
SOFTWARE ARCHITECTURE DESIGN

Afetbilgi.com

**Moustafa Ismail Hamed Mohamed Ismail - 2465813
Toygar Ateş - 2469161**

May 31, 2023

Contents

List of Figures	4
List of Tables	5
1 Introduction	7
1.1 Purpose and objectives of afetbilgi.com	7
1.2 Scope	7
1.3 Stakeholders and their concerns	8
2 References	9
3 Glossary	10
4 Architectural Views	11
4.1 Context View	11
4.1.1 Stakeholders' uses of this view	11
4.1.2 Context Diagram	11
4.1.3 External Interfaces	13
4.1.4 Interaction scenarios	13
4.2 Functional View	16
4.2.1 Stakeholders' uses of this view	16
4.2.2 Component Diagram	16
4.2.3 Internal Interfaces	18
4.2.4 Interaction Patterns	20
4.3 Information View	24
4.3.1 Stakeholders' uses of this view	24
4.3.2 Database Class Diagram	24
4.3.3 Operations on Data	25
4.4 Deployment View	26
4.4.1 Stakeholders' uses of this view	26
4.4.2 Deployment Diagram	26
4.5 Design Rationale	28
5 Architectural Views for Suggestions to Improve the Existing System	29
5.1 Context View	29
5.1.1 Stakeholders' uses of this view	29
5.1.2 Context Diagram	29
5.1.3 External Interfaces	31

5.1.4	Interaction scenarios	32
5.2	Functional View	33
5.2.1	Stakeholders' uses of this view	33
5.2.2	Component Diagram	33
5.2.3	Internal Interfaces	34
5.2.4	Interaction Patterns	36
5.3	Information View	37
5.3.1	Stakeholders' uses of this view	37
5.3.2	Database Class Diagram	37
5.3.3	Operations on Data	37
5.4	Deployment View	39
5.4.1	Stakeholders' uses of this view	39
5.4.2	Deployment Diagram	39
5.5	Design Rationale	41

List of Figures

4.1	System Context Diagram and Explanations	12
4.2	External Interfaces Diagram	13
4.3	Activity Diagram for Emergency Gathering Areas	14
4.4	Activity Diagram for Temporary Accomadation Places	15
4.5	Component Diagram for Afetbilgi	16
4.6	Class Diagram for DataInterface	18
4.7	Class Diagram for PDFInterface	19
4.8	Class Diagram for HTMLDeneratorInterface	19
4.9	Class Diagram for MapInterface	20
4.10	Sequence Diagram Getting Active Hospitals	21
4.11	Sequence Diagram Getting City Map	22
4.12	Sequence Diagram for Donations	22
4.13	Sequence Diagram for City PDF	23
4.14	Database Class Diagram	24
4.15	Deployment Diagram for Afetbilgi	27
5.1	System Context Diagram and Explanation Improvements	30
5.2	External Interfaces Class Diagram Improvements	31
5.3	Activity Diagram for Select Temporary Places Improvements	32
5.4	Component Diagram for Improved Afetbilgi	33
5.5	Class Diagram for AuthenticationInterface	35
5.6	Class Diagram for AuthorizationInterface	35
5.7	Sequence Diagram For DataItem Modification	36
5.8	Database Class Diagram	37
5.9	Deployment Diagram for Improved Afetbilgi	40

List of Tables

3.1	Glossary	10
4.1	Database Class Diagram Operation Descriptions	25
5.1	Database Class Diagram Improvements Operation Descriptions	38

Revision History

Revision	Date	Author(s)	Description
1.0.0	10.05.2023	M. , T.	SAD First Draft
1.1.0	30.05.2023	M. , T.	SAD Final

1 Introduction

1.1 Purpose and objectives of afetbilgi.com

The main goal of the system is help to people who are affected by the Pazarcik earthquake. The development of Afetbilgi.com project started shortly after the biggest influential disaster in the Turkish Republic for providing easy access to up-to-date information needed. The type of information offered by the web site is related to all facilities that may be needed by individuals affected from the earthquake. The website has also official donation addresses for those who want to help individually. The project offers both online resources and offline PDFs for all current information. It is worth mentioning that while the city infrastructures and roads were unrecognizable, Afetbilgi.com had the foresight include a map software that allow people to see facilities and key locations near to their current location.

1.2 Scope

The name of the project is Afetbilgi which is a combination of the two words disaster and information in the scope of this system, the website should benefit affected individuals and help in enhancing their life quality and chances to live during dangerous times.

The project is created for the sake of charity and its main goal of the system is to offer help to impacted individuals, through collecting information from various resources and delivering it to those who need it. The information show on Afetbilgi will be available only for cities affected by the earthquake in Turkey, and other countries like Syria will not be included.

The platform allows users to find the location of help places but cannot facilitate getting help through the platform. To make all necessary information available to people affected by the earthquake and people who want to help, the project requires a web based system for accessing data, a data collection and validation system, and a database and database parser.

To summaries, the scope of the Afetbilgi is to:

- The platform shall collect information from users and various other sources and present information in four different languages for people in need.
- The system requires a deployment system, a data collection and validation system, a database, and database parser.
- The platform will enable users to see all information on a map.
- The users shall be able to a PDF of all information.
- The goal is to make all necessary information available to people affected by the earthquake and people who want to help.

1.3 Stakeholders and their concerns

There are several stakeholders in the Afetbilgi project. These are as follows:

- Impacted individuals**

Impacted individuals are the people who are affected by natural disasters. These people are concerned with using Afetbilgi to get access to information that would help them. Their access to information shall be as easy and as efficient as possible. The SAD may be of importance to impacted individuals to help them understand how the system works to satisfy their needs.

- Volunteers**

Volunteers are individuals who are eager to extend a hand. They can have a profession in a particular field, for instance, digger operators or medical doctors which can play a crucial role post-disaster. They may be desired to help personally to others and make a positive impact on professional societies. The SAD may help them understand how to utilize the system in the best possible manner.

- Data Collectors and Validators**

They are gathering and verifying information about emergency situations and phone numbers. They can communicate with nongovernmental organizations, government agencies, or other societies involved in disaster struggles. They can use various ways such as interviews to collect data and validate its accuracy. The need SAD to understand the interfaces of the system such that they can use them to modify and validate the data existing in the system.

- Programmers**

Programmers are responsible for developing and maintaining Afetbilgi software system. They would use SAD to understand the architectural design of the system so that they can develop it efficiently. The SAD would also serve as a guide for developers whose role is the maintain the software after the development phase.

- Testers**

Testers are individuals responsible for creating tests for the system. They would use the SAD document to create effective tests for the system.

2 References

This document is written according to the specifications written in the document below:

Rozanki, Woods. (2022). A Viewpoint Catalog (2nd ed.). Addison Wesley.

Alpaylan. (2023). Alpaylan/Afetbilgi.com. GitHub. Retrieved May 16, 2023, from <https://github.com/alpaylan/Afetbilgi.com>

Afetbilgi. (2023, February). Retrieved May 16, 2023, from <https://www.Afetbilgi.com/>

3 Glossary

Term	Definition
API	Application Programming Interface.
AWS	Amazon Web Services.
Data Collectors and Validators	People working on collecting and validating data.
Facilities	Any intuitions that offers help to impacted individual such as AFAD.
GitHub	Web-based version control and collaboration platform for software developers.
HTTP/HTTPS	Hypertext Transfer Protocol/ Hypertext Transfer Protocol Secure.
Impacted Individuals	People impacted by a natural crisis
Volunteers	People offering help to impacted individuals
Database	It's a special software to store and maintain data.
Vercel	It's cloud platform for hosting website applications.
Cloudflare	It's a global networks that provides content delivery.

Table 3.1: Glossary

4 Architectural Views

4.1 Context View

4.1.1 Stakeholders' uses of this view

This point of view can provide an overall view of the whole system for all the stakeholders including impacted individual, developers, volunteers and data collector and validators. This also encompasses the system's influence on its surroundings and the utilization of external entities and services from a contextual perspective.

4.1.2 Context Diagram

This context diagram shows all external entities that may interact with the Afetbilgi system.

- **Impacted Individual:** They enters their city, language neighbourhood location and use the website for reaching many needs. For instance, gathering areas, accomadation places, transportation aid, active hospitals and open pharmacies.
- **Software Developers:** They develop software solutions and updates website properly.
- **Data collectors and Validators:** They take the unverified data from Afetbilgi.com and give back the data authorized or invalidated.
- **Volunteers:** They gathers help direcly for disaster area or Afetbilgi includes donation links to who want to help financially.
- **Google maps:** Provides visualized map locations.
- **AWS:** Provides storage and networking for project.

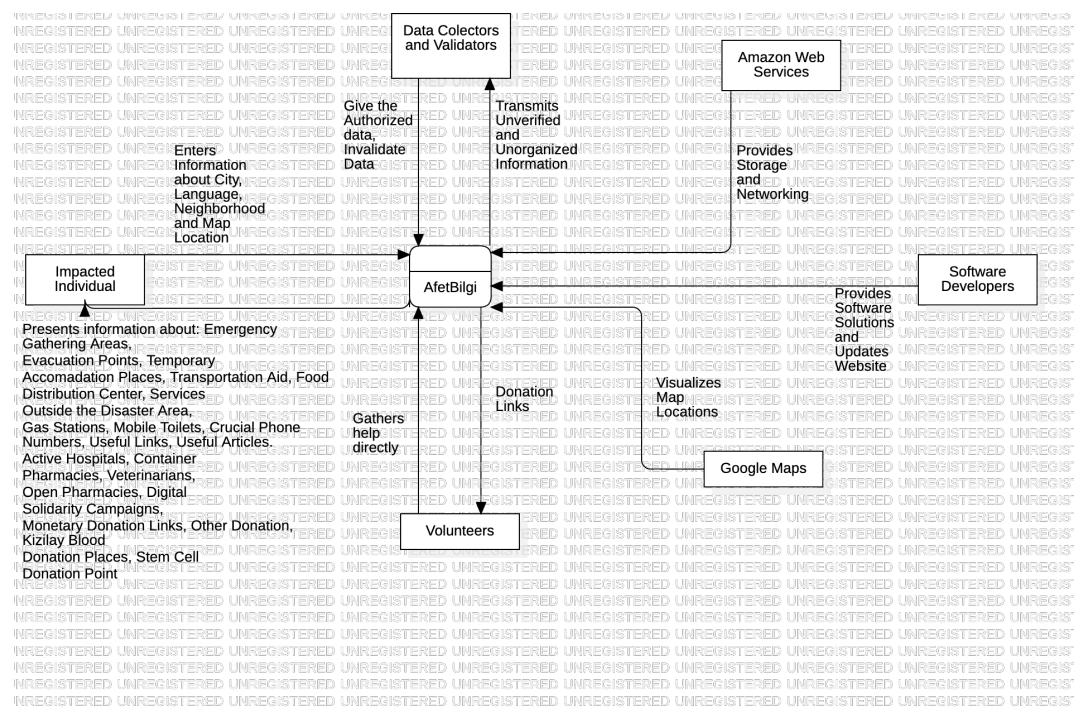


Figure 4.1: System Context Diagram and Explanations

4.1.3 External Interfaces

The external interfaces are the interfaces provided by Aftetbilgi for various entities external to the system. There 3 significant external interfaces which will be explained in this document.

Volunteers External Interface:

- This interface is used to get the data from website after choosing country, city and the language of the needed situation. Try to provide communication between impacted individuals and helpers.

Impacted Individuals External Interface:

- This interface is used by disease impacted people with giving their information about city, language, country and try to reach best possible options for help.

Collectors and Validators External Interface:

- This interface is used by Collectors and Validators to get invalidated data from the system, add data to the system, and update existing data in the system.

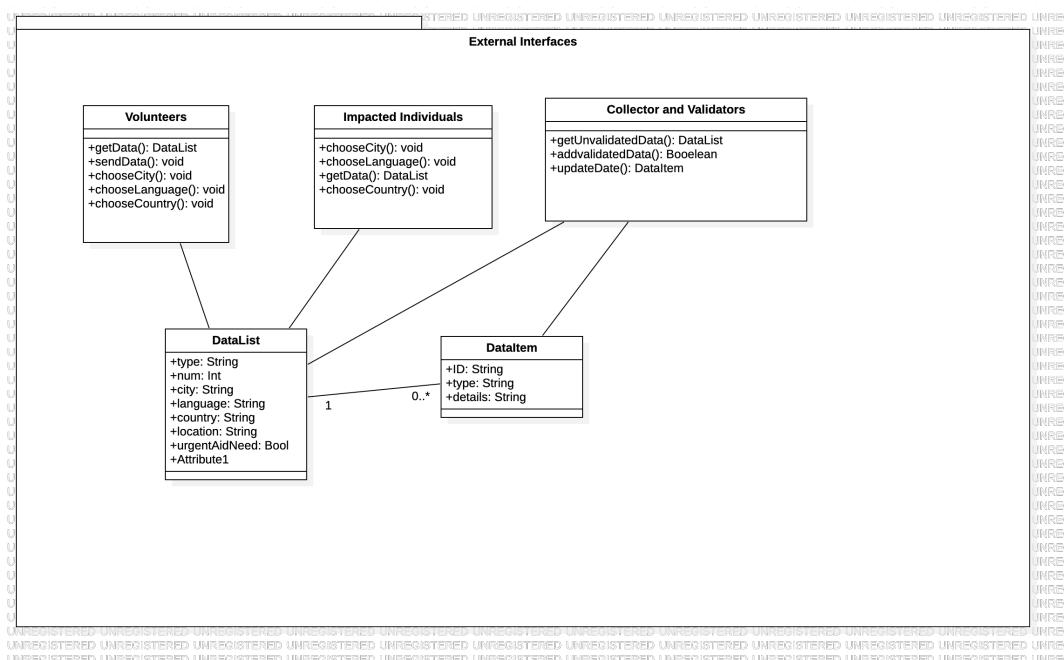


Figure 4.2: External Interfaces Diagram

4.1.4 Interaction scenarios

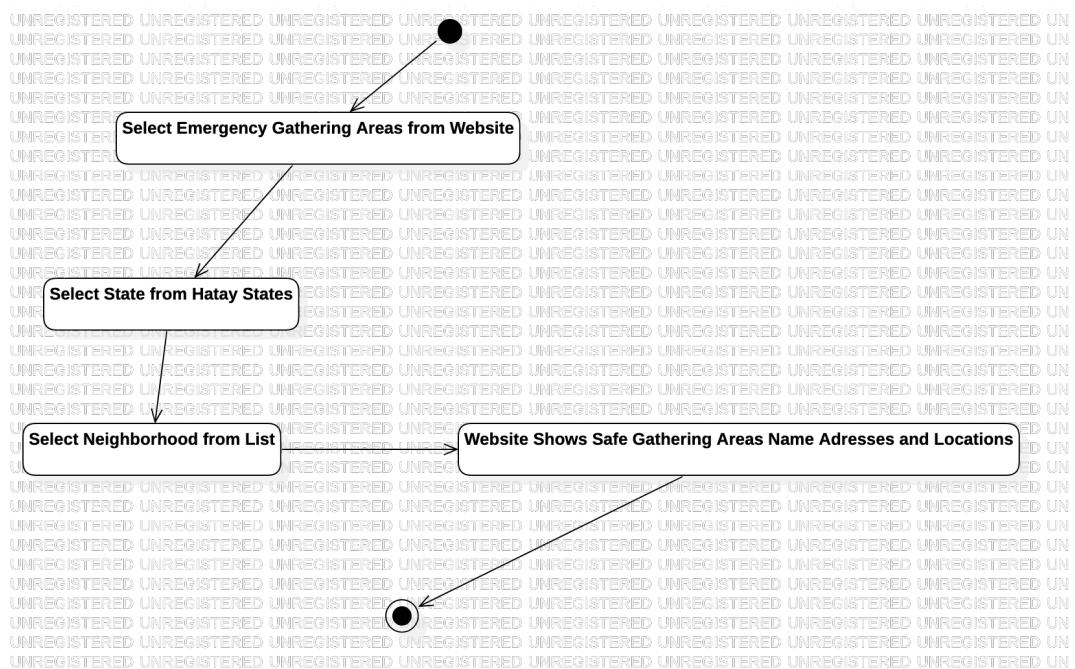


Figure 4.3: Activity Diagram for Emergency Gathering Areas

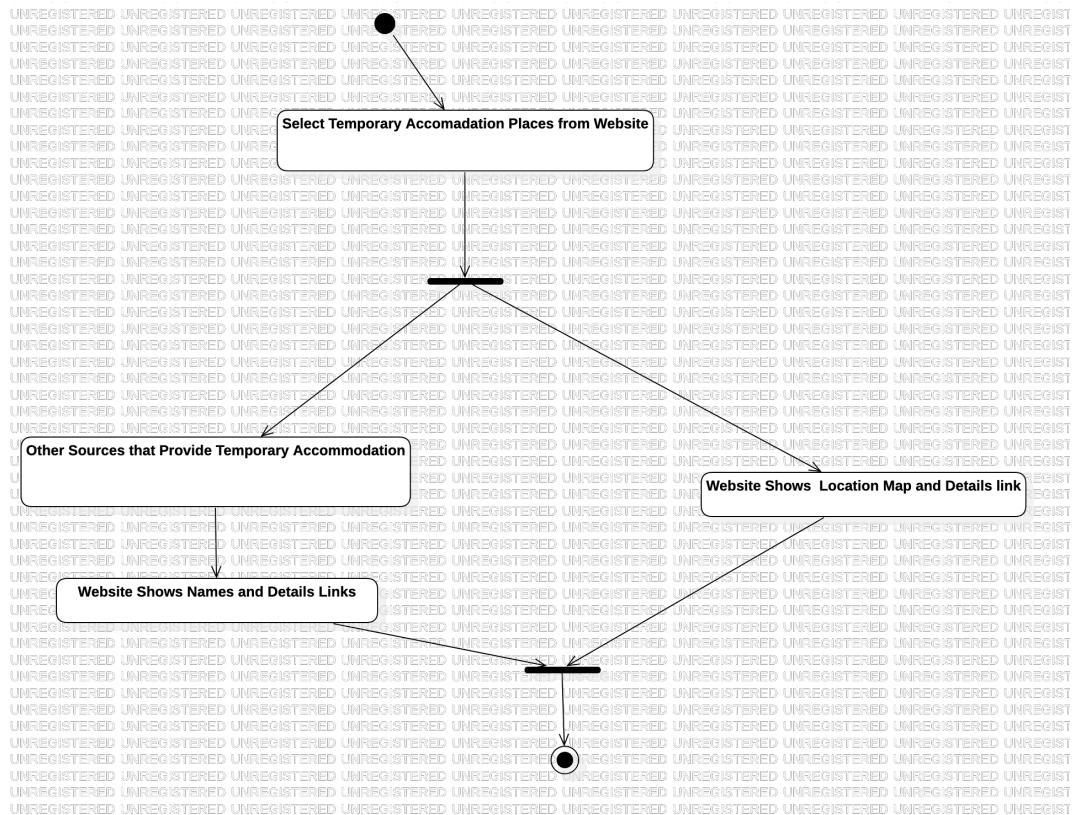


Figure 4.4: Activity Diagram for Temporary Accommodation Places

4.2 Functional View

4.2.1 Stakeholders' uses of this view

The Functional view provides various stakeholders with useful insights about the system that would the way in which they intend to interact with the system. For Afetbilgi, there are mainly several stakeholders who would benefit from the functional view of the system as follows:

- Developers: they would use the functional view to gain insight about the components of the system and their interaction. It would help them develop components that satisfies the required functionality.
- Data collectors and Validators: the functional view would help them understand the functionalities and interfaces which would help them in interacting with the system.
- Testers: they would use the functional view of the system to understand the functionality required by the system and how it is satisfied which would help the prepare effective tests for the system.
- Volunteers: functional view would help them understand the functionalities provided by the system which would allow them to use the system effectively.

4.2.2 Component Diagram

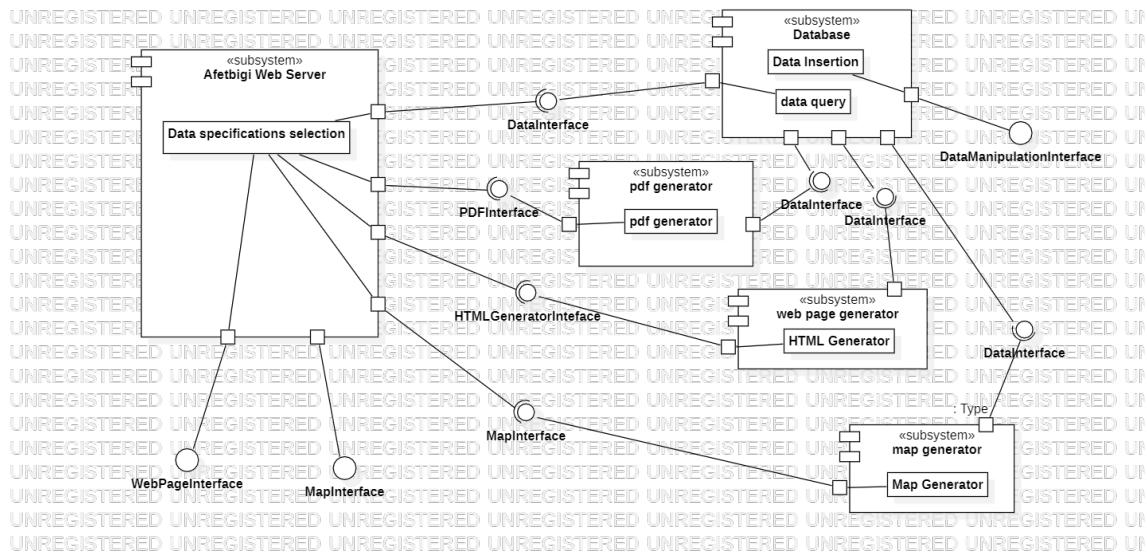


Figure 4.5: Component Diagram for Afetbilgi

The component diagram shows various components of the system and the interaction between them. Afetbilgi consists of 5 main subsystems that provides all of the functionality of the system. The details of each subsystem and how it interacts with other subsystem is as follows:

Afetbilgi web server:

- Delivers HTML-based interface for affected people and volunteers from which they can send various request to the system
- Respond to various requests by affected people and volunteers by constructing the appropriate response.
- Interacts with all parts of the system while create the responses for various requests.
- Deals with interfaces provided by other components to get the required data.

Database:

- It stores and manages all the data of the system
- Offers a DataInterface for other components of the system from which they can query and get the required data.
- It offers a DataManipulationInterface interface for Data collectors and validators from which they can insert new data into the system and validate or invalidate the existing data
- It shall offer quality features such as data Persistence and recover.
- It shall also offer data security and integrity features by limiting the ability of various components to manipulate existing data.

PDF Generator:

- It offers PDFInterface for other components that would allow them to generate a PDF containing specific data provided by other components.
- It would generate PDF document with the required data and specification.

Map Generator:

- It provides MapInterface for other components that would allow them to get a map with the required filters and specifications.
- It would generate Map with specification related to type and amount of data put on the map.

Web Page Generator:

- It offers HTMLGeneratorInterface for other components that would allow them to get a HTML with the required data presented in the HTML document.
- It would generate HTML pages with the provided data and specifications.

4.2.3 Internal Interfaces

The internal interfaces are interfaces between various components of the system. Afetbilig contains 4 significant internal interfaces which are described and specified in this section in details.

DataInterface

- It's the main interface exposed by the database for various components to interact with the database.
- It has functions to query and get all the data needed from the database.

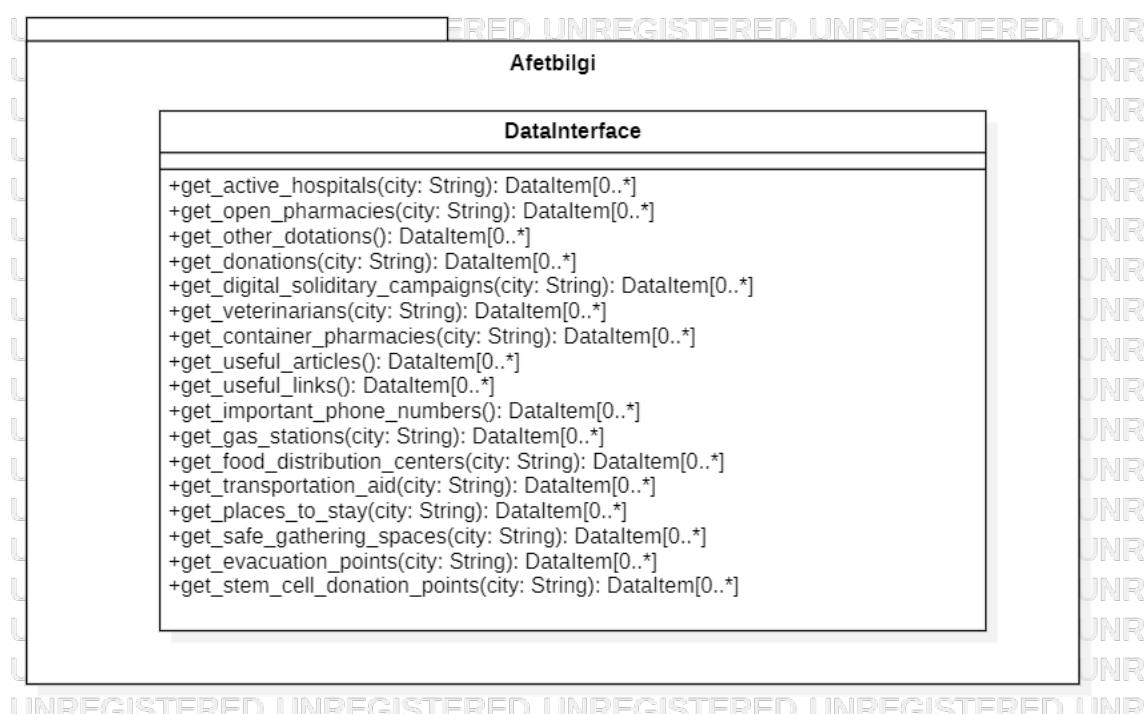


Figure 4.6: Class Diagram for DataInterface

PDFInterface

- It's the main interface for the PDF generator module that expose all of its functionalities.
- It's used to create a PDF document for various data stored in the database.

HTMLGeneratorInterface

- This interface is used show the functionalities that is used to generate HTML documents that shows various data in the database.

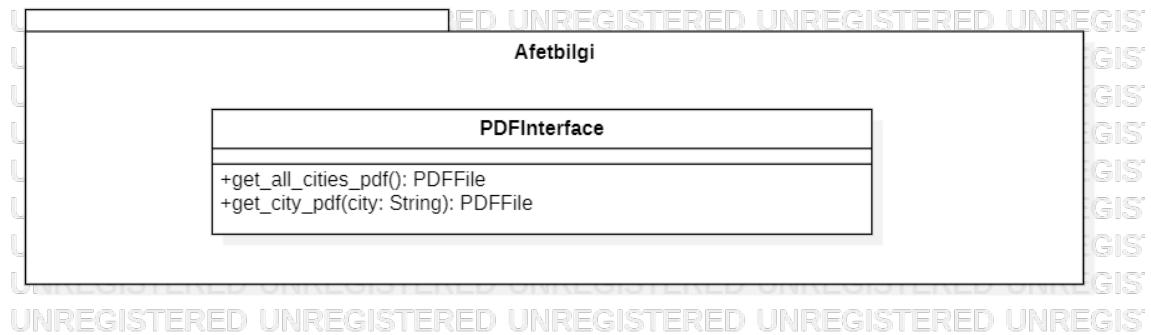


Figure 4.7: Class Diagram for PDFInterface

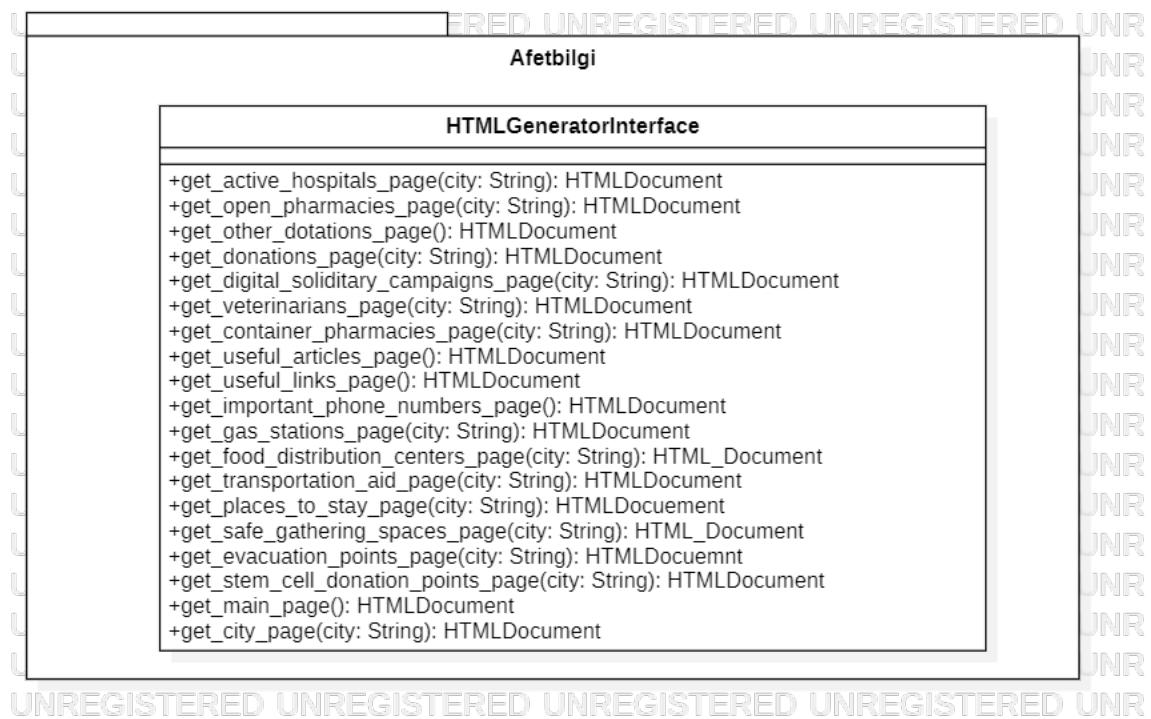


Figure 4.8: Class Diagram for HTMLDeneratorInterface

MapInterface

- This interface exposed the utilities offered by the MapGenerator module.
- It's used get a Map object with various specifications based through this interface.

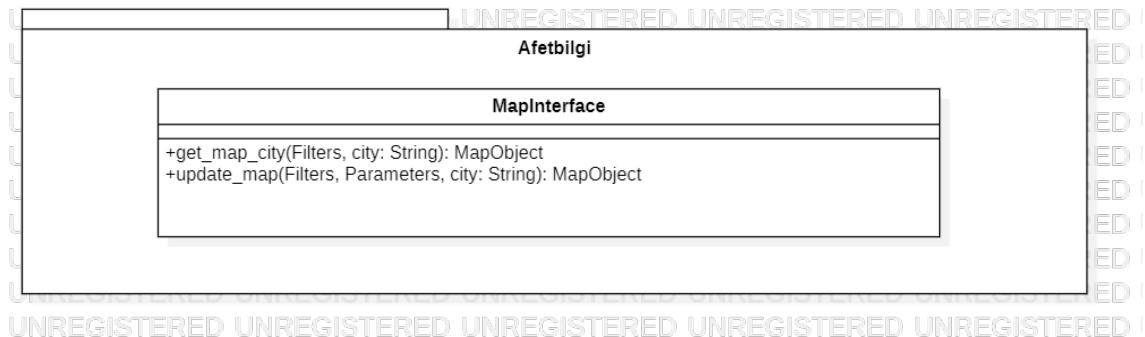


Figure 4.9: Class Diagram for MapInterface

4.2.4 Interaction Patterns

The interaction patterns show how the various interactions would take place between various entities in the system of Afetbilgi. There are 4 Interaction patterns that we have focused on in this document.

Getting Active Hospitals

This interaction shows how various components of the system would work together to deliver information about active hospitals in a city

Getting City Map

This interaction shows how various components of the system would work together to show a map of a city with various information on the Map

Getting Donations

This interaction shows how various components of the system would work together to show information about donation institutions

Getting City PDF

This interaction shows how various components of the system would work together to deliver a PDF containing information about a specific city

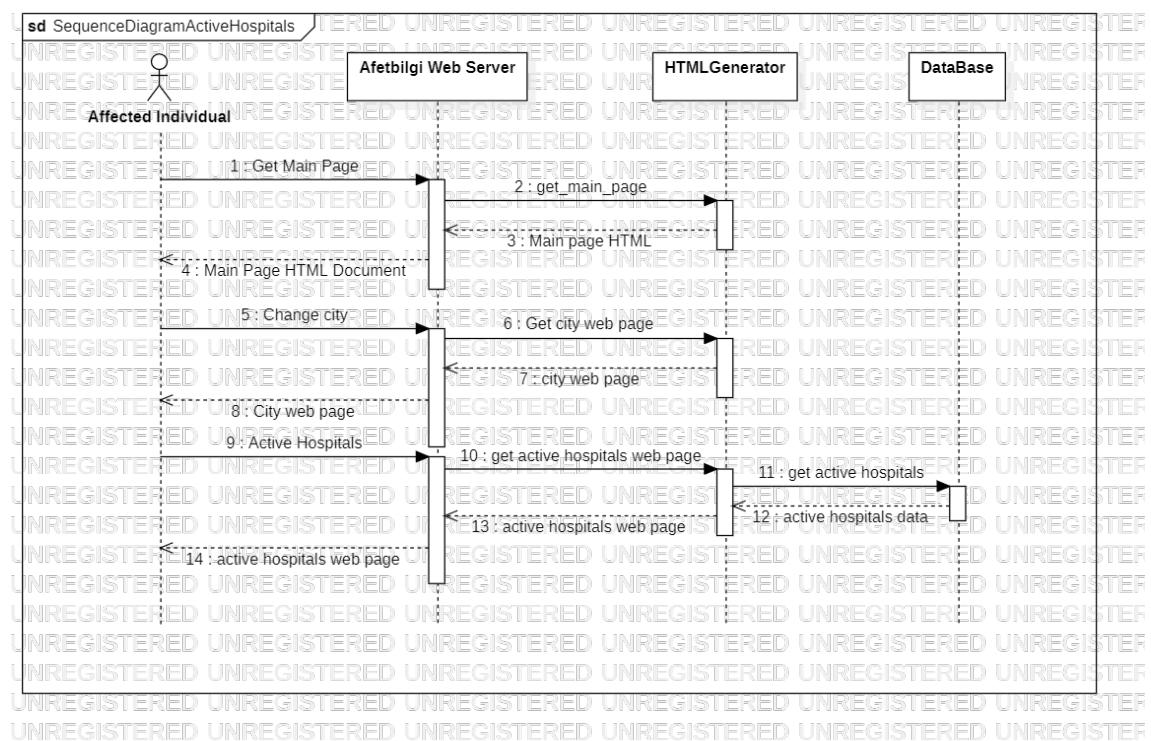


Figure 4.10: Sequence Diagram Getting Active Hospitals

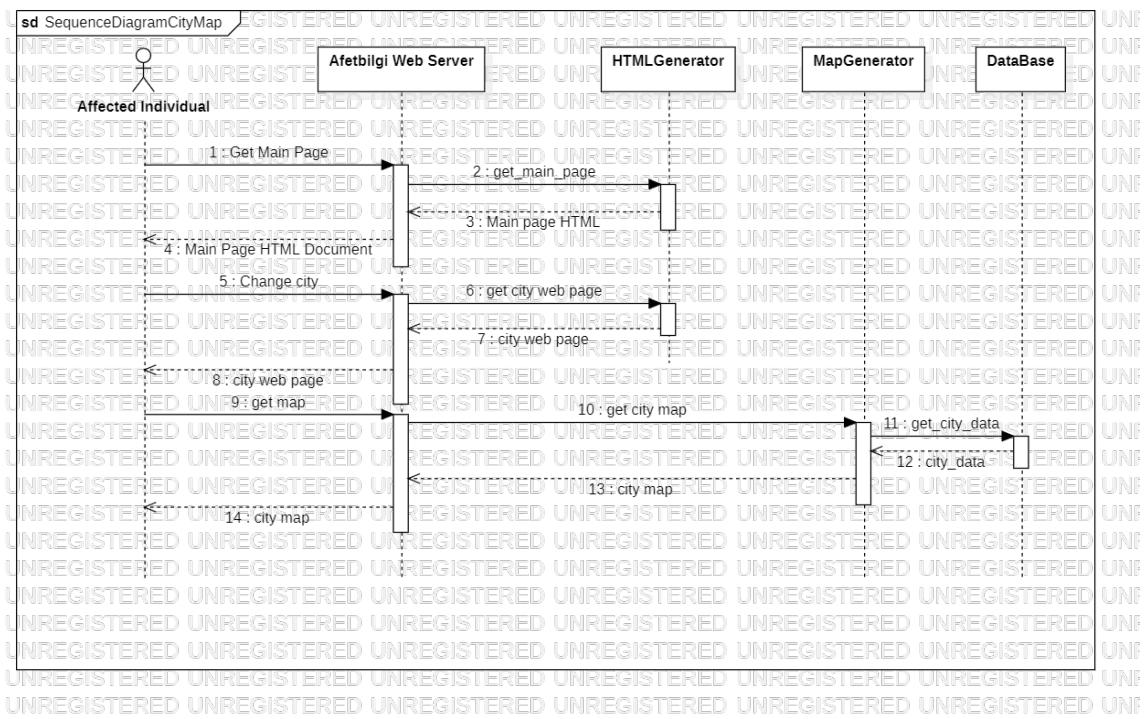


Figure 4.11: Sequence Diagram Getting City Map

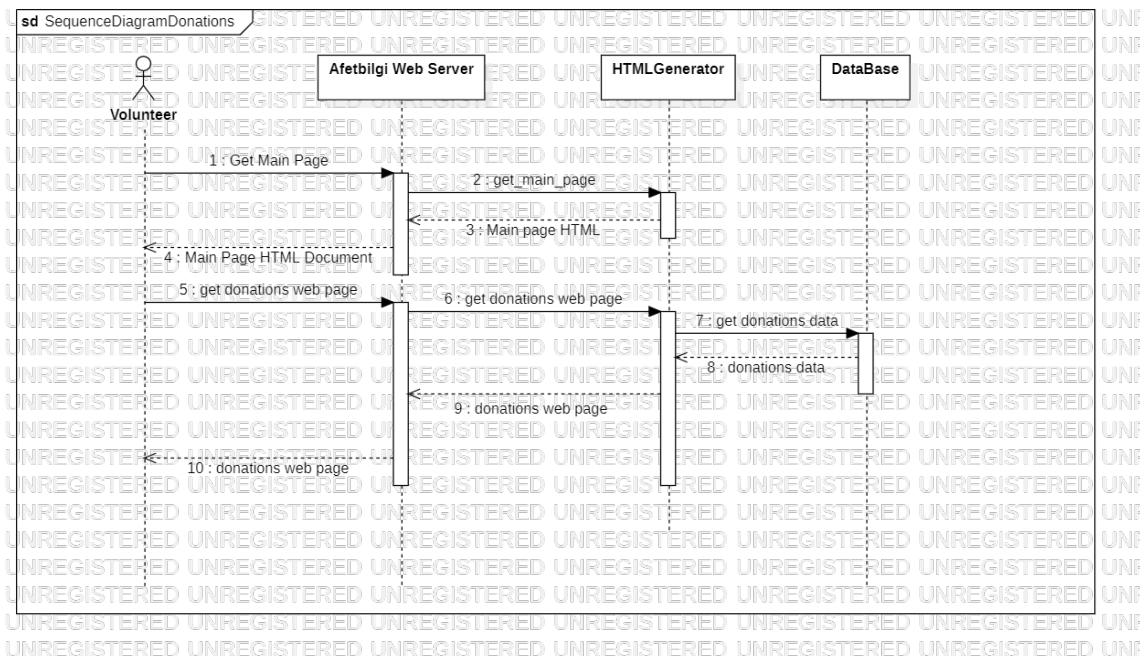


Figure 4.12: Sequence Diagram for Donations

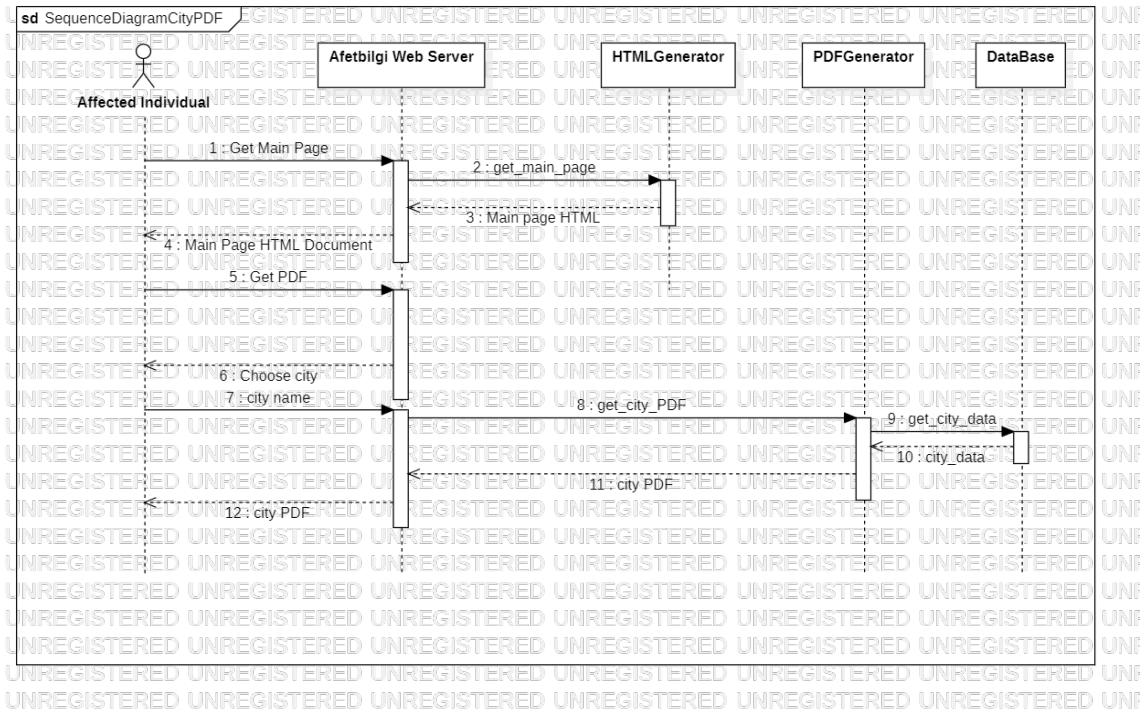


Figure 4.13: Sequence Diagram for City PDF

4.3 Information View

4.3.1 Stakeholders' uses of this view

In this point of view, various features of the Afetbilgi.com, main page interfaces with the system, features and relations will be shown. This view is crucial for giving idea to stakeholders for the Afetbilgi.com. Hence it exhibits quality of features and limits for the website.

4.3.2 Database Class Diagram

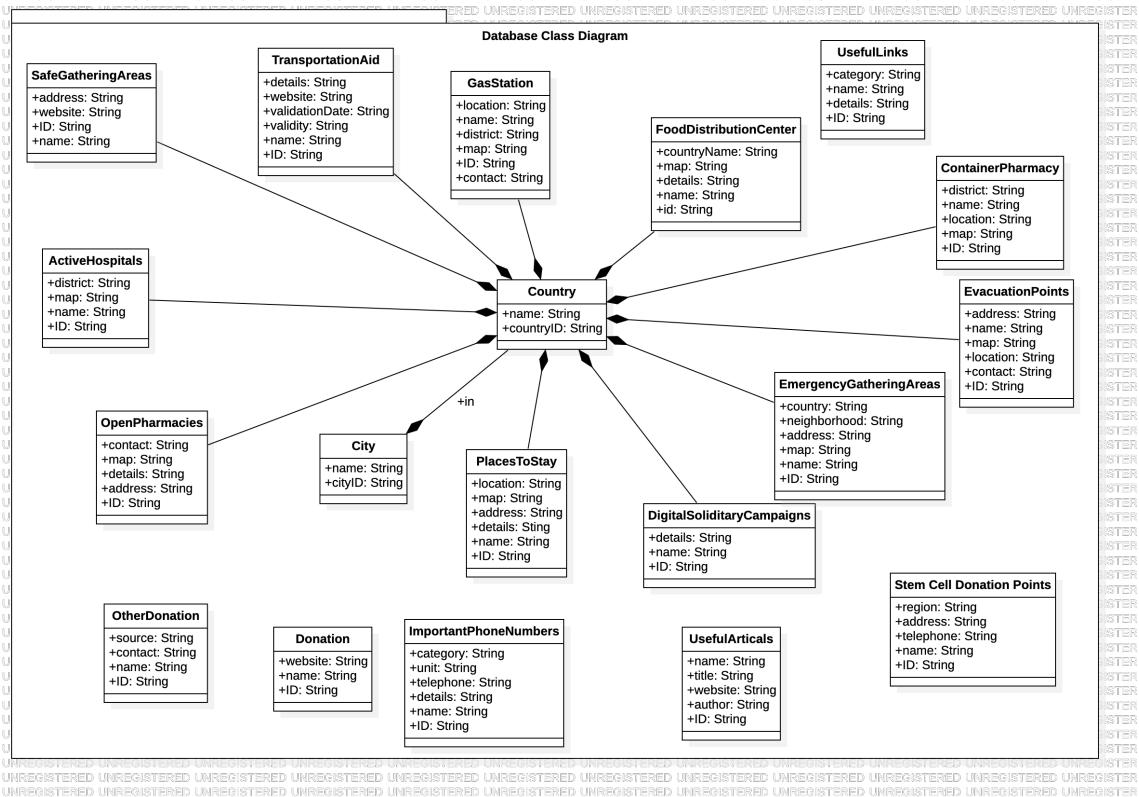


Figure 4.14: Database Class Diagram

4.3.3 Operations on Data

Operations on data are shown in table 4.1.

Function	Summary
City	Lets the user choose a city from Turkey, sorted with the emergency arrangement.
EmergencyGatheringAreas	Asks the user for the district and then asks for the neighborhood, providing an address name and a map location link.
SafeGatheringPlaces	Asks the user for the city and provides safe places for a convention.
EvacuationPoints	Asks the user for the city and provides evacuation areas.
TransportationAid	Informs the user about transportation facilities, details, website, verification dates, and end date.
FoodDistributionCenter	Asks the user for the city and state, and provides places for free food. It also includes quick links for holiday and special websites that provide places to stay in other cities.
GasStations	Asks the user for the city and state, and shows the address, map location, and phone numbers of gas stations.
ImportantPhoneNumber	Provides categories, units, phone numbers, and details for help.
UsefulLinks	Provides different categories, names, and detailed links to help-oriented websites.
UsefulArticles	Provides headers, writers, and website links for beneficial texts.
ActiveHospitals	Asks the user for the city, and provides hospital addresses, shows active hospitals, and gives detailed links.
OpenPharmacies	Asks the user for the city and state, and provides the name, address, map location, and phone number of open pharmacies.

Table 4.1: Database Class Diagram Operation Descriptions

4.4 Deployment View

4.4.1 Stakeholders' uses of this view

The Deployment view provides various stakeholders with useful insights about the system components, execution environment for each components, and how these components would be interacting with each other. For Afetbilgi, there are mainly several stakeholders who would benefit from the Deployment view of the system as follows:

- Developers: they would use the deployment view to gain insight about the components of the system, execution environments, and communication protocol between various components.
- Testers: they would use the deployment view of the system to test whether the developed modules would run efficiently on the deployment environment
- System Administrators: they would use this view to manage and maintain the deployed system.

4.4.2 Deployment Diagram

The deployment diagram shows the physical architecture of a system. It illustrates how various components of Afetbligi are distributed on various hardware and how they are connected. There are various servers that hosts various software components of Afetbilgi as follows:

- AWS : Amazon web server is a could computing services provided by amazon. It's used to deploy several components of Afetbligi.
- Vercel : It's a cloud platform that is used to host static content of the website such as the picture and HTML documents.
- Cloudflare : It's a cloud platform that is used to increase the performance by caching the Aftebilgi website and responding to various request.

There are various software components that is deployed on various cloud platforms as follows:

- database : It's set of files that is used to keep all the information of Afetbilgi. It's deployed on AWS.
- backup bucket : It's used to backup the whole system and is deployed on AWS
- Aftebilgi Web App : It's a set of scripts that forms the core of Aftebilgi. The scripts include data_validator_script, data_parser_script, pdf_generator_script, map_handler_script. The set of scripts is deployed on AWS.
- Static website : It's a set of HTML documents and pictures that forms the frontend of the Aftebilgi website. They are deployed on vercel cloud platform.
- web server : It's the main component of Afetbligi that responds to various requests by making calls when necessary to the appropriate component of Afetbilgi to form the response.

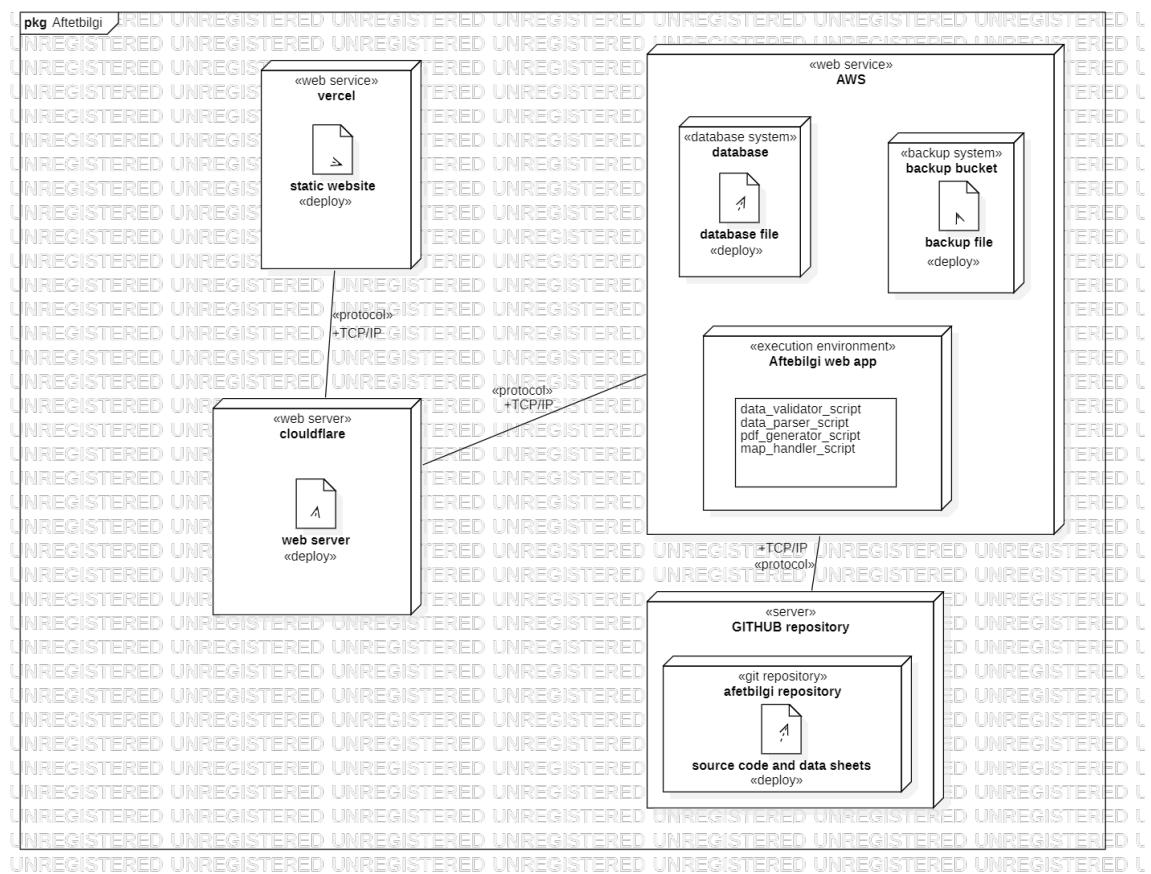


Figure 4.15: Deployment Diagram for Afetbilgi

4.5 Design Rationale

1. **Context View Design Rationale:** The main logic behind the Context View is to show website relationships between impacted individual and the Afterbilgi.com. For elaborating this system context diagram, class diagram for external interfaces and two different activity diagram have used. This point of view ensures the elementary way to understand clearly the relations for stakeholders.
2. **Functional View Design Rationale:** The main logic behind the Functional View is describe the functional operations performed on the website. We seperated all different functions which performed on the simple interface and provided them to process part by part. On the design phase of exhibiting them we have used component diagram, internal interfaces class diagram and three different sequence diagrams.
3. **Information View Design Rationale:** The main logic behind the Information View is the providing full access of required information without storing any data. For this style design phase, Amazon Web Services usage contributed significantly. We have exhibited the system with database class diagram and table.
4. **Deployment View Design Rationale:** The main logic behind the Deployment View is the elaborately clarify the software distribution on the Afetbilgi.com. In the Git-hub page of the project, they have explained the infrastructure with using a clear diagram. External servers and databases has a crucial role in the website. We have used deployment diagram to express the deployment view.

5 Architectural Views for Suggestions to Improve the Existing System

5.1 Context View

5.1.1 Stakeholders' uses of this view

This viewpoint for suggestions by taking into account the system's influence on its environment, this view also highlights the interactions with external entities and services. I suggest that a more comprehensive understanding of the system could aid in identifying opportunities for enhancement, thereby improving the system's effectiveness and reach. This improvement includes the communication, increased understanding, and more effective collaboration among volunteers.

5.1.2 Context Diagram

This context diagram shows all external entities that may interact with the Afetbilgi system.

- **Impacted Individual:** They enter their city, language neighbourhood location and use the website for reaching many needs. For instance, gathering areas, accommodation places, transportation aid, active hospitals and open pharmacies.
- **Software Developers:** They develop software solutions and update website properly.
- **Data collectors and Validators:** They take the unverified data from Afetbilgi.com and give back the data authorized or invalidated.
- **Volunteers:** They gather help directly for disaster area or Afetbilgi includes donation links to who want to help financially.
- **Google maps:** Provides visualized map locations.
- **AWS:** Provides storage and networking for project.
- **Getir and Yemeksepeti:** Sends food for required locations.
- **UrgentAidTeams:** Sends team to the exact urgent locations.

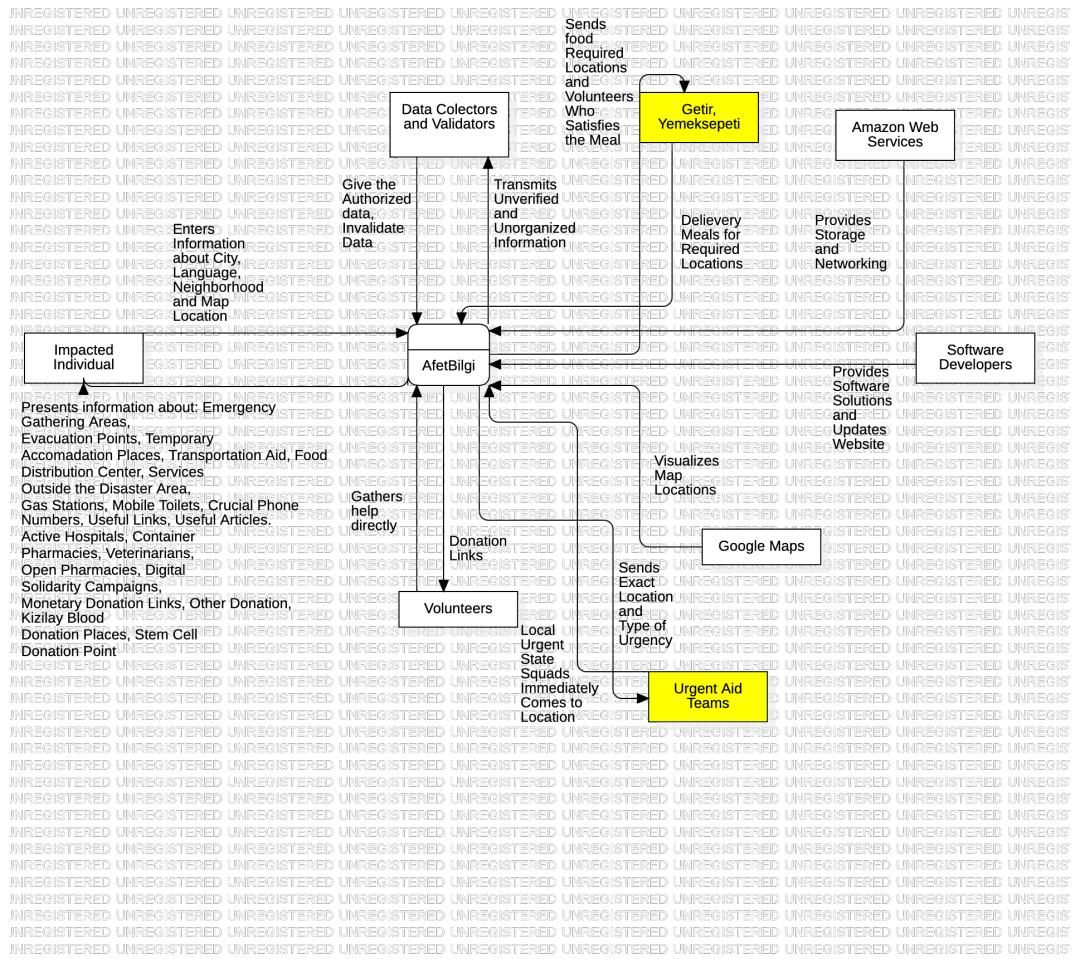


Figure 5.1: System Context Diagram and Explanation Improvements

5.1.3 External Interfaces

Volunteers External Interface:

- This interface is used to get the data from website after choosing country, city and the language of the needed situation. Try to provide communication between impacted individuals and helpers.

Impacted Individuals External Interface:

- This interface is used by disease impacted people with giving their information about city, language, country and try to reach best possible options for help. For improvements they can apply a meal or urgentaid requests and can give a review.

Collectors and Validators External Interface:

- This interface is used by Collectors and Validators to get invalidated data from the system, add data to the system, and update existing data in the system. For improvements they can log-in and log-out the system.

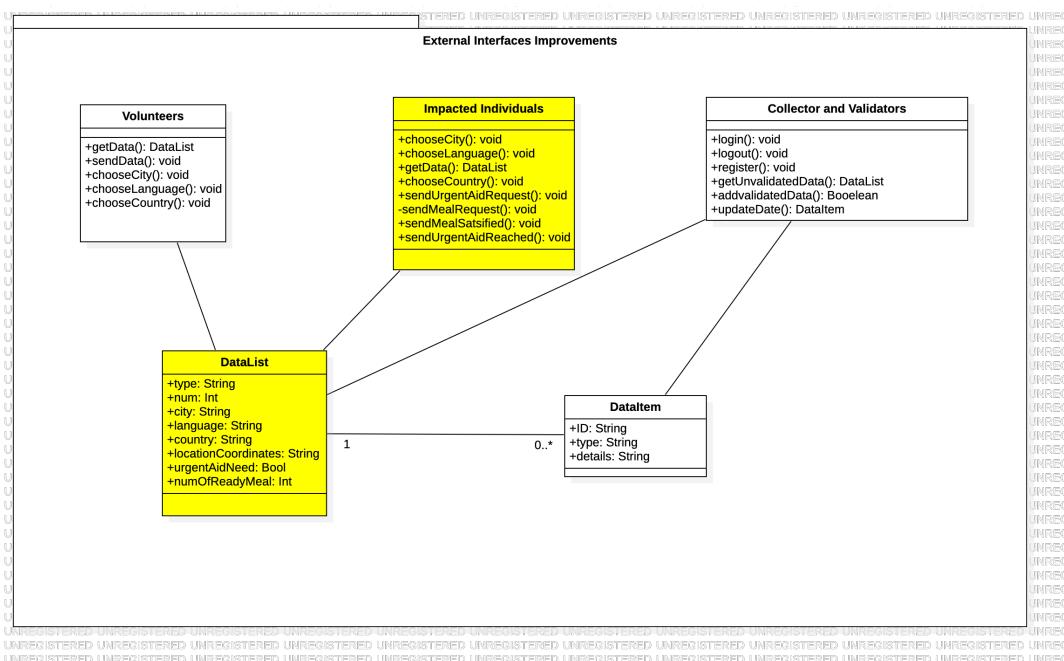


Figure 5.2: External Interfaces Class Diagram Improvements

5.1.4 Interaction scenarios

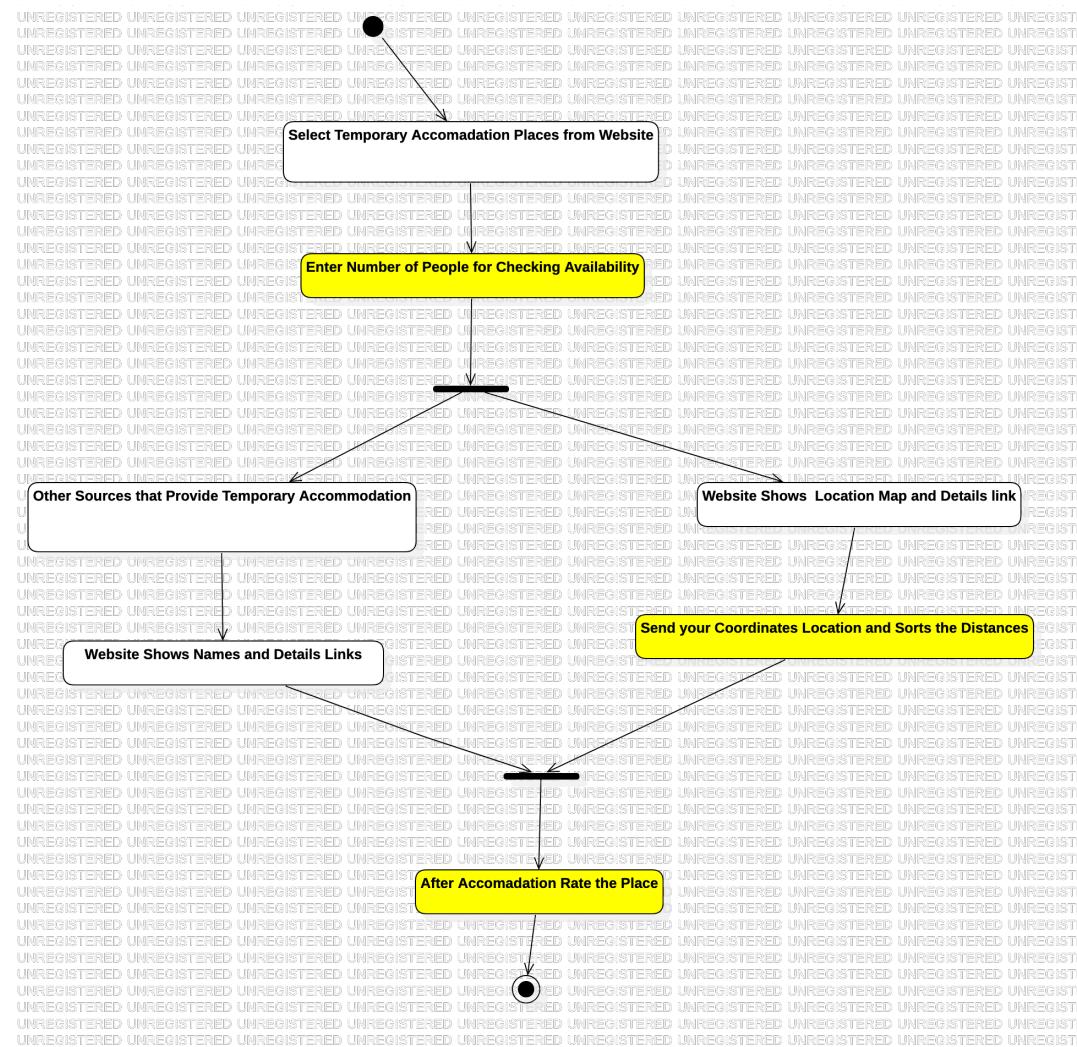


Figure 5.3: Activity Diagram for Select Temporary Places Improvements

5.2 Functional View

5.2.1 Stakeholders' uses of this view

The Functional view provides various stakeholders with useful insights about the system that would the way in which they intend to interact with the system. For Afetbilig, there are mainly several stakeholders who would benefit from the functional view of the system as follows:

- Developers: they would use the functional view to gain insight about the components of the system and their interaction.
- Data collectors and Validators: the functional view would help them understand the functionalities. It would help them deal with the system in a more infromed way.
- Testers: they would use the functional view of the system prepare proper tests for the various functionalities of the system.
- Volunteers: functional view would help them understand the functionalities provided by the system.
- Facilities: they would use this view to understand the functionalities of the system that would allow them to validate various pieces of information.

5.2.2 Component Diagram

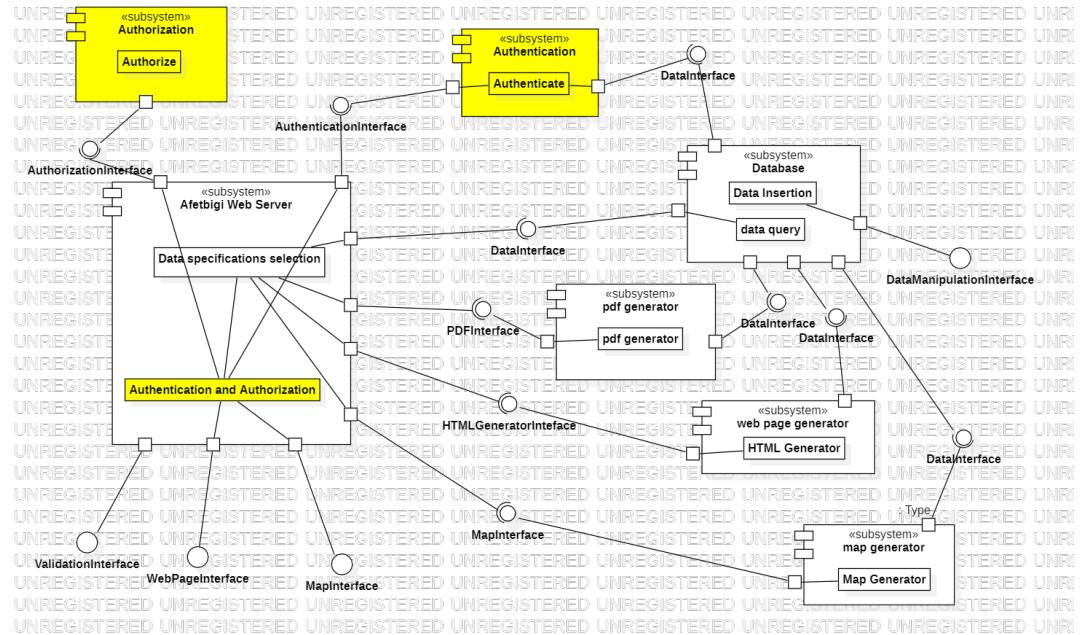


Figure 5.4: Component Diagram for Improved Afetbilgi

The component diagram shows various components of the system and the interaction between them. Improved Aftebilgi consists of 7 main subsystems that provides all of the functionality of the system. The details of new subsystem and how it interacts with other subsystem is as follows:

Authentication:

- It's used to authenticate data collectors and validators and various facilities to allow them to manipulate the data state of the system.
- It interacts with the database to check whether various users are stored in the database.

Authorization:

- It checks whether the users or facilities are authorized to manipulate the state of a piece of data.
- The interaction with this subsystem is done through AuthorizationInterface with exposes the authorization utilities provided by the subsystem.

5.2.3 Internal Interfaces

There are 2 new Internal interfaces in the improved version of Afetbilgi. The detailed information and class diagrams will be given for those 2 new internal interfaces.

AuthenticationInterface

- It shows authentication facilities that can be used by the system to authenticate various types of users
- It exposes two main ways of authentication one using name and password and the other using authentication token
- It also exposes the facility to create new users and register them to the system.

AuthorizationInterface

- It exposes various authorization facilities provided by the authorization module. It allows other components to get list of users authorized to modify a data item.
- It allows other components to check authorized user to modify a data item or do the opposite.



Figure 5.5: Class Diagram for AuthenticationInterface

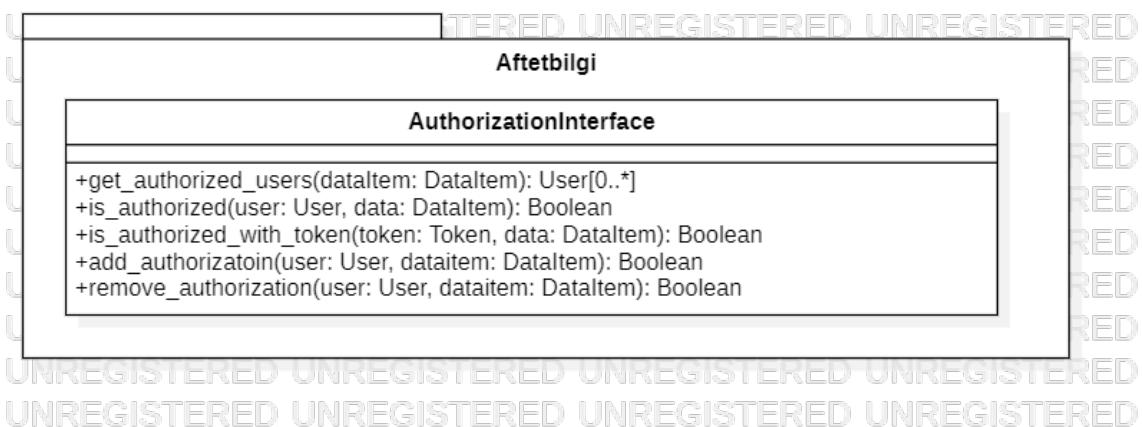


Figure 5.6: Class Diagram for AuthorizationInterface

5.2.4 Interaction Patterns

The interaction pattern shows the dynamics of interactions between various components of improved afetbilgi.

Data Item Modification

This interaction shows registers data validators would modify a piece of data existing in the system. To be able to modify data items the data collectors must be authenticated. They also need to be authorized to modify this piece of data. The sequence diagram shows all the interactions that take place.

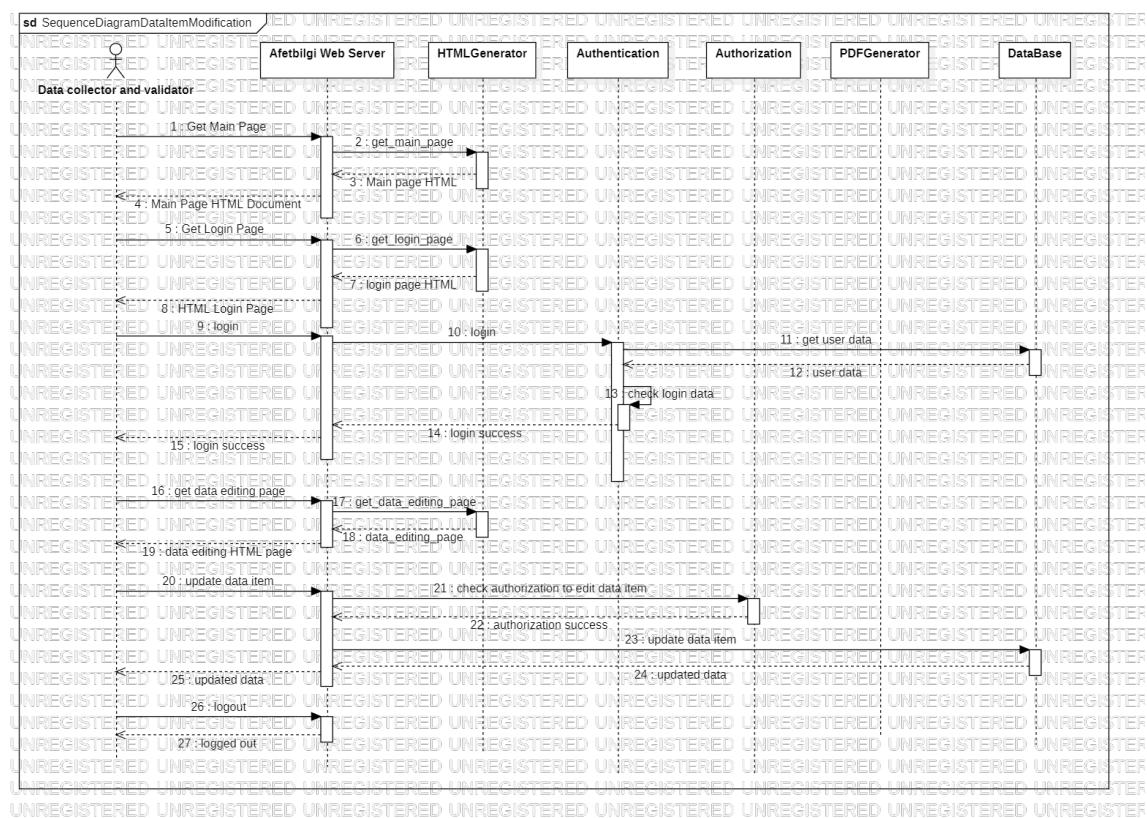


Figure 5.7: Sequence Diagram For DataItem Modification

5.3 Information View

5.3.1 Stakeholders' uses of this view

In improvements of this viewpoint, main page outlook can be improved with using cascading style sheets for display more clear information. The structure of website presents easy experience but for instance features of website can be colored with order of emergency. That improvement in information view would facilitates impacted individual usage which are most important stakeholders.

5.3.2 Database Class Diagram

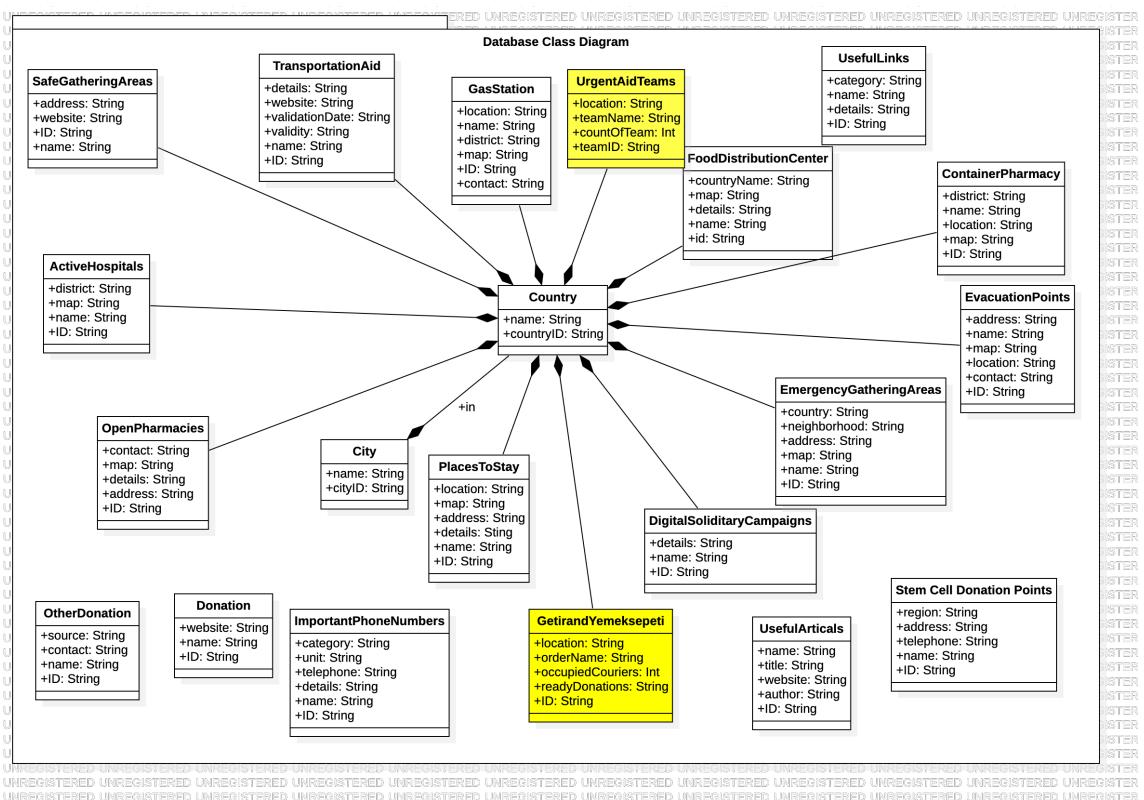


Figure 5.8: Database Class Diagram

5.3.3 Operations on Data

Function	Summary
City	Lets the user choose a city from Turkey, sorted with the emergency arrangement.
EmergencyGatheringAreas	Asks the user for the district and then asks for the neighborhood, providing an address name and a map location link.
SafeGatheringPlaces	Asks the user for the city and provides safe places for a convention.
EvacuationPoints	Asks the user for the city and provides evacuation areas.
TransportationAid	Informs the user about transportation facilities, details, website, verification dates, and end date.
FoodDistributionCenter	Asks the user for the city and state, and provides places for free food. It also includes quick links for holiday and special websites that provide places to stay in other cities.
GasStations	Asks the user for the city and state, and shows the address, map location, and phone numbers of gas stations.
ImportantPhoneNumber	Provides categories, units, phone numbers, and details for help.
UsefulLinks	Provides different categories, names, and detailed links to help-oriented websites.
UsefulArticles	Provides headers, writers, and website links for beneficial texts.
ActiveHospitals	Asks the user for the city, and provides hospital addresses, shows active hospitals, and gives detailed links.
OpenPharmacies	Asks the user for the city and state, and provides the name, address, map location, and phone number of open pharmacies.
UrgentAidTeams	Asks the user for the urgent need, city and the exact location and tries to deliver immediately assistance.
GetirandYemeksepeti	Asks the user for the location and shows the convenient free food and after choosing delivers it. Asks the volunteer the which type of food that volunteer wants to send.

Table 5.1: Database Class Diagram Improvements Operation Descriptions

5.4 Deployment View

5.4.1 Stakeholders' uses of this view

The Deployment view provides various stakeholders with useful insights about the system components, execution environment for each components, and how these components would be interacting with each other. For Afetbilig, there are mainly several stakeholders who would be affected by improvements in the systems. This improved Deployment diagram would make it easier for them to deal with the improved system. The stakeholder who shall benefit from this view are as follow:

- Developers: they would use the improved deployment view to gain insight about the execution environment of the new components of the system. This would help them in developing modules suitable for the execution environment.
- Testers: they would use the improved deployment view of the system to create tests for the newly created modules.
- System Administrators: they would use this view to manage and maintain the improved system.

5.4.2 Deployment Diagram

The deployment diagram shows the physical architecture of a system. It illustrates how various components of Afetbligi are distributed on various hardware and how they are connected. There are several new components that would be deployed on the same existing servers. The new software components would be deployed as follows:

- Authentication script : This component would be deployed on AWS and it would run to authenticate users by the main web server that responds to requests from users.
- Authorization script : This component would be deployed on AWS. It would run to authorized users to modify a specific piece of data by the main web server that responds to requests from users.
- HTML Document: These are several HTML documents for the newly added functionalities such as login, data modification GUI component, and register component. These components would be deployed on vercel.

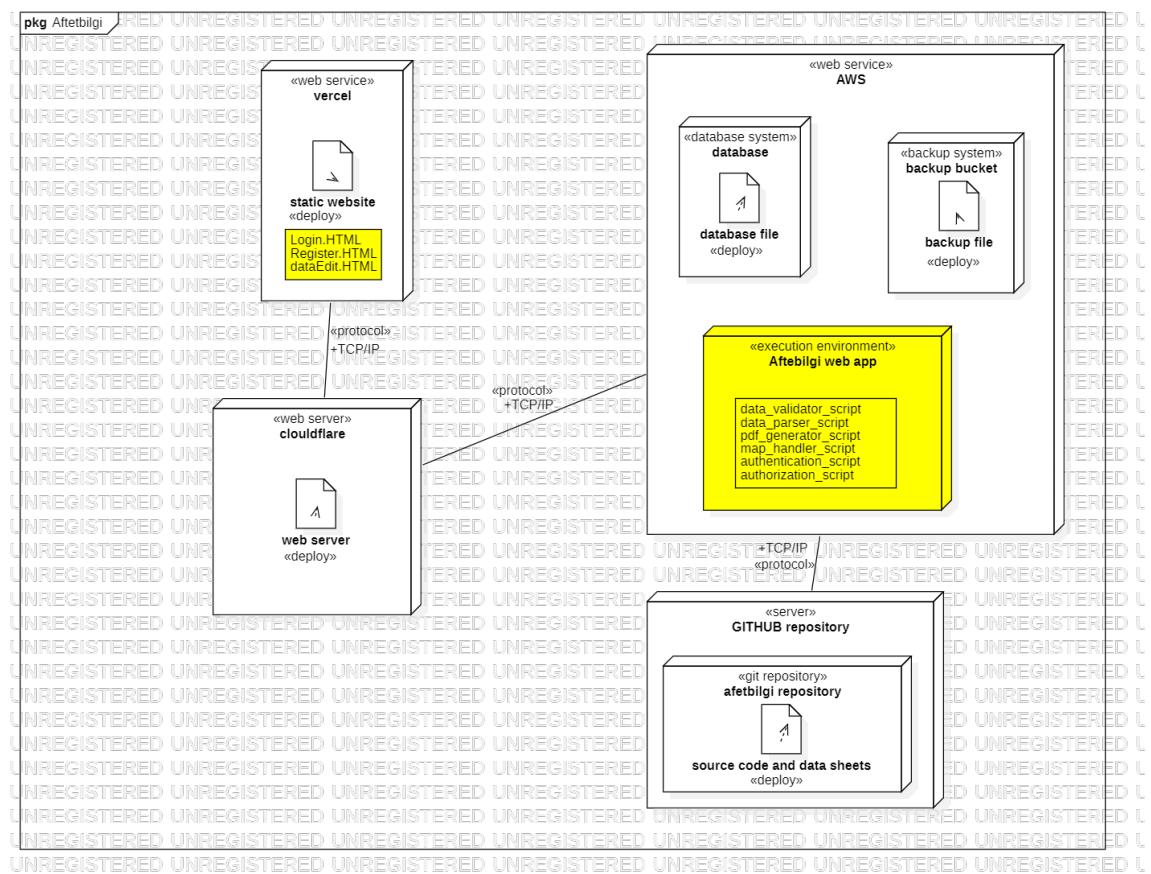


Figure 5.9: Deployment Diagram for Improved Afetbilgi

5.5 Design Rationale

1. **Context View Design Rationale:** The main logic behind the improvement of Context View Design is to extend the capabilities of the system in dealing with new external entities that would lead to core improvements in the functionalities of Afetbilgi. One of the new external entities that would have a communication channel with our system is urgent aid teams.
2. **Functional View Design Rationale:** The main logic behind the addition of authentication to it to allow only authenticated data collectors to be able to modify data so that data integrity is kept for all data points in the database.
3. **Information View Design Rationale:** The main logic behind design decision taken in informational view improvements was to add new entities for keeping information about urgent while keeping the original design of the database consistent.
4. **Deployment View Design Rationale:** The logic behind deploying new HTML Documents on Vercel is to keep modularity in the system since most such that all front end components of the system are deployed in a single server.