



MIDDLE EAST TECHNICAL UNIVERSITY

DEPARTMENT
OF COMPUTER ENGINEERING

CENG350 SOFTWARE ARCHITECTURE
DESCRIPTION
afetbilgi.com Project

Group-93

Mert Can BİLGİN
2655181

17/05/2023

Table of Contents

1. Introduction	6
1.1. Purpose and objectives of afetbilgi.com	6
1.2. Scope	6
1.3. Stakeholders and their concerns.....	6
2. References	7
3. Glossary.....	7
4. Architectural Views	8
4.1. Context View	8
4.1.1. Stakeholders' uses of this view	8
4.1.2. Context Diagram	8
4.1.3. External Interfaces	9
4.1.4. Interaction Scenarios	10
4.2. Functional View	12
4.2.1. Stakeholders' uses of this view	12
4.2.2. Component Diagram	12
4.2.3. Internal Interfaces	13
4.2.4. Interaction Patterns	14
4.3. Information View.....	16
4.3.1. Stakeholders' uses of this view	16
4.3.2. Database Class Diagram	16
4.3.3. Operations on Data	17
4.4. Deployment View	18
4.4.1. Stakeholders' uses of this view	18
4.4.2. Deployment Diagram	18
4.5. Design Rationale	19
5. Architectural Views for Suggestions to Improve the Existing System.....	20
5.1. Context View	20
5.1.1. Stakeholders' uses of this view	20
5.1.2. Context Diagram	20
5.1.3. External Interfaces	21
5.1.4. Interaction Scenarios	21
5.2. Functional View	22
5.2.1. Stakeholders' uses of this view	22
5.2.2. Component Diagram	22
5.2.3. Internal Interfaces	23
5.2.4. Interaction Patterns	24
5.3. Information View.....	24
5.3.1. Stakeholders' uses of this view	24
5.3.2. Database Class Diagram	25
5.3.3. Operations on Data	25
5.4. Deployment View	26

5.4.1.	Stakeholders' uses of this view	26
5.4.2.	Deployment Diagram	26
5.5.	Design Rationale	27

List of Figures

Figure 1: Context Diagram	9
Figure 2: External Interfaces Diagram.....	10
Figure 3: Activity Diagram of Getting the address of a food distribution center in Map.....	11
Figure 4: Activity Diagram of Getting the address of a food distribution center in PDF.....	11
Figure 5: Component Diagram	12
Figure 6: Internal Interfaces Diagram.....	13
Figure 7: Sequence Diagram of Validation	14
Figure 8: Sequence Diagram of Selecting a City	15
Figure 9: Sequence Diagram of Selecting the Language.....	15
Figure 10: Database Class Diagram.....	16
Figure 11: Deployment Diagram	18
Figure 12: System Context Diagram with Suggestions	20
Figure 13: External Interfaces with Suggestions	21
Figure 14: Activity Diagram of Listen PDF	22
Figure 15: Component Diagram After Suggestions.....	23
Figure 16: Internal Interfaces Diagram After Suggestions	23
Figure 17: Sequence Diagram of Request Help.....	24
Figure 18: Database Class Diagram After Suggestions	25
Figure 19: Deployment Diagram After Suggestions	26

List of Tables

Table 1: Table of Operations on Data.....	17
Table 2: Operations on Database Class Diagram After Suggestions.....	26

1. Introduction

1.1. Purpose and objectives of afetbilgi.com

The purpose of developing afetbilgi.com is to create a user-friendly platform that provides accurate and up-to-date information about natural disasters and emergency response procedures. The goal is to help individuals and organizations prepare for, respond to, and recover from disasters more efficiently and effectively. The system aims to address the information gap and communication challenges that often arise during disasters by providing a centralized platform where users can access relevant information and resources. The proposed system will contribute to meeting the business objectives of the organization by improving disaster preparedness and response, minimizing losses, and saving lives. The system will be designed to be easily accessible and user-friendly, allowing stakeholders to quickly and easily find the information they need during times of crisis.

1.2. Scope

The system provide users with various search options for emergency response and relief services. This includes searching for safe gathering places, temporary accommodation areas, evacuation points, transportation aids, food distribution centers, services outside disaster areas, gas stations, mobile toilets, and health services. The search feature will allow users to filter the search results based on location. The software will also provide users with real-time updates on the availability and status of each service. Additionally, the system will enable users to save and share their search results with others through various communication channels.

PDF Generator module allows users to generate PDF reports based on the data collected on the platform.

The Google Maps module is a key component of the proposed system for afetbilgi.com. It will allow users to view and interact with maps of the disaster area, as well as to search for specific locations, such as emergency gathering areas, safe gathering places, evacuation points, and temporary accommodation places. Overall, the search functionality of the software will assist users in finding essential services and resources during a disaster or emergency situation.

Amazon Web Services (AWS) is a cloud computing platform that provides a wide range of services to support the development, deployment, and management of applications in the cloud. The scope of AWS for afetbilgi.com includes the use of AWS services to host the application, store data, and provide other necessary functionalities. The use of AWS will enable the application to scale quickly and efficiently, ensure high availability, and provide reliable and secure services to users.

1.3. Stakeholders and their concerns

There are three main user groups of afetbilgi.com, each with distinct characteristics and requirements. The first stakeholder group comprises the victims of natural disasters. These users

often face limited access to information and resources during emergencies. Therefore, the system aims to provide them with essential information about critical phone numbers, safe gathering places, and other crucial services. To cater to their needs, the system should be user-friendly, easily accessible, and responsive, as these users may have limited technical expertise and require quick access to vital information. The second stakeholder group consists of individuals and organizations who want to provide assistance, such as volunteers and aid organizations. These users seek information on how best to assist the victims, access and provide resources, and coordinate with other groups. They may possess a range of technical expertise, from basic to advanced. Thus, the system should be designed to accommodate users with different levels of knowledge and skills, offering intuitive features and providing guidance to streamline their efforts effectively. The third stakeholder group comprises data collectors and validators responsible for gathering and managing disaster-related information. These users require access to more detailed data and analysis tools to make informed decisions regarding resource allocation and emergency response. With advanced technical expertise, they can leverage a more complex system that offers comprehensive data collection mechanisms, advanced analysis features, and robust reporting capabilities to support their decision-making processes.

2. References

This document is prepared with respect to IEEE 29148-2011 standard:

29148-2018 - ISO/IEC/IEEE International Standard - Systems and software engineering – Life cycle processes –Requirements engineering.

Other sources:

GitHub. (n.d.). GitHub Pages. Retrieved April 21, 2023, from <https://github.com/alpaylan/afetbilgi.com>

Amazon Web Services. (n.d.). What is AWS? Retrieved April 21, 2023, from <https://aws.amazon.com/tr/what-is-aws/>

3. Glossary

- React: A JavaScript library for building user interfaces.
- TypeScript: It is a superset of JavaScript that adds features to improve development experience.
- HTTP (Hypertext Transfer Protocol): It is a standar communication protocol used to transfer data between web servers and clients, such as web browsers.
- Proxy Server: It is an intermediary server that acts as a gateway between a client and the internet to provide additional security, privacy and control.

- **SaaS (Software as a Service):** It is a cloud computing model where software applications are provided by a third-party provider over the internet, allowing users to access the software and its features without the need for local installation or maintenance.
- **API (Application Programming Interface):** It is a set of protocols and tools for building software applications that allows different software systems to communicate and share data with each other.
- **TLS (Transport Layer Security):** It is a cryptographic protocol used to establish a secure communication channel between two computer systems, typically a web server and a web browser, to ensure the confidentiality and integrity of data transmitted over the internet.

4. Architectural Views

4.1. Context View

4.1.1. Stakeholders' uses of this view

The stakeholders primarily utilize the context view to gain a high-level understanding of the system and its interactions with the external environment. This view provides stakeholders with a comprehensive overview of how the system interacts with its surrounding entities, including external systems and users. They use context view to understand system boundaries. This helps in understanding which components and systems are part of the overall architecture and how they communicate with each other. In addition, they can identify external systems, services, and resources on which the afetbilgi.com relies. Also, they can examine the interfaces and interactions between the system and external entities. This helps in understanding how information and data flow between different components and systems.

4.1.2. Context Diagram

The system perspective of afetbilgi.com involves several main components, including data collection, data validation, and processing, user interface, integration with external entities, and disaster response. Data is collected from various sources, including social media, news outlets, and official government agencies, and is validated by data validators to ensure accuracy and reliability. The data is then processed and shared with people, which can be used to guide decision-making during disaster response efforts. The user interface provides a simple, fast, and user-friendly way for users to access information, including search functions and data visualization. The system integrates with external entities such as Google Maps and Amazon Web Services to provide additional functionality and support. Finally, the ultimate goal of the system is to support disaster response efforts by providing critical information to first responders and other stakeholders, including information on the location of victims, and the availability of resources.

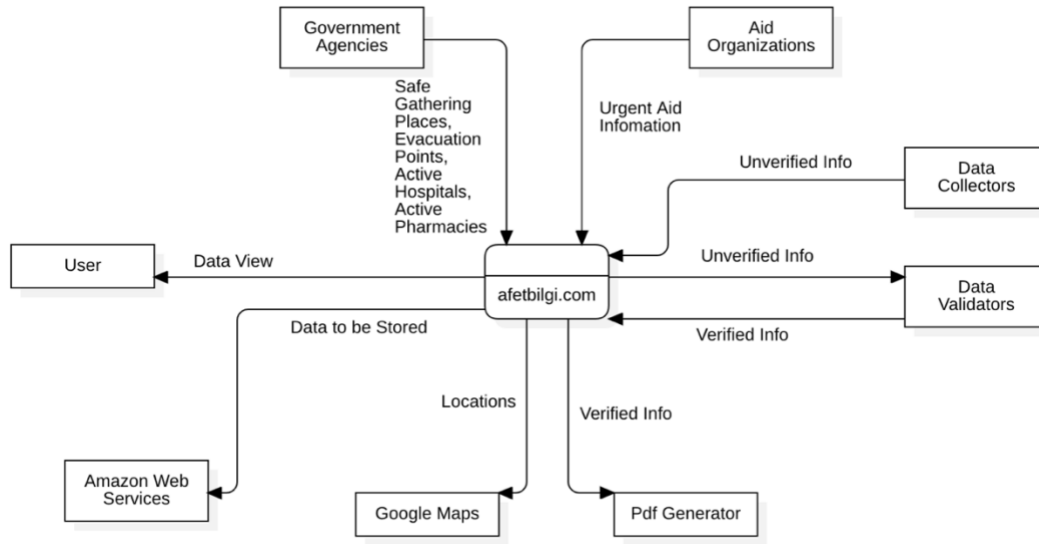


Figure 1: Context Diagram

4.1.3. External Interfaces

The user interface acts as the primary interface for users to interact with the system. Users can initiate operations to retrieve data such as safe gathering places and evacuation points. Also, they can select the language which they want to use. The maps interface enhances the user experience by providing a visual representation of the disaster area. Users can utilize zoomIn, zoomOut, and search operations to navigate the map, locate specific points of interest, and gain a better understanding. The PDF interface complements the user interface by providing the ability to generate and download PDF reports based on the data collected on the platform. Users can access all relevant information about emergency response, to assist in decision-making. Also, this provides a printable documentation.

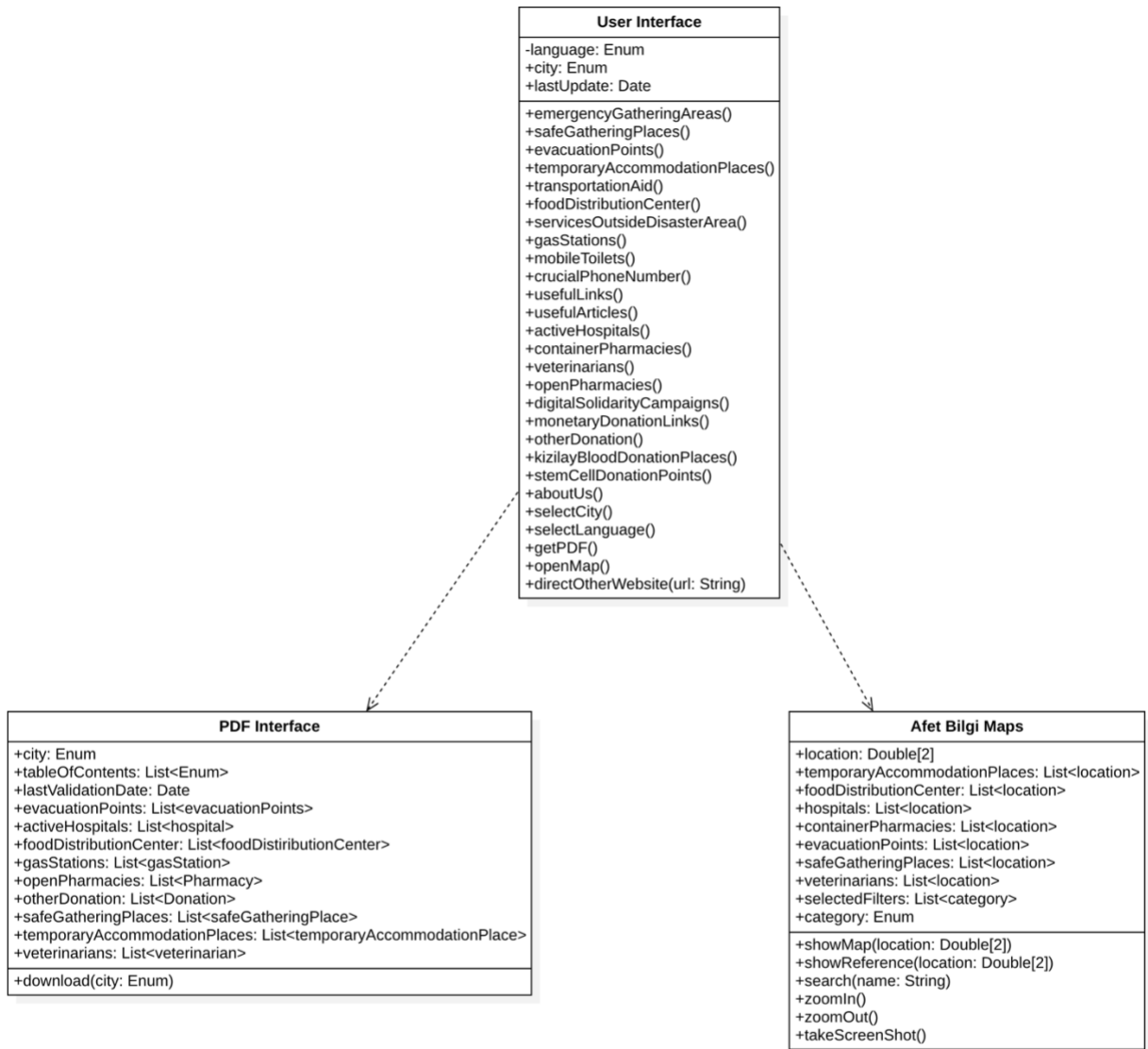


Figure 2: External Interfaces Diagram

4.1.4. Interaction Scenarios

Interaction between interfaces while the getting address of a food distribution center is provided in two different activity diagrams below. The first one shows the interaction between User Interface and Maps, and the second one shows the interaction between User Interface and PDF Generator.

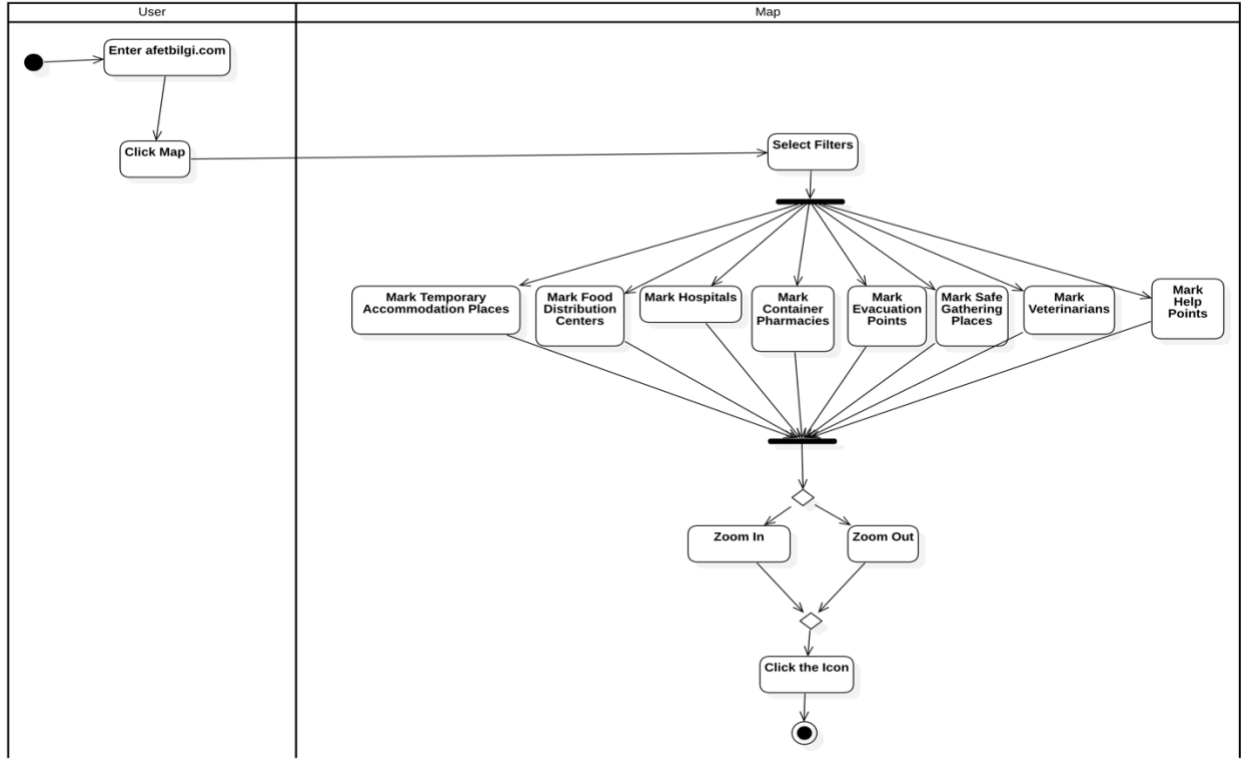


Figure 3: Activity Diagram of Getting the address of a food distribution center in Map

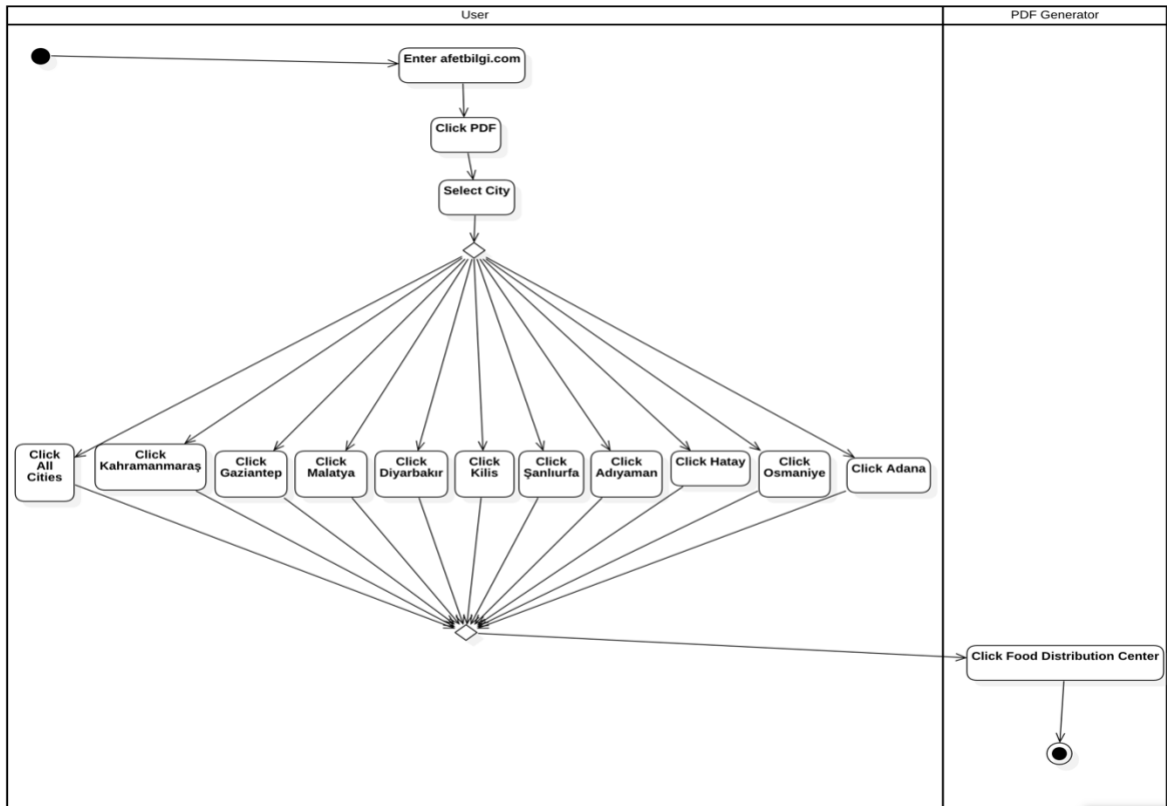


Figure 4: Activity Diagram of Getting the address of a food distribution center in PDF

4.2. Functional View

4.2.1. Stakeholders' uses of this view

The first type of stakeholders which is the victims of natural disasters utilize the Functional View to access the system's user interface and perform essential operations. They rely on functionalities such as retrieving crucial phone numbers during emergencies, searching for safe gathering places for shelter, and finding nearby gas stations. By interacting with the user interface through the Functional View, victims can quickly access the information they need to ensure their safety and well-being during challenging times. The second type of stakeholders involved in providing assistance, such as volunteers and aid organizations, utilize the Functional View to understand the system's functionalities and identify ways to support affected individuals. They explore the available operations in the user interface, such as searching for donation links and searching for evacuation points. By leveraging the Functional View, they can coordinate their actions, utilize the system's functionalities as a resource, and provide timely and targeted assistance to the affected population. The third type of stakeholders is data collectors and validators utilize the Functional View to assess the system's data collection and analysis capabilities. These stakeholders may require advanced functionalities, such as data export or advanced analysis tools, to support their decision-making processes.

4.2.2. Component Diagram

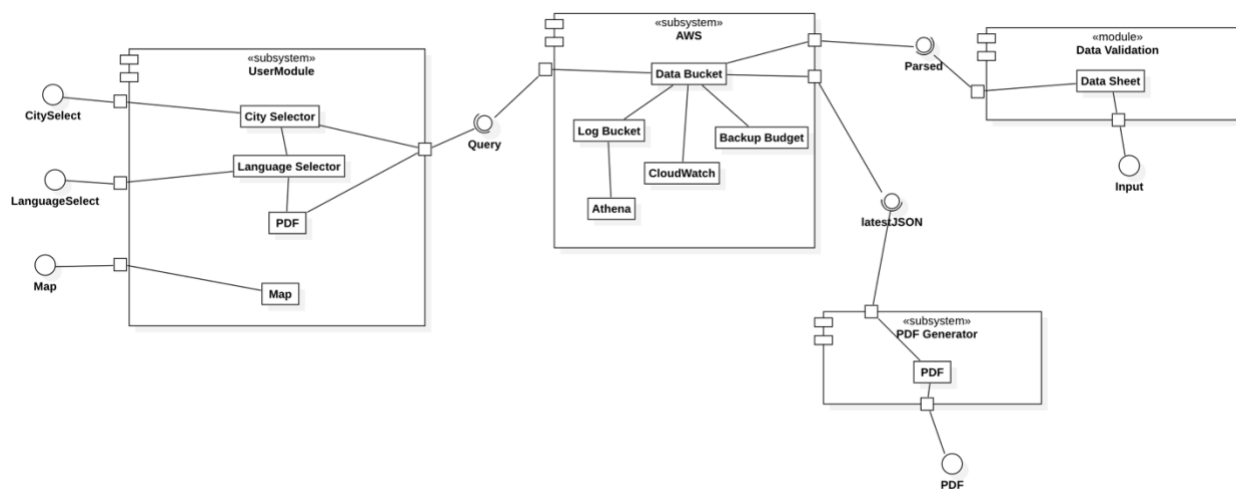


Figure 5: Component Diagram

In the component diagram, there are four subsystems.

The first, in UserModule, City Selector allows users to select their desired city to refine the search results and access specific information related to that city with a provided interface. Language Selector enables users to choose their preferred language for the user interface and content display with a provided interface. PDF part handles the generation of PDF reports based on the data collected from the system. First of all, it gets the city from the city selector, then get the information from AWS, and AWS send the information to the PDF Generator. Maps part integrates with

mapping services to display interactive maps of the disaster area, providing users with visual representations of important locations such as evacuation points, food distribution centers and hospitals.

The second, AWS, Data Bucket represents the storage facility on AWS that holds the data collected and managed by the system. Log Bucket stores the system's logs and related information for auditing and monitoring purposes. Backup Bucket handles the backup and recovery of system data to ensure data integrity and availability in case of any failures. CloudWatch monitors the system's resources, performance, and availability on AWS. Athena enables querying and analyzing data stored in the AWS environment, providing data retrieval and analysis capabilities for the system.

The third, Data Validation, Data Sheet consists of two parts which are Hot and Cold. This part is responsible for validated and verified data and integrity of the data collected within the system. It ensures that the data stored in the system is reliable and trustworthy.

The fourth, PDF Generator, PDF part generates PDF reports based on the data collected and stored in the system. It utilize the data from the AWS storage to generate the reports.

Also, there are three relation between components. The first is Query between AWS and UserModule. It indicates that the User Module interacts with the AWS component to retrieve and query data. AWS has provided interface, and UserModule has required interface in this relation. The second is Parsed Data relation between AWS and Data Validation. The data in the sheets is parsed, and send to the AWS. The third is latestJSON. It shows that the PDF Generator module receives the latest JSON data from the AWS component to generate PDF reports.

4.2.3. Internal Interfaces

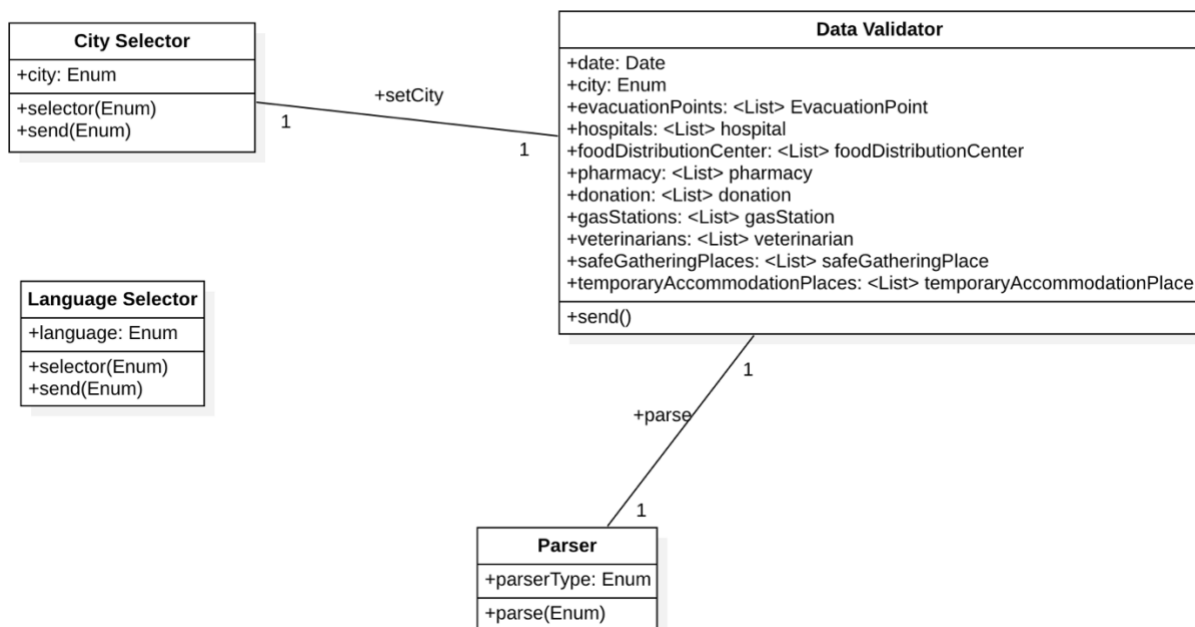


Figure 6: Internal Interfaces Diagram

Data Validator is responsible for validating and verifying the data collected within the system. It ensures that the data is accurate. Parser is responsible for processing and transforming raw data into a structured format that can be easily consumed by other system components. It takes the input data, which can be BarinmaParser, EczaneParser, ToplanmaParser, VeterinerParser, YemekParser, HastaneParser, TahliyeParser. Then it applies necessary formatting. It prepares the data for further processing and validation by the Data Validator. Language Selector allows users to choose their preferred language for the user interface and content display. City Selector component enables users to select their desired city. Both city selector and language selector can send the data to relevant classes.

4.2.4. Interaction Patterns

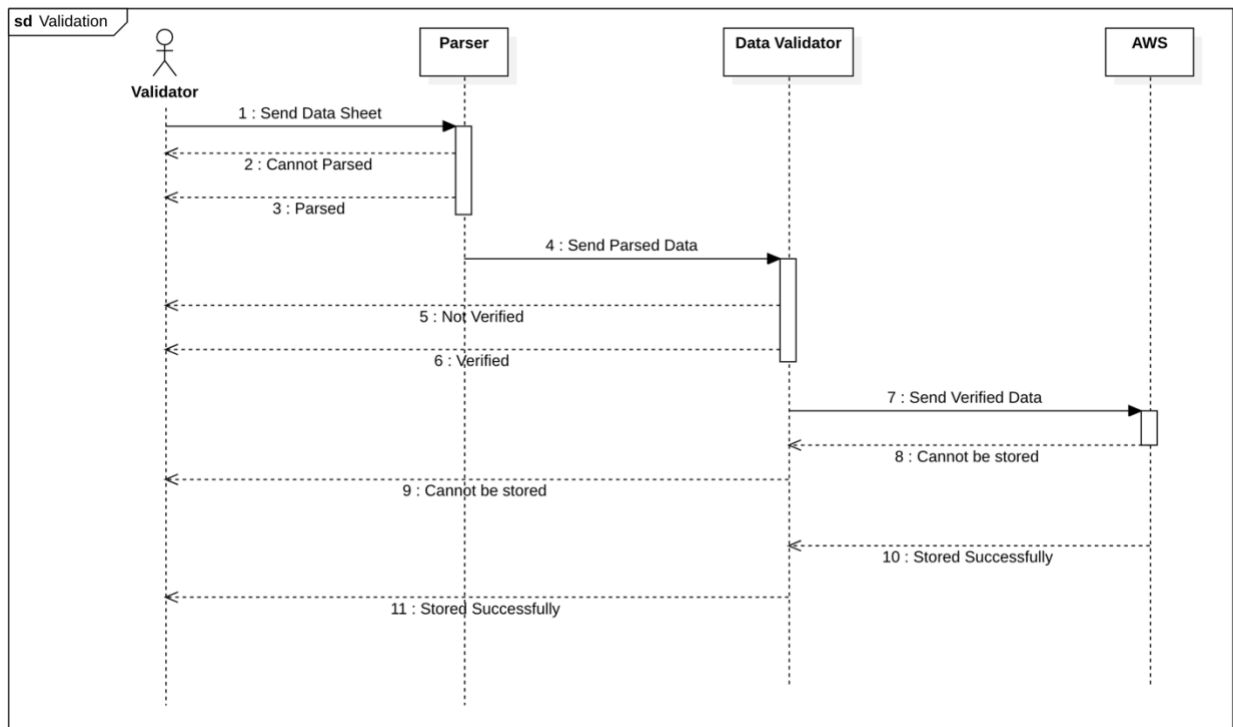


Figure 7: Sequence Diagram of Validation

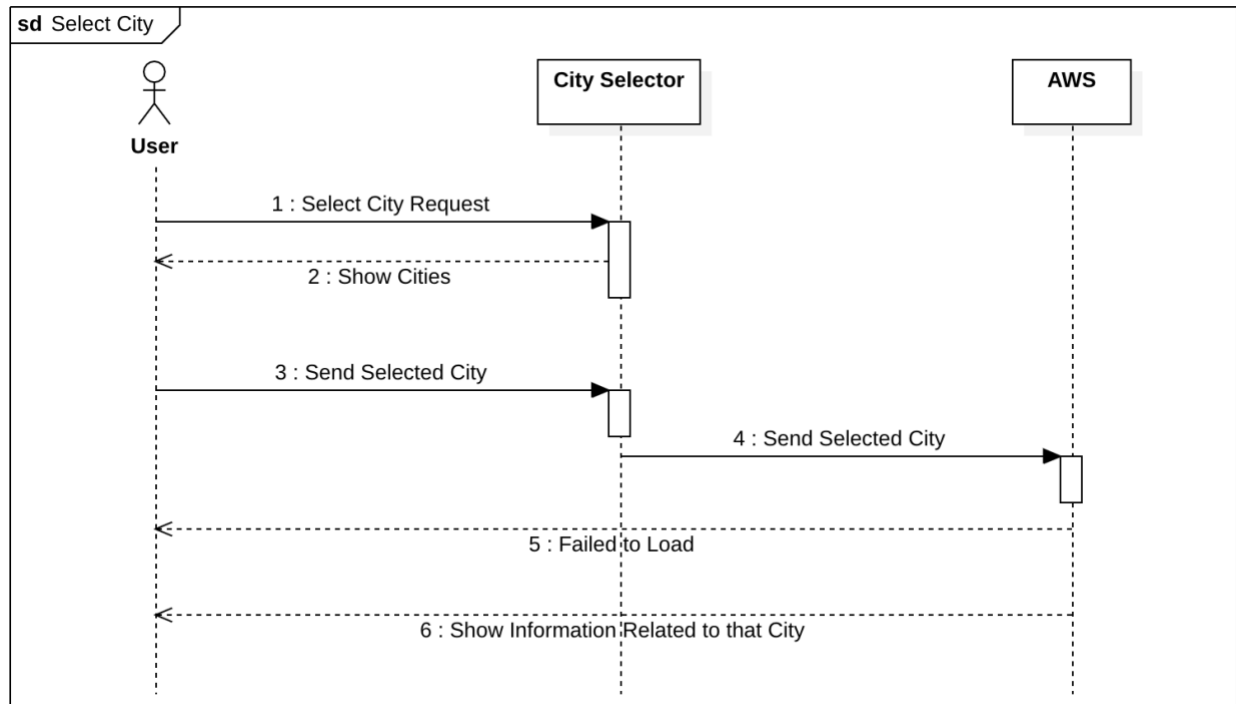


Figure 8: Sequence Diagram of Selecting a City

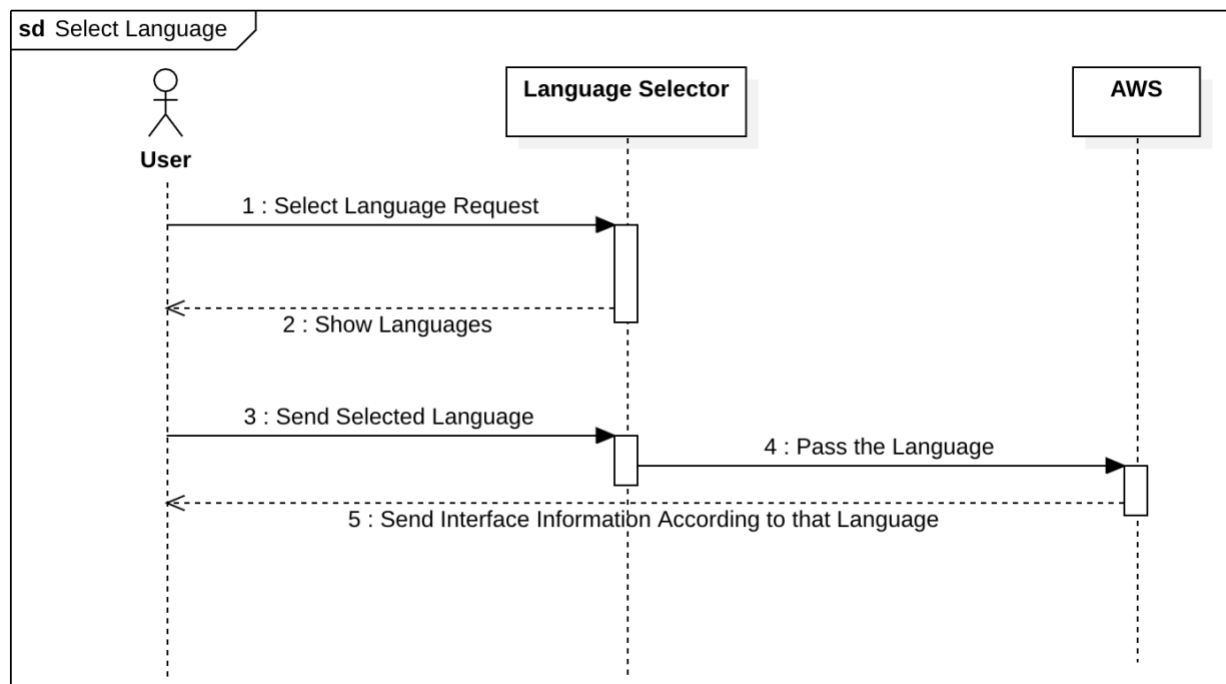


Figure 9: Sequence Diagram of Selecting the Language

4.3. Information View

4.3.1. Stakeholders' uses of this view

The first type of stakeholders which is the victims of natural disasters rely on the Information View to access accurate and up-to-date information related to emergency response and relief services. They utilize the user interface to retrieve essential data such as crucial phone numbers, safe gathering places, gas stations, and other vital resources. The second type of stakeholders involved in providing assistance, such as volunteers and aid organizations rely on the system's data management capabilities to access relevant information and resources needed for their assistance efforts. By leveraging the Information View, these stakeholders can coordinate their actions effectively. The third type of stakeholders is data collectors and validators rely on the Information View to understand the data models, data sources, and data flows within the system. By utilizing the Information View, these stakeholders can ensure data integrity, analyze the quality of collected information, and make informed decisions based on the available data. They may also utilize data validation tools and processes within the system to verify the accuracy and reliability of the collected data.

4.3.2. Database Class Diagram

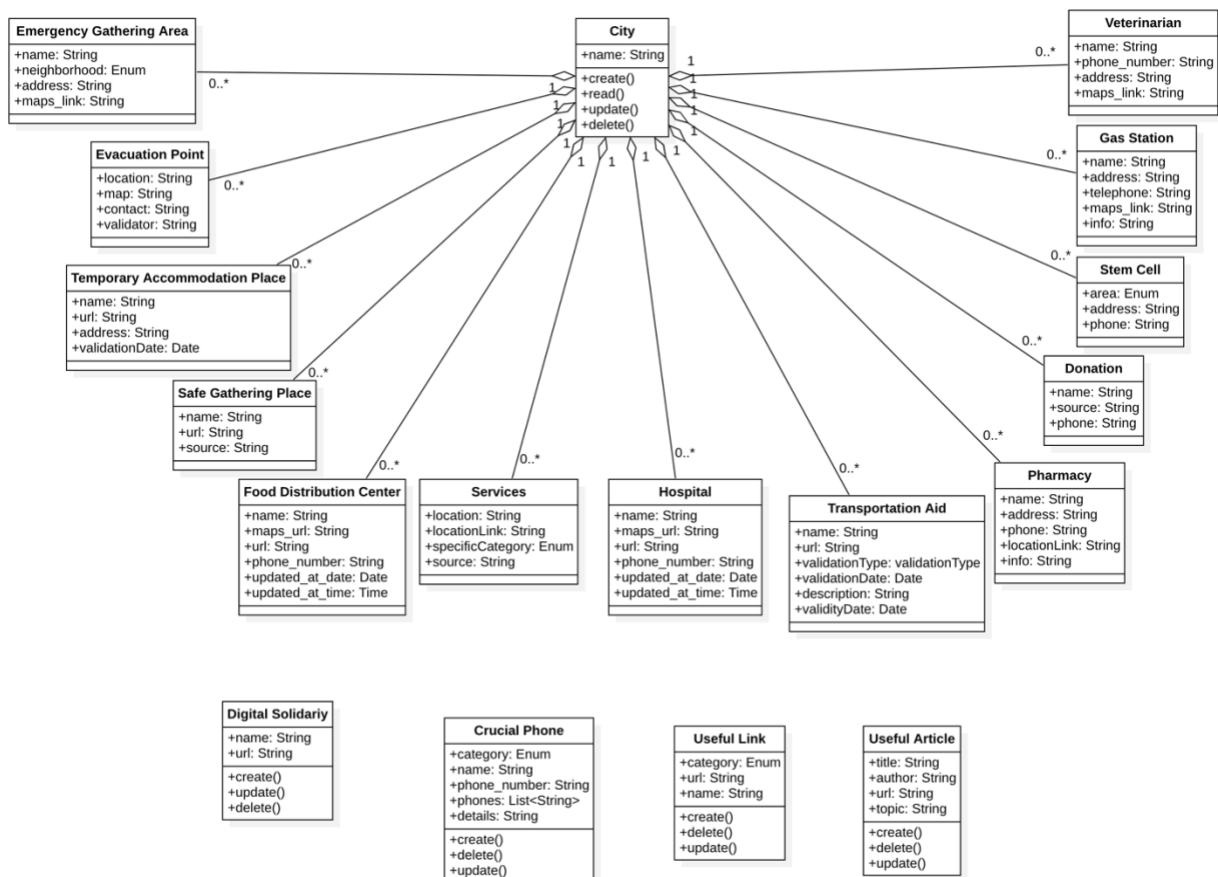


Figure 10: Database Class Diagram

4.3.3. Operations on Data

Operation	Description
create()	The create() operation in the City class is responsible for creating a new city entry in the database. It initializes a new city object with the provided data and saves it to the database, creating a new record for the city.
delete()	The update() operation in the City class is used to modify the existing city information in the database.
update()	The delete() operation in the City class is responsible for removing a city record from the database.
read()	The read() operation in the City class is used to retrieve the information of a specific city from the database.
create() in Digital Solidarity	The create() operation in the Digital Solidarity class is responsible for creating a new entry in the database. It initializes a new object with the provided data and saves it to the database.
update() in Digital Solidarity	The update() operation in the Digital Solidarity class is used to modify the existing information in the database.
delete() in Digital Solidarity	The delete() operation in the Digital Solidarity class is responsible for removing a record from the database.
create() in Crucial Phone	The create() operation in the Crucial Phone class is responsible for creating a new entry in the database. It initializes a new object with the provided data and saves it to the database.
delete() in Crucial Phone	The delete() operation in the Crucial Phone class is responsible for removing a record from the database.
update() in Crucial Phone	The update() operation in the Crucial Phone class is used to modify the existing information in the database.
create() in Useful Link	The create() operation in the Useful Link class is responsible for creating a new entry in the database. It initializes a new object with the provided data and saves it to the database.
delete() in Useful Link	The delete() operation in the Useful Link class is responsible for removing a record from the database.
update() in Useful Link	The update() operation in the Useful Link class is used to modify the existing information in the database.
create() in Useful Article	The create() operation in the Useful Article class is responsible for creating a new entry in the database. It initializes a new object with the provided data and saves it to the database.
delete() in Useful Article	The delete() operation in the Useful Article class is responsible for removing a record from the database.

Table 1: Table of Operations on Data

4.4. Deployment View

4.4.1. Stakeholders' uses of this view

The first type of stakeholders which is the victims of natural disasters rely on the Deployment View to ensure the system's availability and accessibility during emergency situations. They depend on the system administrators to deploy the application on robust and resilient infrastructure, ensuring uninterrupted access to critical information and services. The second type of stakeholders involved in providing assistance, such as volunteers and aid organizations rely on the system administrators to ensure the system's scalability, performance, and availability to handle increased user demand during disaster events. By utilizing the Deployment View, these stakeholders can plan their assistance efforts effectively, knowing that the system is deployed on a robust infrastructure capable of supporting their activities. The third type of stakeholders is data collectors and validators utilize the Deployment View to design the system's deployment architecture, select appropriate infrastructure providers, and configure the necessary resources.

4.4.2. Deployment Diagram

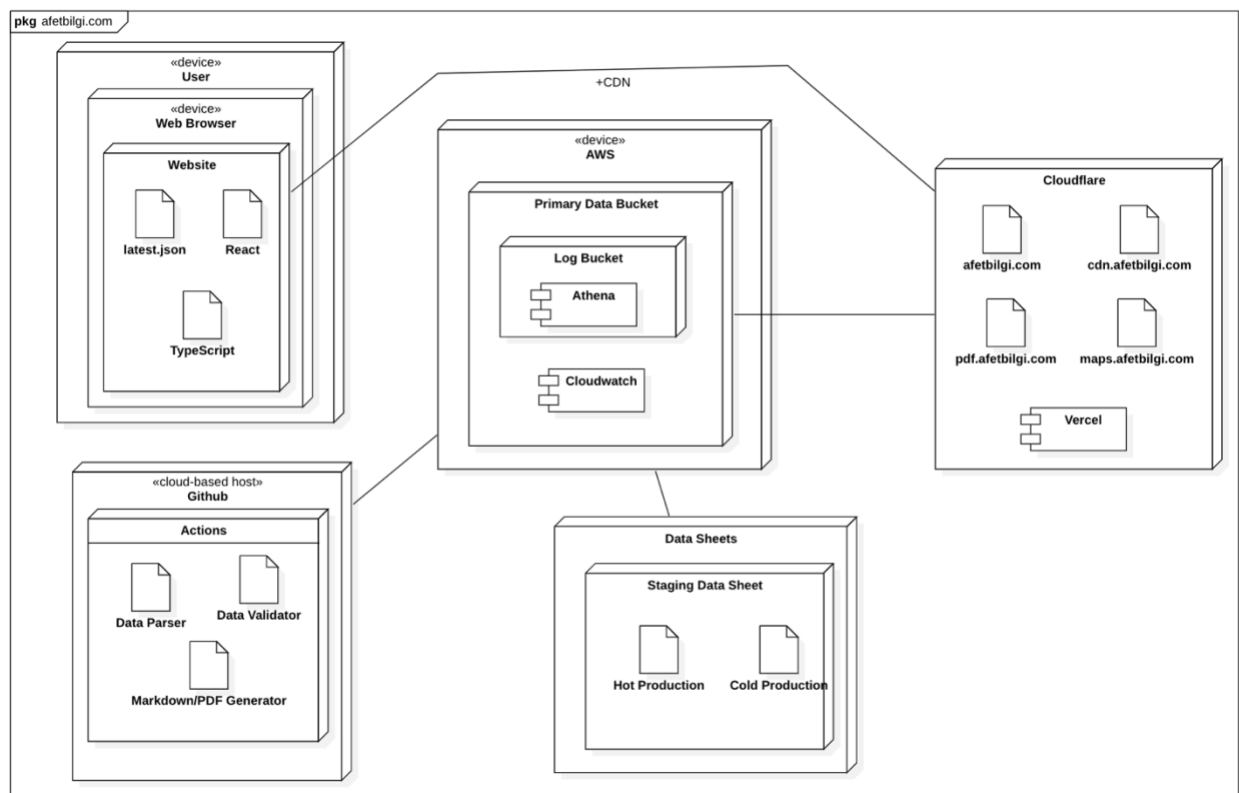


Figure 11: Deployment Diagram

- AWS node represents the cloud computing platform provided by Amazon Web Services (AWS). It stores the data. User Device node represents the end-user devices, such as desktop computers, laptops, tablets, or mobile devices, through which users access and interact with the `afetbilgi.com` website.

- Cloudflare Node represents the content delivery network (CDN). It acts as an intermediary between the website and end-users, caching and delivering static content from edge servers located worldwide to improve performance and security.
- Actions Node represents the various actions such as parsing data, generating PDFs, and data validation within the afetbilgi.com system.
- Data Sheets Node represents the data sheets used by the afetbilgi.com system to store and manage data. It includes entities such as hospitals, evacuation points, temporary accommodation places, and other relevant information related to disaster response.
- Website Node represents the afetbilgi.com website itself. It consists of the TypeScript and React artifacts, which are the technologies used for developing the frontend of the website. These artifacts contain the code and resources required to render the user interface and provide the desired functionality.
- Primary Data Bucket Node represents the primary data storage facility in the AWS environment. It is where the afetbilgi.com system stores and retrieves data, including the data sheets mentioned earlier.

4.5. Design Rationale

i) Context View

The inclusion of User, AWS, Government Agencies, Data Validators, Data Collectors, PDF Generator, and Google Maps in the context view is justified by the need to establish collaboration channels among stakeholders during natural disasters.

ii) Functional View

The inclusion of a language selector is justified by the need to cater to a diverse user base with varying language preferences. So that the system's user accessibility is enhanced.

iii) Information View

The adoption of a database structure with classes like City, Hospitals, Evacuation Points, and the use of JSON files in the information view is justified by the need to organize and manage data efficiently. By structuring data into meaningful classes, the system can provide targeted access to specific information related to cities. The utilization of JSON files allows for flexible data storage and retrieval, facilitating seamless updates and modifications when necessary.

iv) Deployment View

The data parser action is essential to transform the raw data received from various sources into a structured format that can be easily processed and utilized by the system. In a disaster information system like afetbilgi.com, data is collected from diverse sources such as government agencies, data validators, and data collectors. These sources may have different formats, structures, or data models. The data parser action enables the system to integrate and harmonize these disparate data sources by standardizing them into a common format.

5. Architectural Views for Suggestions to Improve the Existing System

5.1. Context View

5.1.1. Stakeholders' uses of this view

The first type of stakeholders which is the victims of natural disasters utilize the "Request Help" section to directly communicate their assistance needs to the authorities, facilitating faster response times and efficient resource allocation. Additionally, the inclusion of a PDF reader and voice record download option enhances accessibility for users with disabilities, ensuring that critical information and resources are accessible to all. The second type of stakeholders involved in providing assistance, such as volunteers and aid organizations can use the "Help Requests" section to respond to requests for help for more effective coordination and allocation of resources. The third type of stakeholders is data collectors and validators verify and validate the assistance requests, ensuring the accuracy and reliability of the data.

5.1.2. Context Diagram

Help requests is collected from various resources, and is validated by data validators to ensure accuracy and reliability. Also, the PDF Reader component sends voice data to the main system, enabling users to access voice-based information.

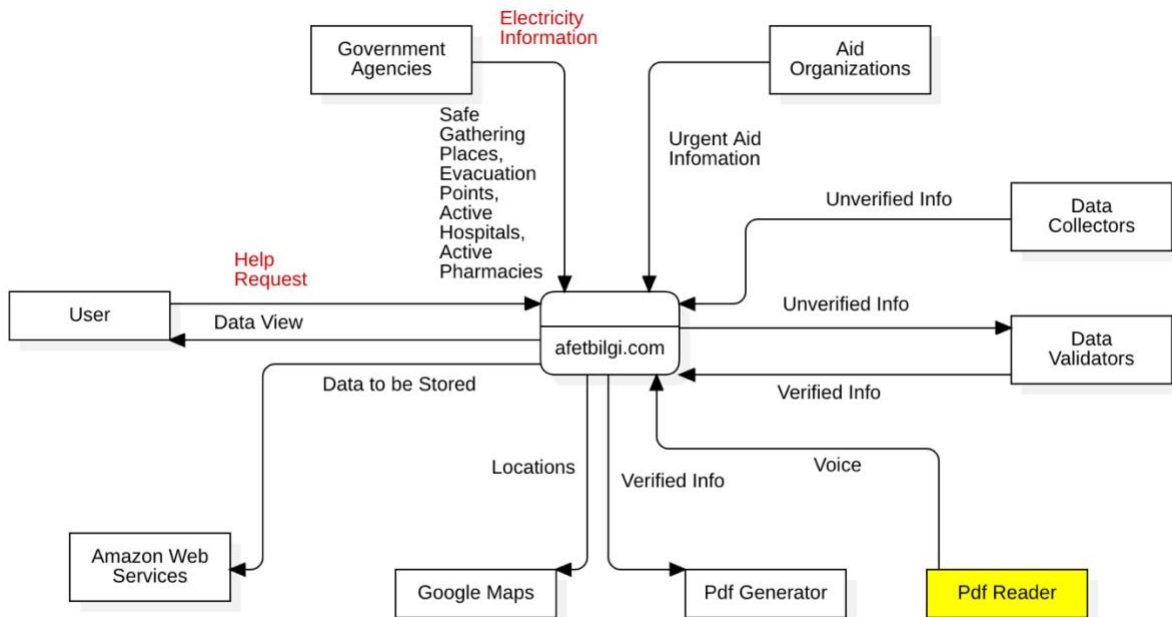


Figure 12: System Context Diagram with Suggestions

5.1.3. External Interfaces

The "requestHelp" operation allows users to submit their assistance needs, providing crucial information. The "markets" operation assists users in locating nearby markets, while the "activeElectricity" operation allows users to check the availability of electricity in a specific area. The "read" operation allows users to listen information from PDF reports.

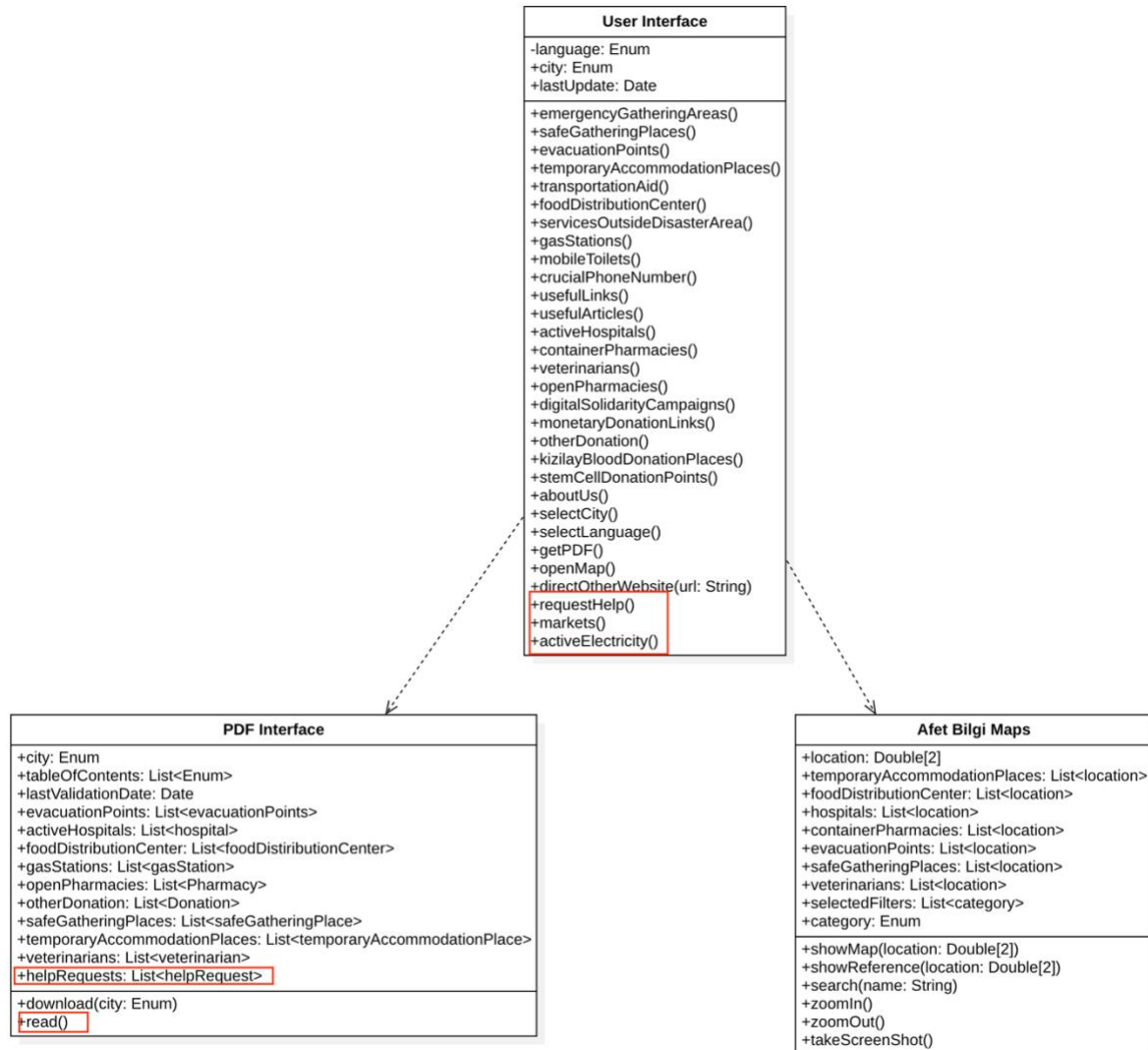


Figure 13: External Interfaces with Suggestions

5.1.4. Interaction Scenarios

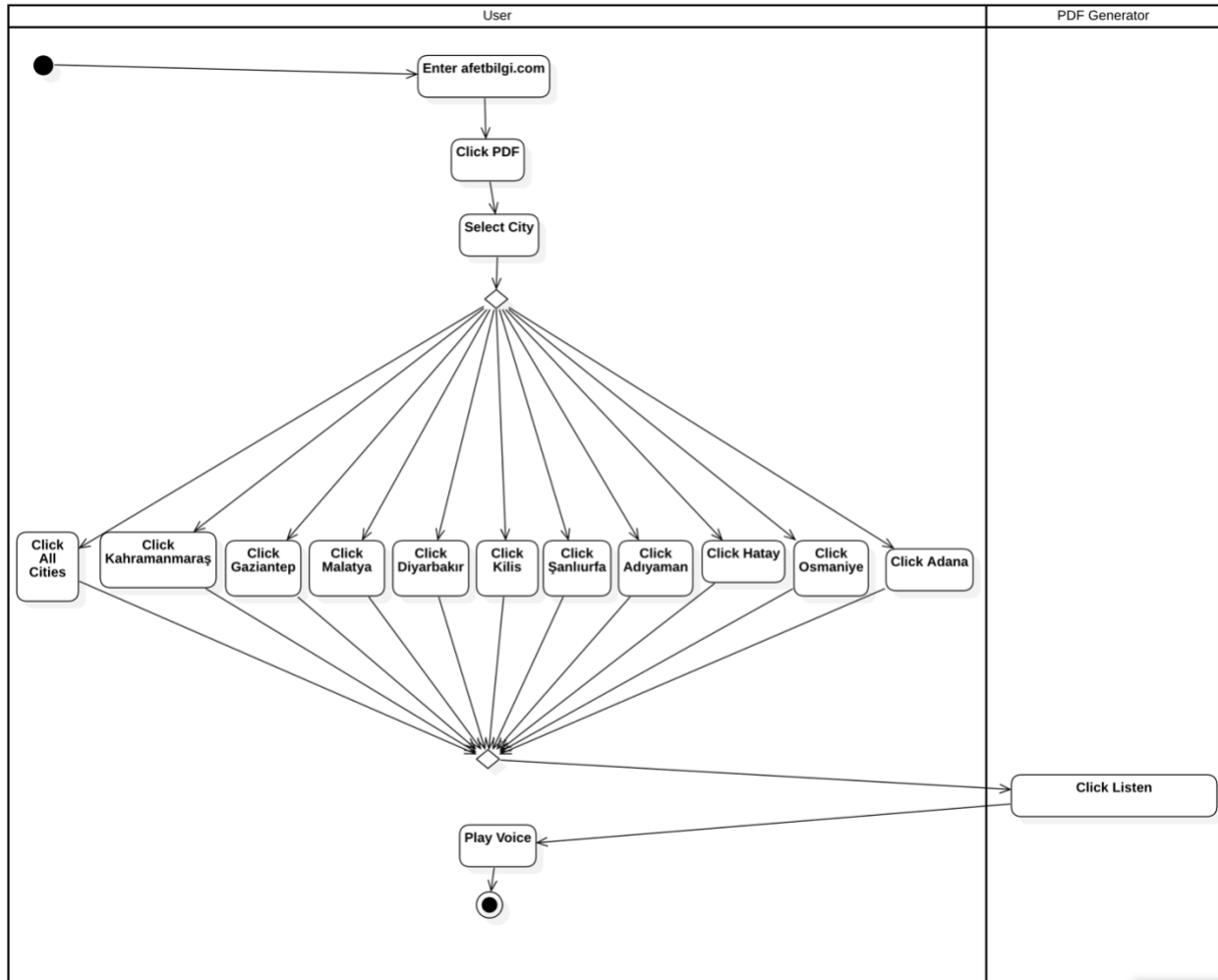


Figure 14: Activity Diagram of Listen PDF

5.2. Functional View

5.2.1. Stakeholders' uses of this view

For the victims of natural disasters, the addition of the "Request Help" section allows them to submit their assistance needs directly through the system. They can also benefit from the PDF reader feature, enabling access to critical information through voice-based features. Individuals and organizations providing assistance can utilize the "Help Requests" section to offer their support and resources to those in need.

5.2.2. Component Diagram

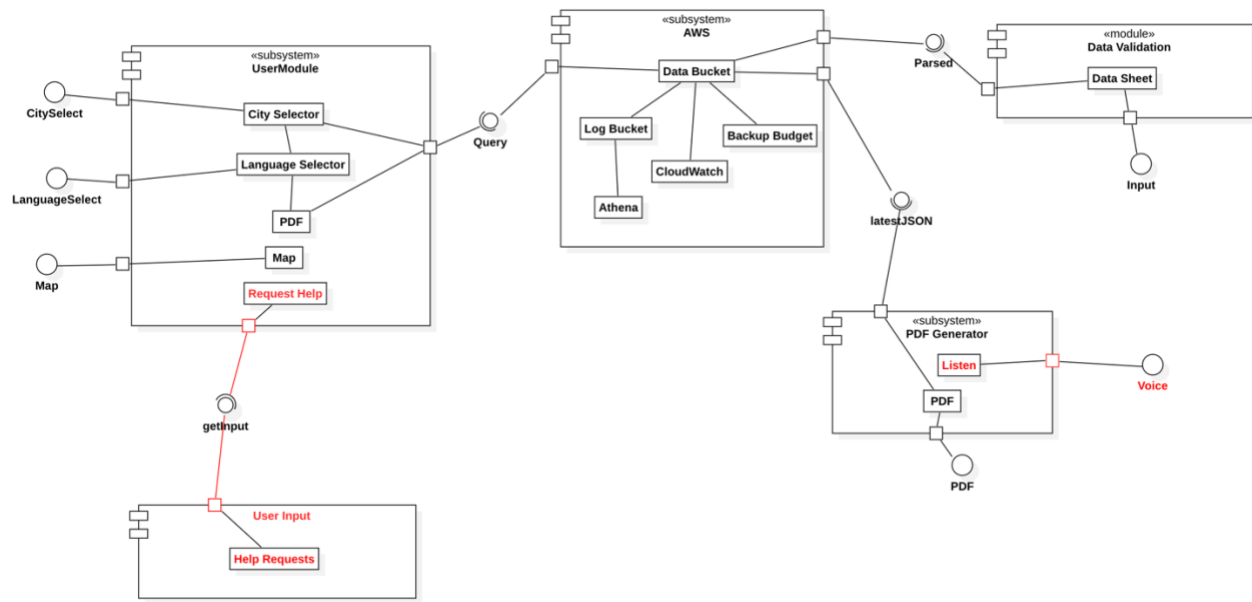


Figure 15: Component Diagram After Suggestions

The UserInput component represents the user interface element responsible for capturing input from users for the "Request Help" feature. It provides an interface that allows users to input their phone number, address, and the type of assistance they require. The PDF Listen component part provides an interface that enables users to access and listen to the voice data embedded in PDF files, providing an alternative means of consuming information for users with disabilities or those who prefer auditory content.

5.2.3. Internal Interfaces

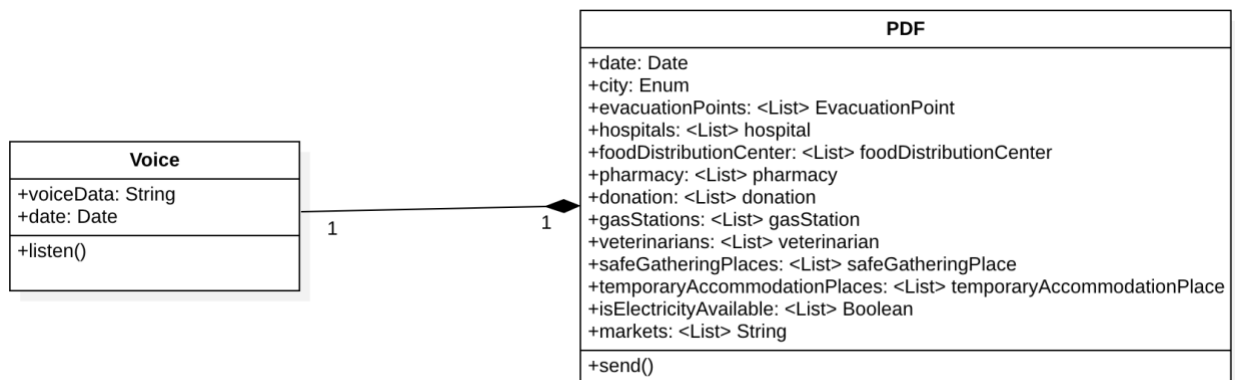


Figure 16: Internal Interfaces Diagram After Suggestions

The Voice Class represents the internal interface responsible for managing voice data within the system. It consists of two attributes: date and voiceData. The "date" attribute stores the date associated with the voice data, providing information about when the data was generated. The "voiceData" attribute holds the actual voice data. The PDF Class is associated with the Voice Class through a relation, indicating that PDF files within the system may include embedded voice data.

This association allows the system to provide users with the capability to access and listen to voice content embedded within PDF files.

5.2.4. Interaction Patterns

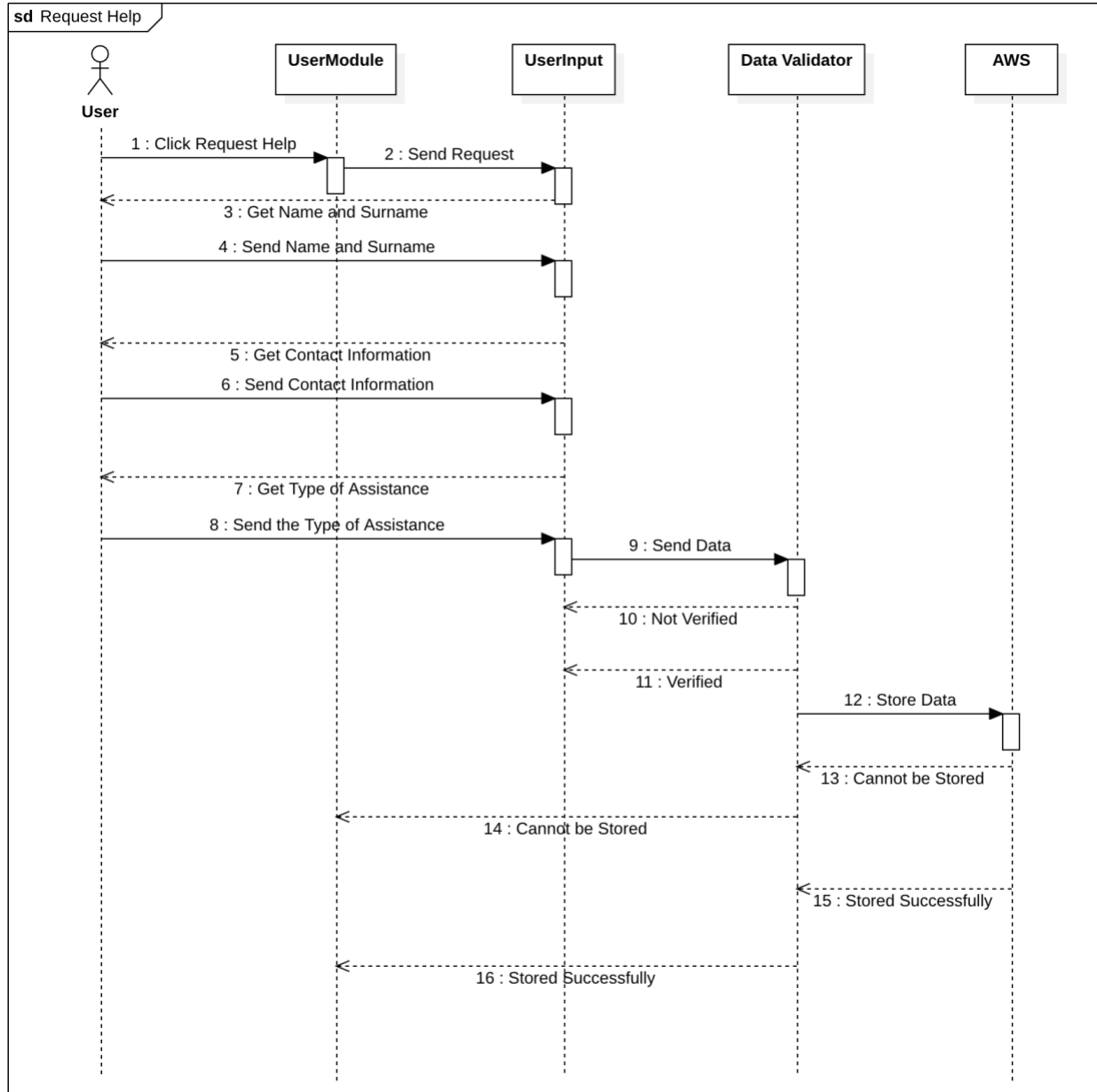


Figure 17: Sequence Diagram of Request Help

5.3. Information View

5.3.1. Stakeholders' uses of this view

For the victims of natural disasters, the addition of the "Request Help" section provides a convenient way to input their assistance requirements and contact information. The integration of a PDF reader with voice capabilities enables individuals with disabilities to access important information in an accessible format. Furthermore, the inclusion of a "Help Requests" section allows individuals and organizations providing assistance to view and respond to specific requests, with better coordination. Data collectors and validators should ensure the accuracy and reliability of the data collected for help requests.

5.3.2. Database Class Diagram

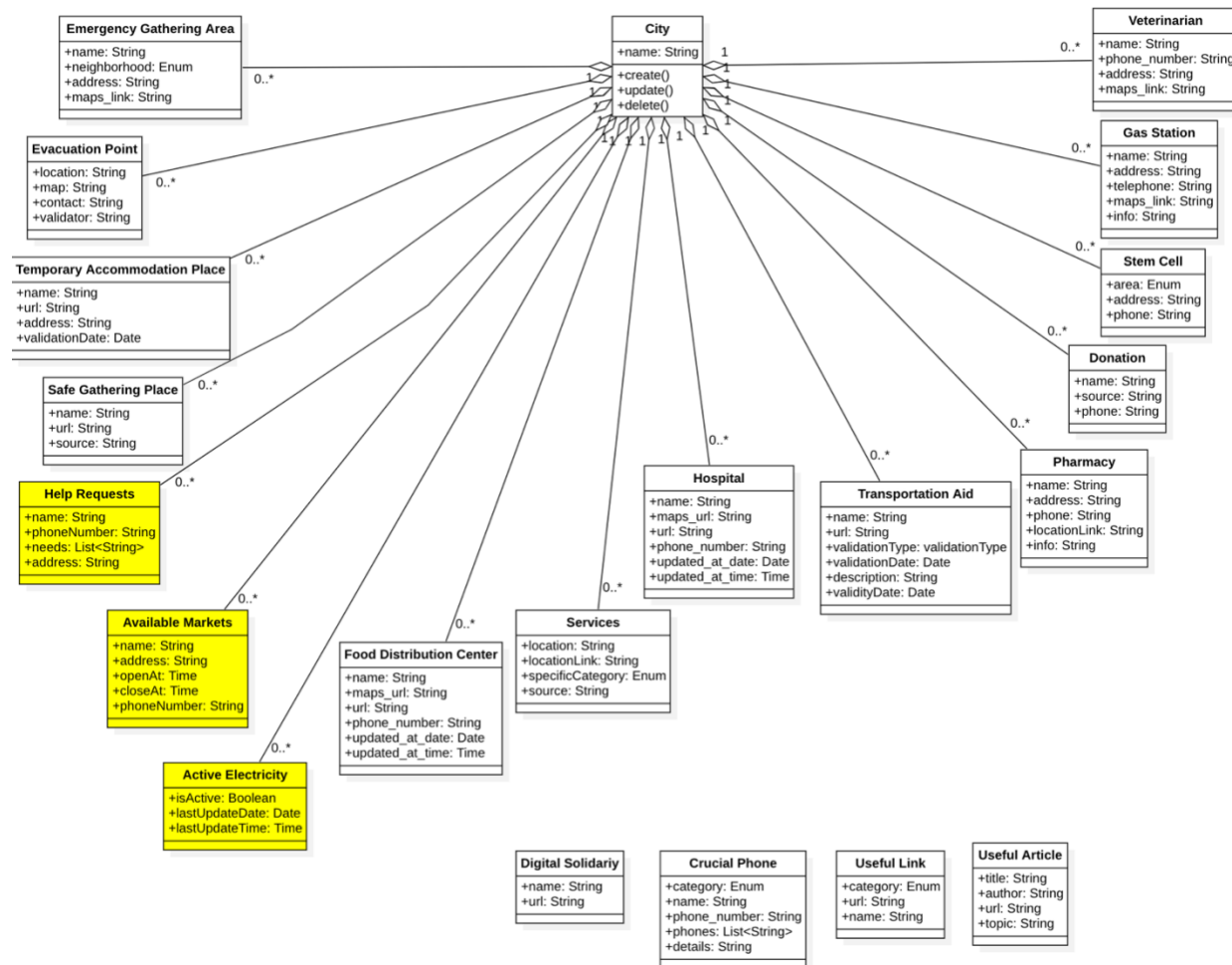


Figure 18: Database Class Diagram After Suggestions

5.3.3. Operations on Data

Operation	Description
create()	The create() operation in the City class is responsible for creating a new city entry in the database. It initializes a new city object with

	the provided data and saves it to the database, creating a new record for the city.
update()	The update() operation in the City class is used to modify the existing city information in the database.
delete()	The delete() operation in the City class is responsible for removing a city record from the database.

Table 2: Operations on Database Class Diagram After Suggestions

5.4. Deployment View

5.4.1. Stakeholders' uses of this view

The inclusion of the "Help Requests" section necessitates a reliable deployment environment capable of handling concurrent user interactions and storing and retrieving assistance request data securely. Data collectors and validators rely on a properly deployed system to manage and validate the information gathered through the "Request Help" section. The deployment view ensures that the system's infrastructure can handle the storage, retrieval, and processing of large amounts of data efficiently.

5.4.2. Deployment Diagram

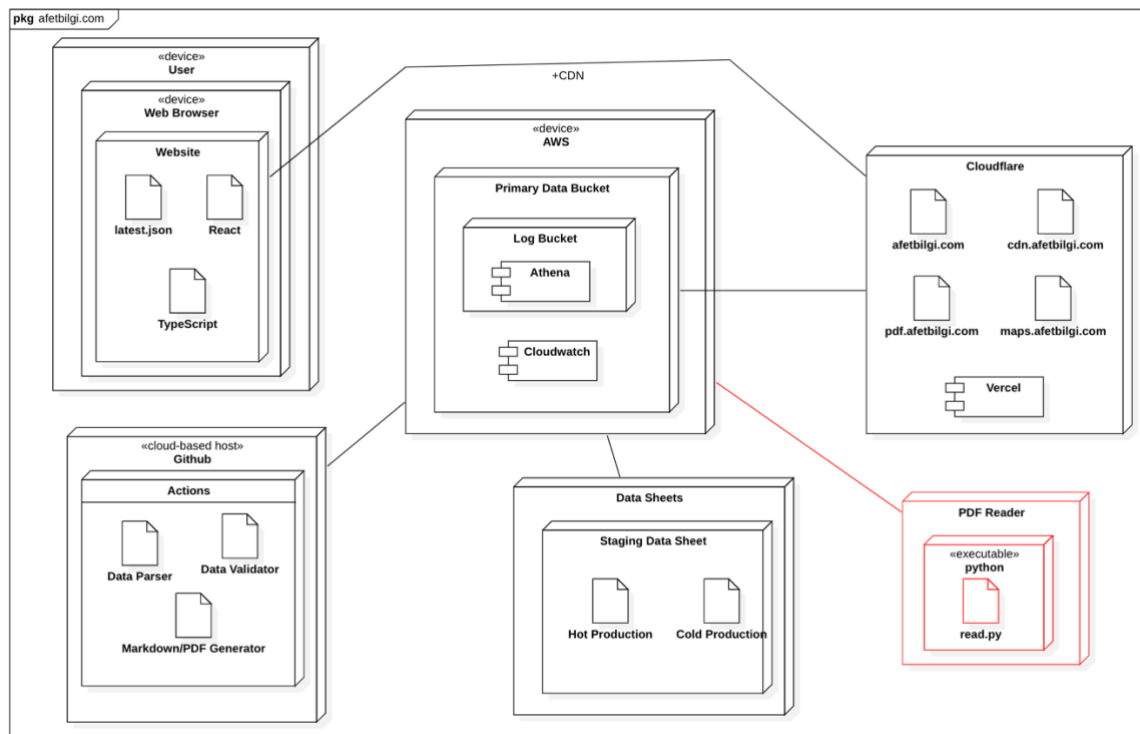


Figure 19: Deployment Diagram After Suggestions

The PDF Reader Node, along with the executable, enables the system to offer users the capability to read and interact with PDF files. When a user initiates an action to read a PDF document, the executable within the PDF Reader Node is triggered. It receives the information from the AWS, and play the voice.

5.5. Design Rationale

i) Context View

The inclusion of the PDF Reader component with the capability to enable users to listen to PDF files is motivated by the goal of enhancing accessibility and usability for individuals who may have visual impairments or prefer audio-based content consumption. This feature allows these users to access and consume the information contained within PDF files by listening to it instead of reading it visually.

ii) Functional View

The inclusion of the "Request Help" part is driven by the objective of facilitating effective communication and assistance between individuals in need and relevant authorities or aid organizations during natural disasters. The "Request Help" part allows users to input their phone number, address, and the type of assistance they require. Firstly, it provides a channel for individuals affected by a natural disaster to directly request help and support. By capturing essential information such as contact details and specific needs, the system becomes a centralized platform for collecting and organizing assistance requests. Secondly, the "Request Help" part enables data validators to verify and validate the information provided by users. This step ensures the accuracy and reliability of the help requests.

iii) Information View

The inclusion of the "isElectricityAvailable" is driven by the need to provide critical information to users regarding the availability of electricity in a specific region during or after a natural disaster. It can aid emergency response teams, government agencies, and utility providers in assessing the impact of the disaster on the power infrastructure and planning their response and recovery efforts accordingly.

iv) Deployment View

The PDF Voice Player component is deployed as a separate module, allowing for scalability and flexibility in terms of technology choices and integration with other system components.