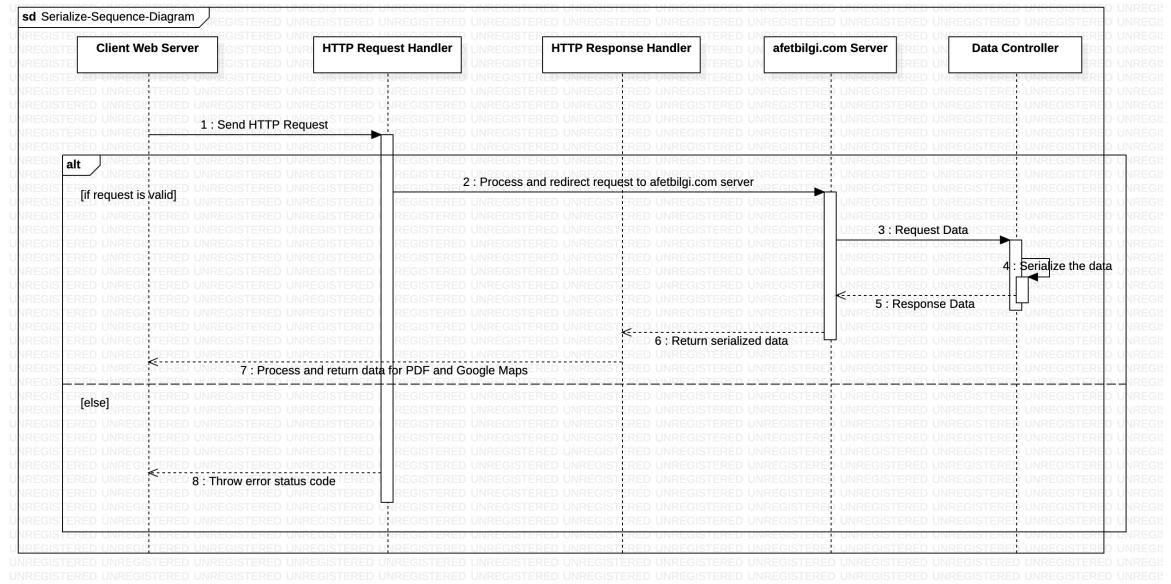


Figure 7: Serialize Sequence Diagram

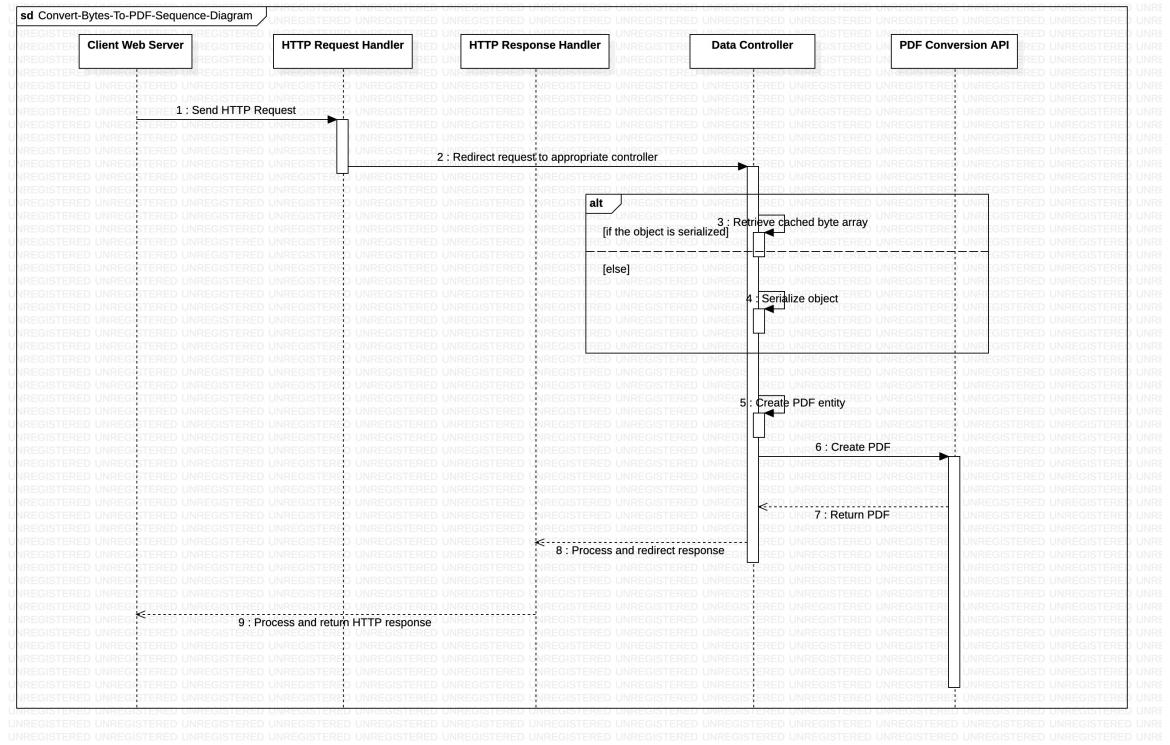


Figure 8: Convert Bytes To PDF Sequence Diagram

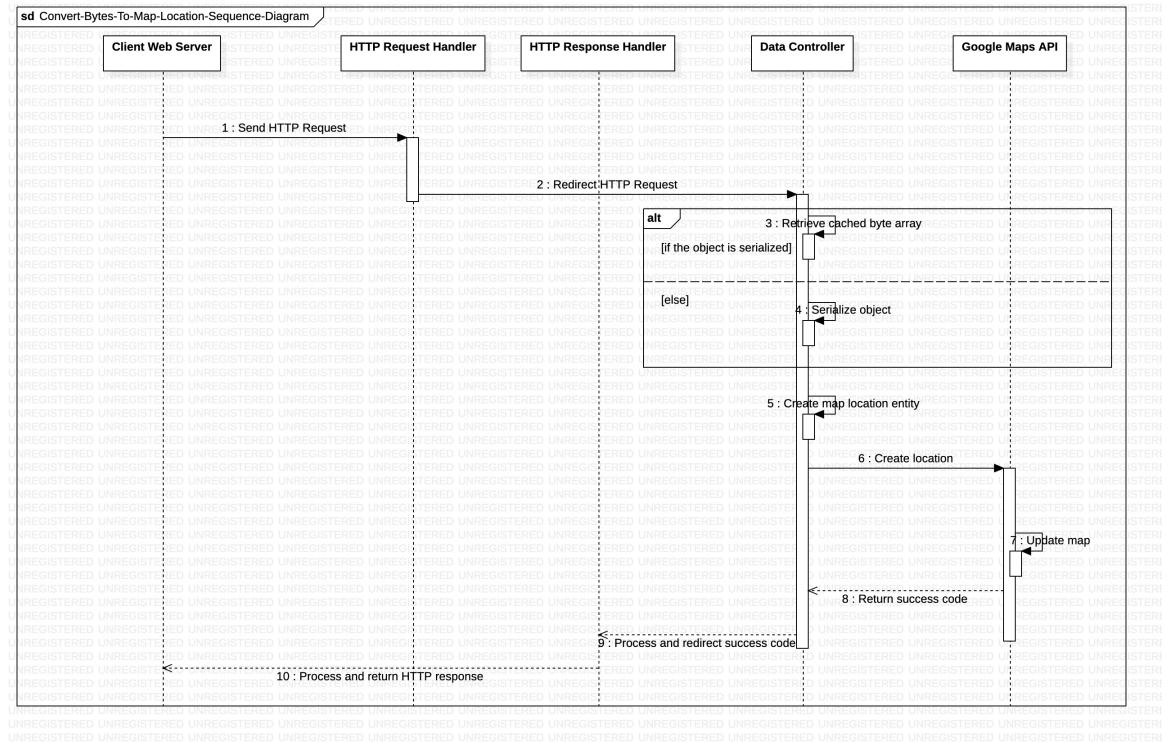


Figure 9: Convert Bytes To Map Location Sequence Diagram

4.3 Information View

4.3.1 Stakeholders' uses of this view

There are 5 main stakeholders of afetbilgi.com, which are suppliers, admins, users, government, and developers. The government uses this view to understand how the data is flowing throughout the program and if it's legal or not. Admins and developers use this view to understand the strategies used within the process of data transfer, data modeling, and structuring of the data. Users and suppliers use this view to understand how to contribute information to the system or take information from the system.

4.3.2 Database Class Diagram

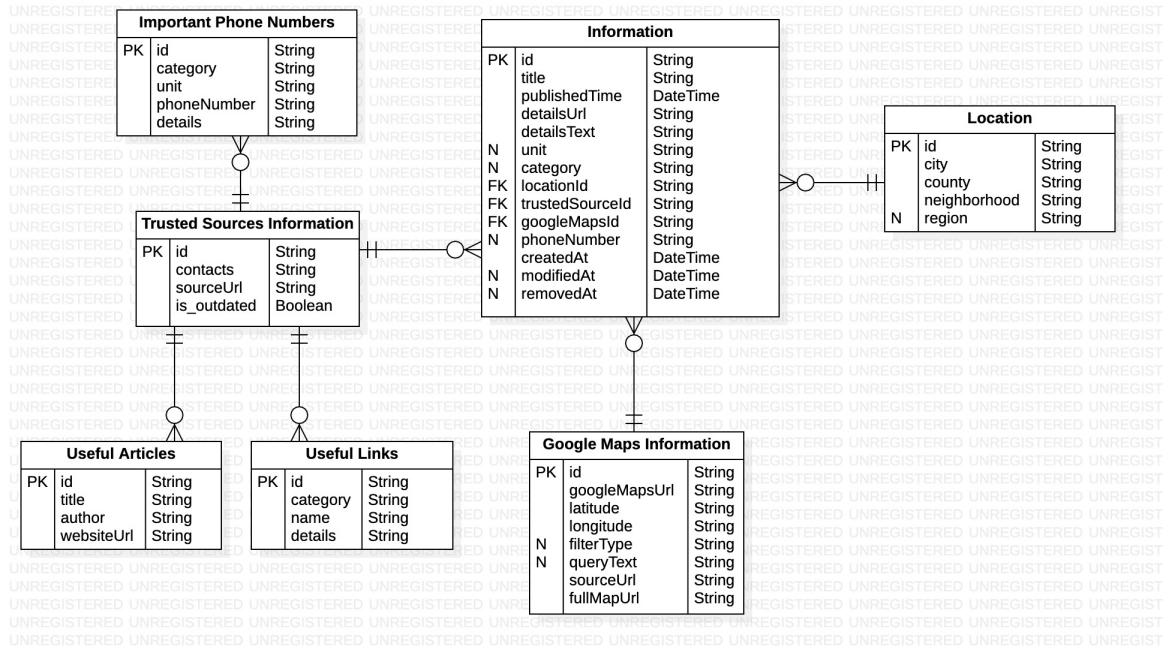


Figure 10: Database Class Diagram

4.3.3 Operations on Data

Operation	Description
isOutdated	Create: Read: Information Update: Delete:
addInformation	Create: Information Read: Update: Delete:

getInformation	Create: Read: Information Update: Delete:
updateInformation	Create: Read: Update: Information Delete:
deleteInformation	Create: Read: Update: Delete: Information
getCity	Create: Read: City Update: Delete:
setCity	Create: Read: Update: City Delete:
getCounty	Create: Read: County Update: Delete:
setCounty	Create: Read Update: County Delete:
getNeighborhood	Create: Read: Neighborhood Update: Delete:
setNeighborhood	Create: Read: Update: Neighborhood Delete:

getRegion	Create: Read: Region Update: Delete:
setRegion	Create: Read: Update: Region Delete:
getImportantPhoneNumbers	Create: Read: ImportantPhoneNumbers Update: Delete:
setImportantPhoneNumbers	Create: Read: Update: ImportantPhoneNumbers Delete:
getUsefulArticles	Create: Read: UsefulArticles Update: Delete:
setUsefulArticles	Create: Read: Update: UsefulArticles Delete:
getUsefulLinks	Create: Read: UsefulLinks Update: Delete:
setUsefulLinks	Create: Read: Update: UsefulLinks Delete:

Table 4: CRUD Operations

4.4 Deployment View

4.4.1 Stakeholders' uses of this view

There are 5 main stakeholders of afetbilgi.com, which are suppliers, admins, users, government, and developers. The government uses this view to understand how the system is deployed and if it's legal or not. Developers and admins use this view to understand the technologies and strategies used within the deployment process. Suppliers and users use this view to understand the capabilities of the system.

4.4.2 Deployment Diagram

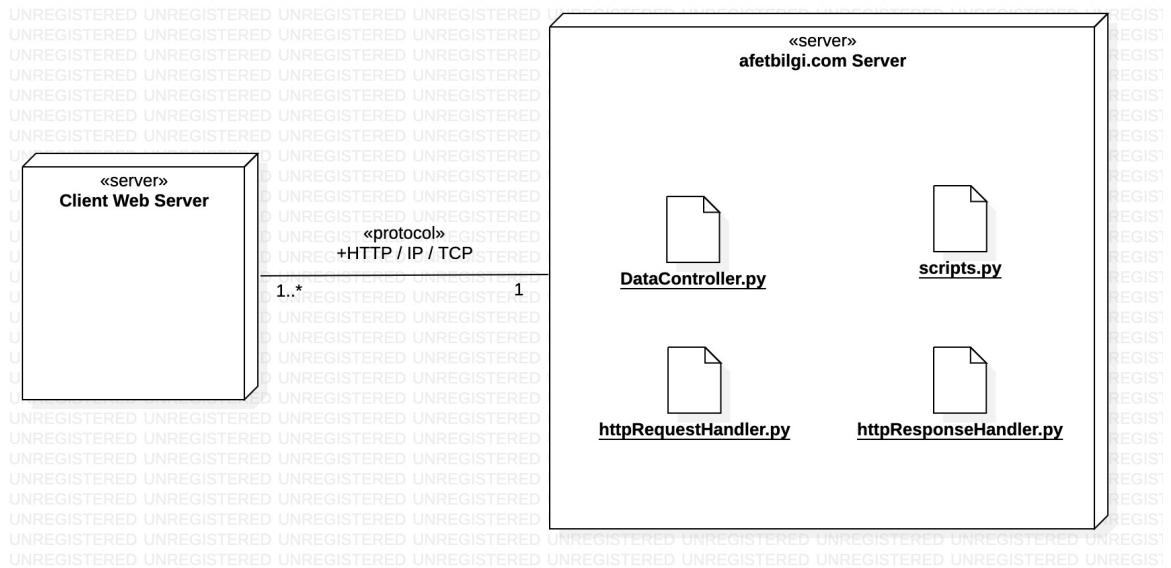


Figure 11: Deployment Diagram

- Client Web Server and afetbilgi.com Server communicate with the protocols HTTP, IP, and TCP.
- All the scripts, handlers, and controllers are written in Python.
- DataController.py provides all the business logic to handle information flow to the user from the server or vice versa.
- HTTPRequestHandler.py provides the implementation of the HTTP interface with the query parameters, path variables, request headers, and request body of the HTTP Request.
- HTTPResponseHandler.py provides the implementation of the HTTP interface for handling the responses.
- scripts.py provides various database scripts and scripts for small tasks.

4.5 Design Rationale

4.5.1 Context View

The rationale behind the context view is to provide an overall understanding of the entire system with its external interfaces, interactions and its impact as a whole. This view mainly focuses on requirements analysis, interactions with stakeholders, identification of external dependencies, and system scope. Also, the impact of the system on earthquake victims is being discussed in this section.

4.5.2 Functional View

The rationale behind the functional view is to illustrate the functioning components of the system and the interactions between the components. This view mainly focuses on high-level functionality of the system for stakeholders to comprehend how the system is performing the tasks they wanted to be done. It also touches on scalability, modularization, requirements validation and verification concerns of the system.

4.5.3 Information View

The rationale behind the information view is to depict how the information is processed throughout the entire program flow. This view mainly focuses on data, more specifically, the structure of the data, the interactions between data, the transformation process of the data, and how the data is persisted. It also touches on consistency, interoperability, integration, security, and privacy concerns of the system.

4.5.4 Deployment View

The rationale behind the deployment view is to provide a comprehensive overview of how the website will be deployed to its target environment. This view mainly focuses on the physical infrastructure of the system and explains how the modules, submodules, and components are distributed to make up the whole system. It also touches on scalability, performance, availability, fault tolerance, communication between components, and resource management factors of the system.

Chapter 5

Architectural Views for Suggestions to Improve the Existing System

5.1 Context View

5.1.1 Stakeholders' uses of this view

There are 5 main stakeholders of afetbilgi.com, which are suppliers, admins, users, government, and developers. In this viewpoint, quite a few materials are used to explain the context of the system, such as, a context diagram, an external interfaces diagram, and an interaction scenarios diagram. Stakeholders may make use of this viewpoint in order to understand different perspectives and interactions between the website and any kind of users, such as, earthquake victims, people who try to help and organizations who try to supply goods to people in need. Furthermore, the impact of the website on earthquake victims and how external entities are used will be discussed from a context view.

5.1.2 Context Diagram

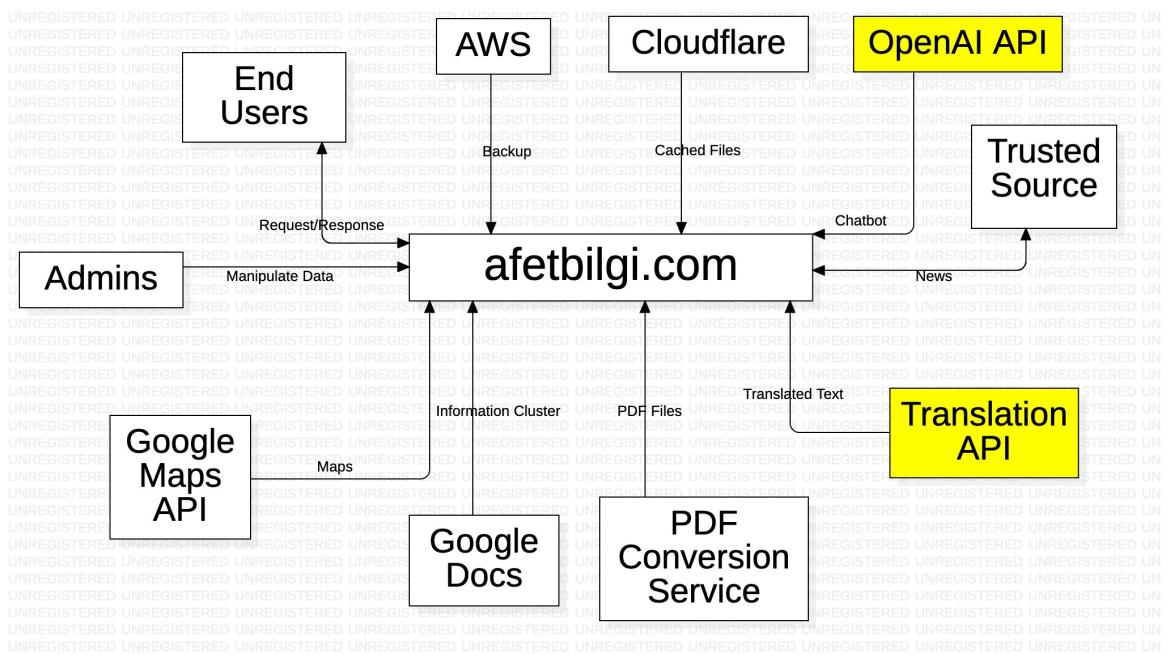


Figure 12: Improved Context Diagram

- **OpenAI API:** The system needs some artificial assistance for users to make their life easier and help them find what they are looking for. OpenAI is known for its very complex and high-performance servers and models. Therefore, there is no need to reinvent the wheel. We can trust and rely on this API. Searching all through the website and trying to find what they need is very cumbersome for especially the people in rural areas, elders, children, and so on. Chatbot powered by OpenAI solves this problem efficiently.
- **Translation API:** Users can see all the Information List Interfaces, Google Maps Interfaces, Google Docs Interfaces, Topic Selection Interfaces and finally generated PDFs in their language thanks to the Translation API. It handles all the complex text and generates the response very quickly. However, in the old version, everything was not integrated into the translation, and they were several interfaces that does not have language support. Even if we select a specific language, they are a lot of text in Turkish, which is not convenient for foreign users. For example, maps.afetbilgi.com does not have any language support. Translation API solves this.

5.1.3 External Interfaces

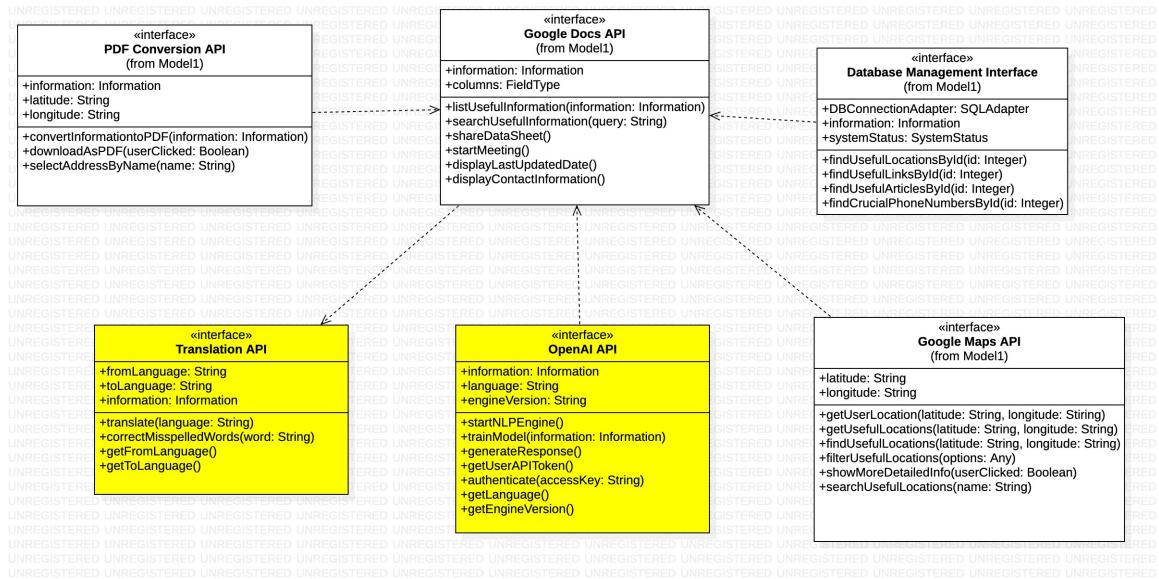


Figure 13: Improved External Interfaces Diagram

Operation	Description
translate	Takes language as a parameter, and it converts the text to the desired language.
correctMisspelledWords	Corrects misspelled words.
getFromLanguage	Returns the language that is being translated.
getToLanguage	Returns the language that is being translated into.
startNLPEngine	Starts the NLP engine.
trainModel	Trains the model based on given inputs and configuration.
generateResponse	Generates the response based on given inputs and configurations.
getUserAPIToken	Returns current user's OpenAI API Token.
authenticate	Authenticates the users based on their OpenAI API Token.
getLanguage	Gets the current language that the user chooses to communicate with the AI chatbot.
getEngineVersion	Gets the engine's current version.

Table 5: Improved External Interface Operation Descriptions

5.1.4 Interaction scenarios

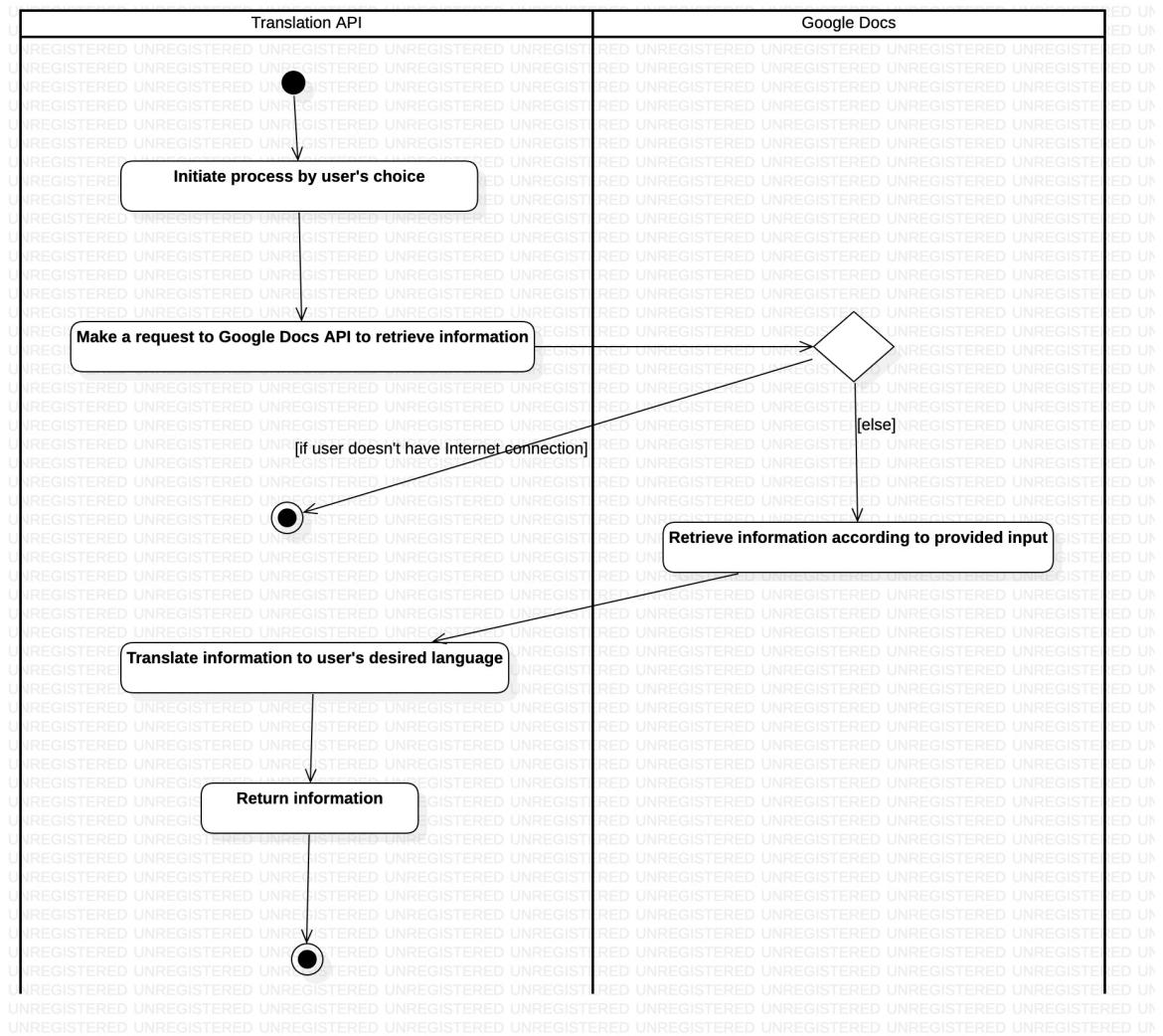


Figure 14: Translation API Activity Diagram

5.2 Functional View

5.2.1 Stakeholders' uses of this view

There are 5 main stakeholders of afetbilgi.com, which are suppliers, admins, users, government, and developers. Generally, users, the government, and the suppliers will use this view to grasp the limits and capabilities of the website. The developers and admins may use this view to understand

the functional design implemented in the website. The government may use this view to understand the legal issues of the website.

5.2.2 Component Diagram

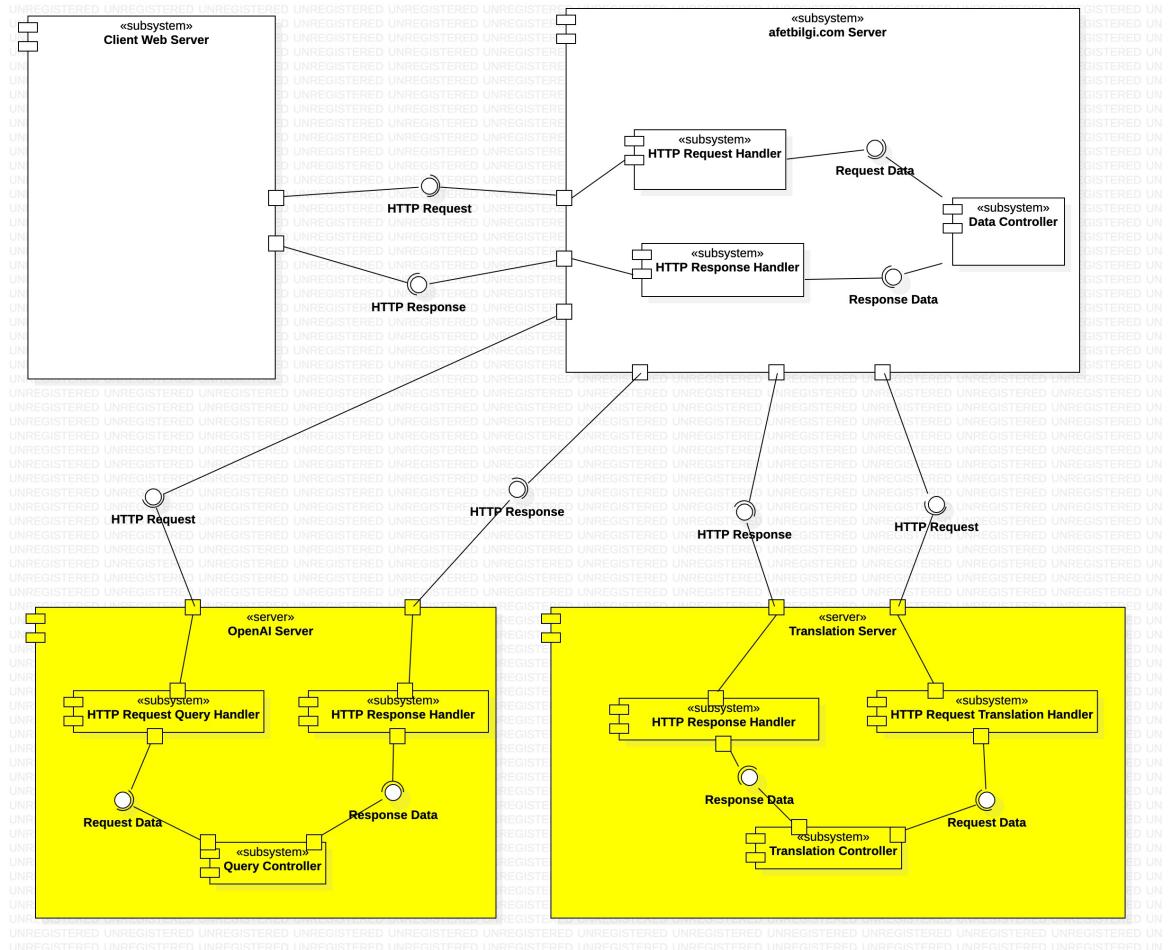


Figure 15: Improved Component Diagram

- afetbilgi.com ecosystem consists of two new components, namely OpenAI Server and Translation Server.
- OpenAI Server requires an HTTP Request interface from the afetbilgi.com and it provides an HTTP Response interface to the afetbilgi.com Server.
- Translation Server requires an HTTP Request interface from the afetbilgi.com and it provides an HTTP Response interface to the afetbilgi.com Server.

- HTTP Request Query Handler provides Request Data interface to Query Controller, and Query Controller provides Response Data interface to HTTP Request Handler.
- Query Controller subsystem handles the chatbot actions based on given user input and configurations.
- HTTP Request Translation Handler provides Request Data interface to Translation Controller, and Translation Controller provides Response Data interface to HTTP Request Handler.
- Translation Controller subsystem handles the translation service based on given user input and configurations.

5.2.3 Internal Interfaces

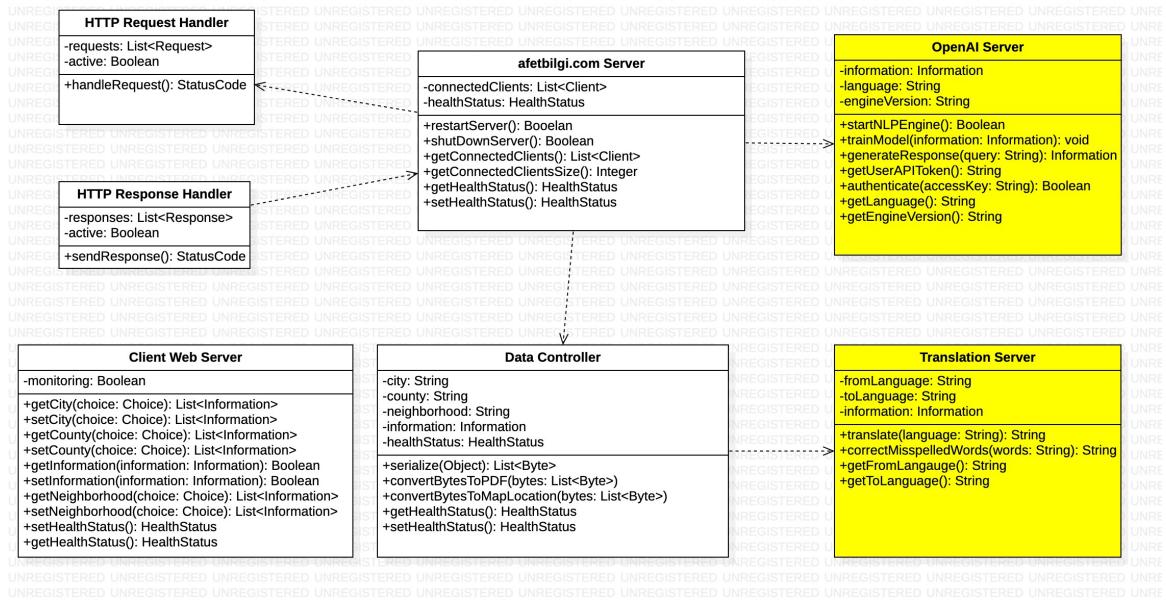


Figure 16: Improved Interface Class Diagram

Operation	Description
translate	Takes language as a parameter, and it converts the text to the desired language.
correctMisspelledWords	Corrects misspelled words.
getFromLanguage	Returns the language that is being translated.
getToLanguage	Returns the language that is being translated into.
startNLPEngine	Starts the NLP engine.
trainModel	Trains the model based on given inputs and configuration.
generateResponse	Generates the response based on given inputs and configurations.
getUserAPIToken	Returns current user's OpenAI API Token.
authenticate	Authenticates the users based on their OpenAI API Token.
getLanguage	Gets the current language that the user chooses to communicate with the AI chatbot.
getEngineVersion	Gets the engine's current version.

Table 6: Improved Internal Interface Operation Descriptions

5.2.4 Interaction Patterns

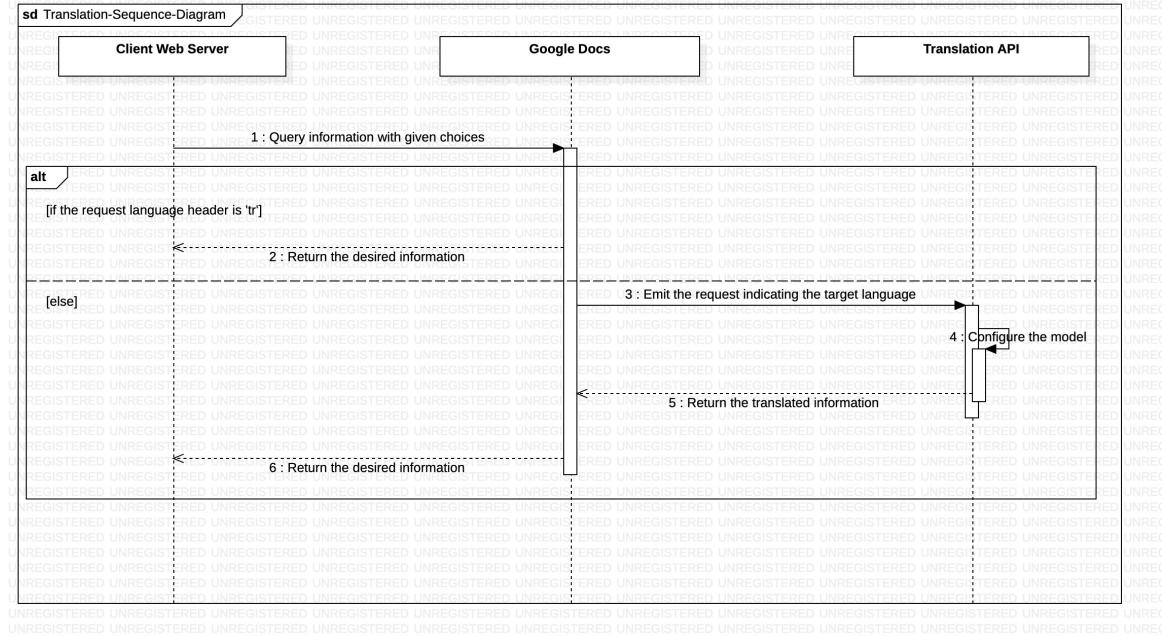


Figure 17: Translation Sequence Diagram

5.3 Information View

5.3.1 Stakeholders' uses of this view

There are 5 main stakeholders of afetbilgi.com, which are suppliers, admins, users, government, and developers. The government uses this view to understand how the data is flowing throughout the program and if it's legal or not. Admins and developers use this view to understand the strategies used within the process of data transfer, data modeling, and structuring of the data. Users and suppliers use this view to understand how to contribute information to the system or take information from the system.

5.3.2 Database Class Diagram

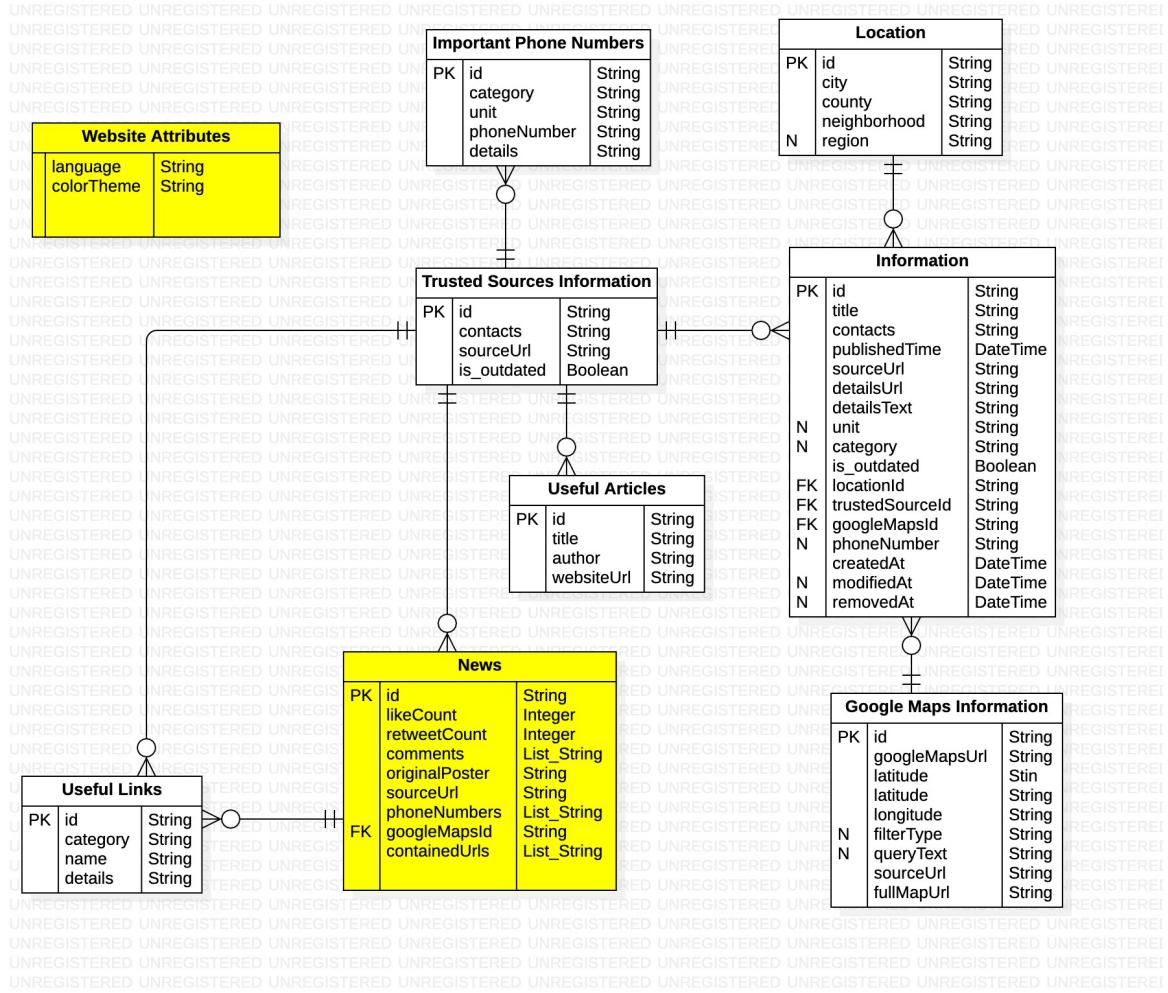


Figure 18: Improved Database Class Diagram

5.3.3 Operations on Data

Operation	Description
-----------	-------------

getLanguage	Create: Read: Language Update: Delete:
getFromLanguage	Create: Read: Language Update: Delete:
getToLanguage	Create: Read: Language Update: Delete:
correctMisspelledWords	Create: Read: Update: Information Delete:
translate	Create: Read: Update: Information Delete:
generateResponse	Create: Information Read: Update: Delete:
getEngineVersion	Create: Read: Version Update: Delete:
getWebsiteLanguage	Create: Read: Language Update: Delete:
setWebsiteLanguage	Create: Read: Update: Language Delete:

getWebsiteTheme	Create: Read: Color Update: Delete:
setWebsiteTheme	Create: Read: Update: Color Delete:

Table 7: Improved CRUD Operations

5.4 Deployment View

5.4.1 Stakeholders' uses of this view

There are 5 main stakeholders of afetbilgi.com, which are suppliers, admins, users, government, and developers. The government uses this view to understand how the system is deployed and if it's legal or not. Developers and admins use this view to understand the technologies and strategies used within the deployment process. Suppliers and users use this view to understand the capabilities of the system.

5.4.2 Deployment Diagram

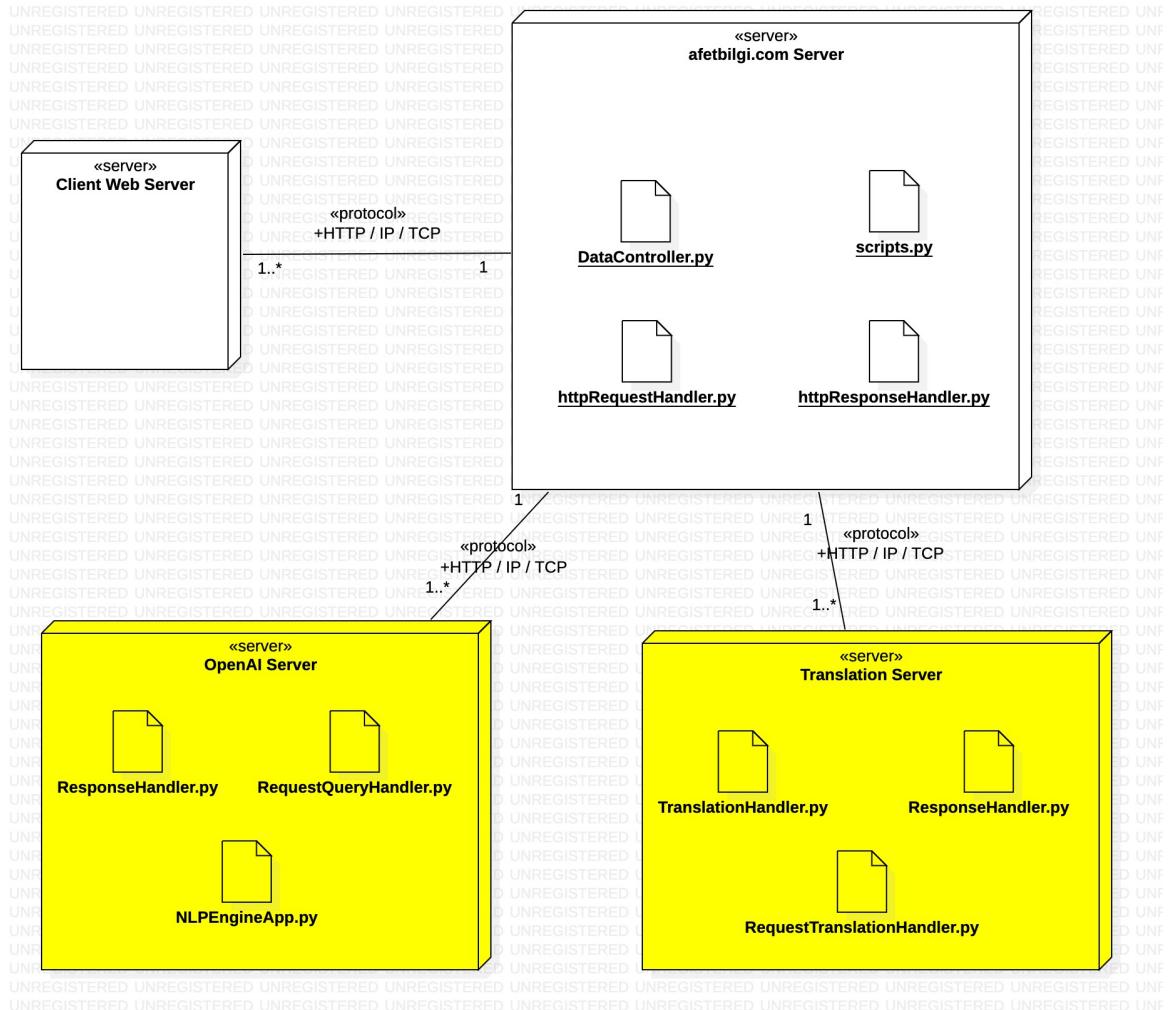


Figure 19: Improved Deployment Diagram

- OpenAI Server and Translation Server communicate with the afetbilgi.com servers using the HTTP / IP / TCP protocols.
- afetbilgi.com ecosystem consists of two new servers, namely OpenAI server, and Translation Server, which are written in Python.
- Open AI Server has Response Handler and Request Query Handler files which handle the user queries in the form of HTTP and returns a response with the answer to the user's query in the form of HTTP.

- Translation Server has Response Handler and Request Translation Handler files which handle the user queries in the form of HTTP and returns the translation of the user's input in the form of HTTP.

5.5 Design Rationale

5.5.1 Context View

The rationale behind the context view is to provide an overall understanding of the entire system with its external interfaces, interactions and its impact as a whole. This view mainly focuses on requirements analysis, interactions with stakeholders, identification of external dependencies, and system scope. Also, the impact of the system on earthquake victims is being discussed in this section.

5.5.2 Functional View

The rationale behind the functional view is to illustrate the functioning components of the system and the interactions between the components. This view mainly focuses on high-level functionality of the system for stakeholders to comprehend how the system is performing the tasks they wanted to be done. It also touches on scalability, modularization, requirements validation and verification concerns of the system.

5.5.3 Information View

The rationale behind the information view is to depict how the information is processed throughout the entire program flow. This view mainly focuses on data, more specifically, the structure of the data, the interactions between data, the transformation process of the data, and how the data is persisted. It also touches on consistency, interoperability, integration, security, and privacy concerns of the system.

5.5.4 Deployment View

The rationale behind the deployment view is to provide a comprehensive overview of how the website will be deployed to its target environment. This view mainly focuses on the physical infrastructure of the system and explains how the modules, submodules, and components are distributed to make up the whole system. It also touches on scalability, performance, availability, fault tolerance, communication between components, and resource management factors of the system.