2022-2023 Spring
CENG350 Term Project

Software Architecture Description for afetbilgi.com

Hasan Ege Meyvecioğlu 2449783
Enes Şanlı 2375749

Contents

List of Figures

List of Tables

1 Introduction

This document is the Software Archtiecture Description (SAD) of the website **afetbilgi.com** created by METU students and graduates. The document includes architectural views of the platform, including context, functional, information and deployment views.

1.1 Purpose and objectives of afetbilgi.com

The objective of the afetbilgi.com is to confirm and spread information about the massive disaster, namely Kahramanmaras Earthquakes happened in the southeast part of the Turkey on February 6, 2023. While doing that, it aims to deliver the information in fast, accurate and simple ways.

1.2 Scope

Scope of the project can be listed as:

- Having a simple and well-organized interface to deliver safety-related knowledge.
- Providing accurate and up-to-date information about the Kahramanmaras Earthquakes of February 6, 2023.
- Being able to show every place that are somehow related to the disaster on the simple map view to ease reaching out for users.
- Demonstrating only the necessary information, by filtering based on location and related need.
- Providing a communication page for the people who wants to send up-to-date and verified information about earthquakes.
- Being accessible on multiple kind of electronic devices to ensure anyone can access as fast as possible.
- Having multiple language options, especially the languages that are spoken in the disaster area.

1.3 Stakeholders and their concerns

- **Users Affected by The Earthquake:** People who are victims of earthquake or who are faced with earthquake. These stakeholders need earthquake information and guidance before, during and after the earthquake. The concern of these steakholders is that the interface of the site is easy to use so that they can find the information they want on the site.
- **Users Not Affected by The Earthquake:** Users who want to add information to afetbilgi.com, who want to help the earthquake zone, who want to donate, but who are not in the earthquake zone. The concerns of these steakholders are the security of the accounts they use to donate and the ability to donate with an easy interface.

- **Users Who Are On The Search and Rescue Team:** These are the teams that respond quickly in case of earthquake and try to save the victims. These stakeholders need the information on the website to track the location, situation and needs of earthquake victims. The concern of these steakholders is to have an easy-to-use interface where earthquake victims can easily learn their needs and locations.
- **Related Institutions:** Institutions working on earthquake, such as public or private institutions such as municipalities, health institutions, public institutions. These stakeholders can prepare by using the information on the website before, during and after the disaster. The concern of these stakeholders is to provide earthquake victims with accurate location and assistance through the site interface.
- **Volunteer Developers:** The people or team who manage the website create and update the site content and keep the website running smoothly. These stakeholders determine the requirements of the website and try to make the website user-friendly. The concern of these stakeholders is to write code that is reliable, secure and meets the users' needs.
- **Media:** Media organizations that publish news about earthquake. These stakeholders can present accurate and reliable news by using the information on the website. The concern of these steakholders is to easily access the correct information and news about the earthquake from the interface of afetbilgi.com.

2 References

This document is written with respect to IEEE 42010 standard, using the source below:
42010-2011 - ISO/IEC/IEEE International Standard - Systems and software engineering– Architecture Description.

3 Glossary

Term	Definition
AI	Artifical Intelligence
API	Application Programming Interface
AWS	Amazon Web Services
ChatGPT	A large language model designed to chat with people
PDF	Portable Document Format

Table 1: Glossary

4 Architectural Views

The architectural view provides an overview of the system's high-level structure and components, their interactions, and how they achieve the system's goals.

4.1 Context View

4.1.1 Stakeholders' uses of this view

Stakeholders use this view to access relevant earthquake information and guidance, either for personal safety or to understand the situation in the affected area.

4.1.2 Context Diagram

afetbilgi.com is not part of a large system. However, it interacts with **Google Maps** services to retrieve any kind of location info. Data Collectors&Validators are simply the volunteers that shares confirmed information which are gathered from any kind of channel. All this information is presented by afetbilgi.com to users in a simple interface.

The context diagram is given in Figure 1.

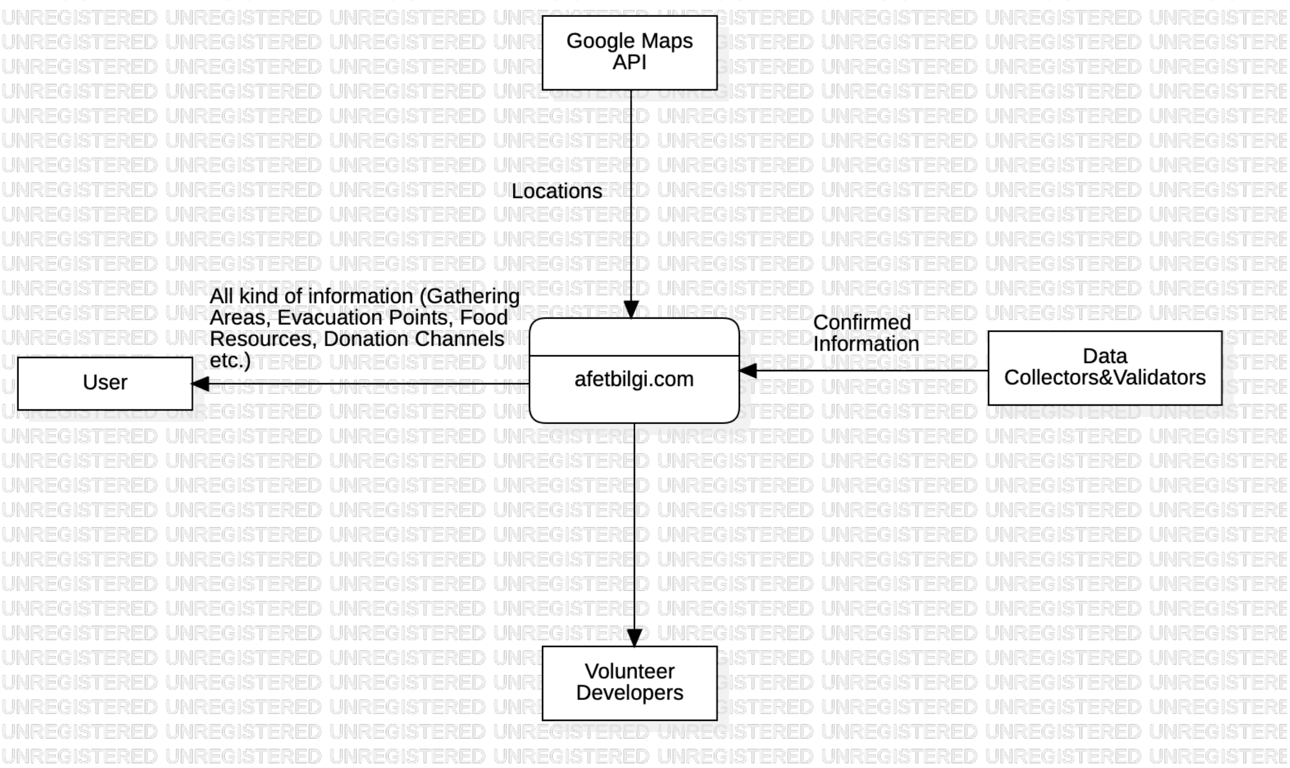
Google Maps API
Locations
All kind of information (Gathering Areas, Evacuation Points, Food Resources, Donation Channels etc.)
User
afetbilgi.com
Confirmed Information
Data Collectors&Validators
Volunteer Developers

Figure 1: System Context Diagram

4.1.3 External Interfaces

- **Main Server Interface:** This interface is responsible for managing the server's status and active visitors. It provides methods to start and stop the server, set the status, initialize the map service, and check the status and connection of the map service. Additionally, it can respond to information and location requests.

- **Human User Interface:** This interface represents the interface through which users interact with the system. It has an IP address and can connect to the system. Users can send information requests through this interface.

- **User:** This interface represents the interface between the system and the human user. It facilitates communication and interaction between the user and the system.

- **Data Collector&Validators**: This interface is used by data collectors and validators to interact with the system. It enables them to input and validate data for the system.

- **Volunteer Developers:** This interface is designed for volunteer developers who contribute to the system's development. It allows them to collaborate and interact with the system.

- **Google Maps API:** This interface represents the connection to the Google Maps API. It provides a connection status and the ability to respond to location requests.

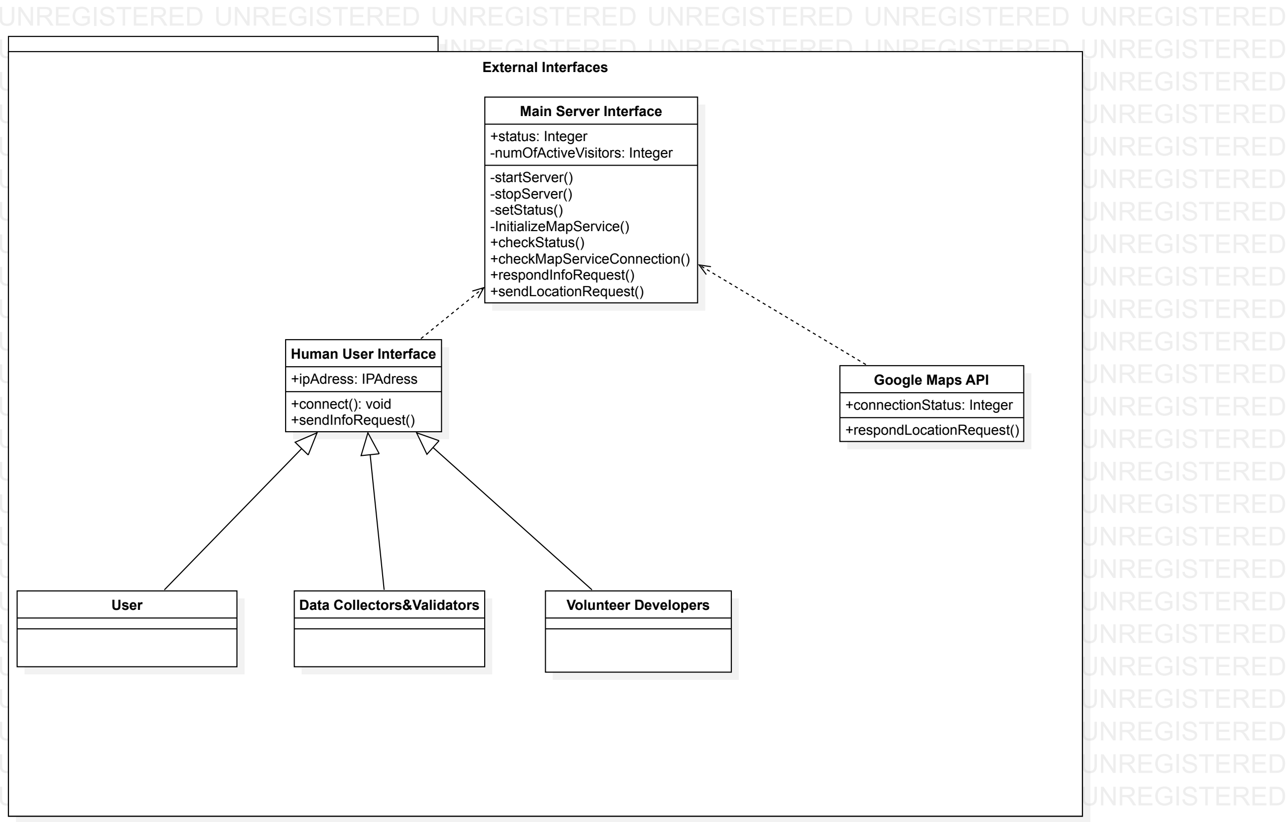

Figure 2: External Interfaces Class Diagram

4.1.4 Interaction scenarios

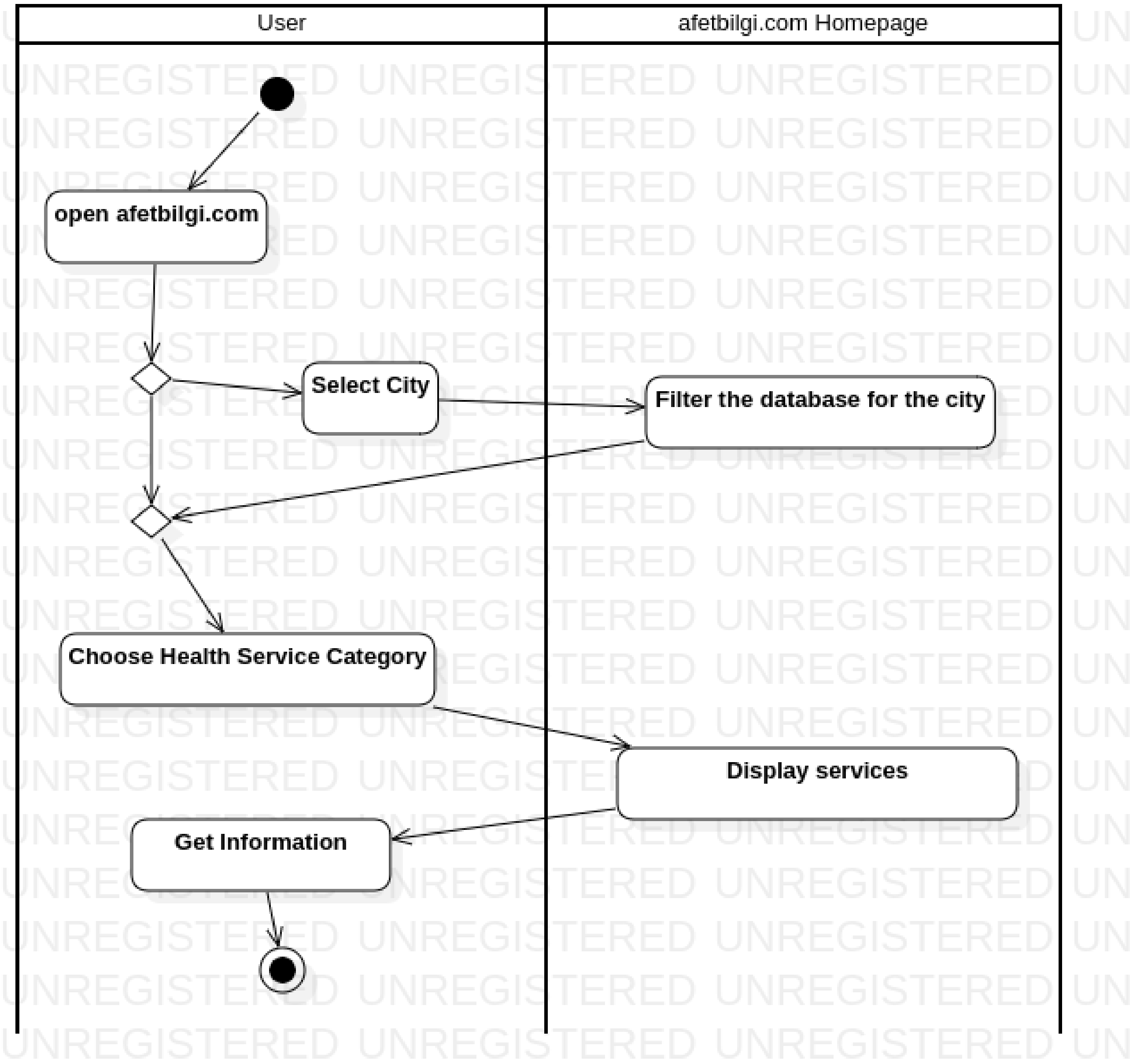

Figure 3: Activity Diagram for Learn Health Services

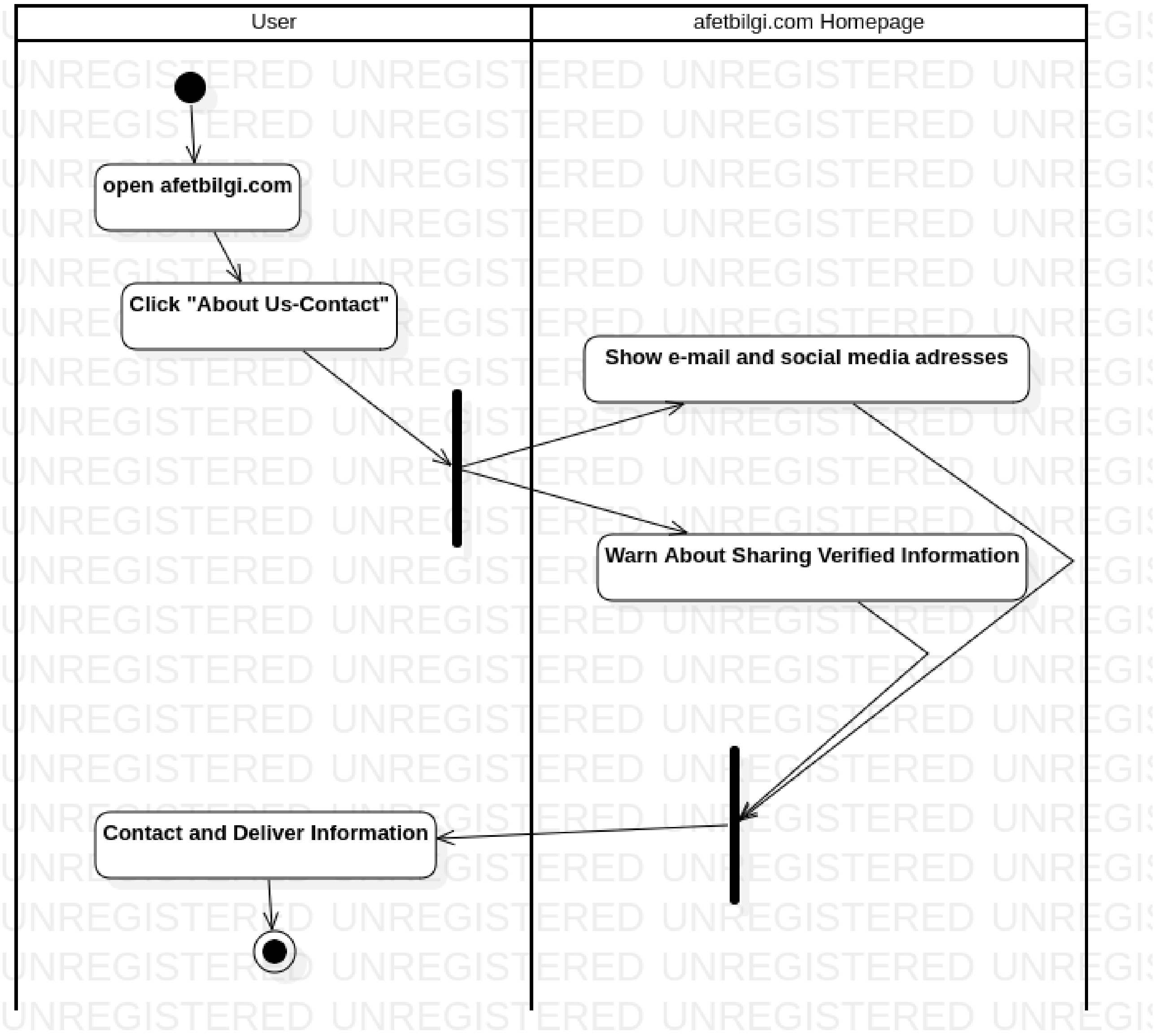
User
afetbilgi.com Homepage
open afetbilgi.com
Click "About Us-Contact"
Show e-mail and social media adresses
Warn About Sharing Verified Information
Contact and Deliver Information

Figure 4: Activity Diagram for Deliver Verified Information

4.2 Functional View

4.2.1 Stakeholders' uses of this view

Stakeholders use this view to interact with the website's features and functionalities, such as searching for information, donating, volunteering, or accessing emergency contacts.

4.2.2 Component Diagram

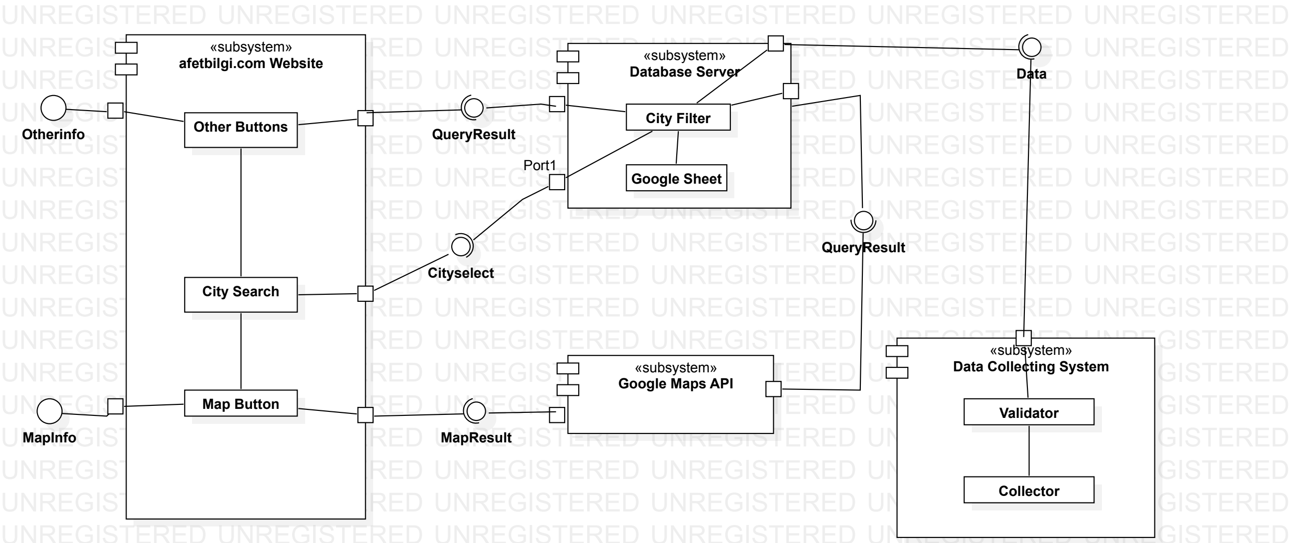

Figure 5: Component Diagram

- There are 4 subsystem for afetbilgi.com website. These are afetbilgi.com Website, Database Server, Google Maps API and Data Collecting System.
- afetbilgi.com Website consist of Other Buttons, City Search and Map Button.
- Other Buttons and Map Button provide an interface for users to select the information they want, and they get information from the database based on the button they clicked.
- City Search offers an interface that allows users to select the cities they want to get information about.
- Database Server consist of City Filter and Google Sheet.
- City Filter allows to filter the data in the database according to the selected city.
- Google Sheet allows us to store data from Data Collecting System. Likewise, when a data is requested, we give the desired information to the user by using Google Sheet.
- Data Collecting System consist of Validator and Collector.
- Collector gives validators the information it collects and learns and sends it to Google Sheet after this information is approved.
- The validator verifies the accuracy of the information from the collector. If the information is correct, it will send it to Google Sheet.
- Google Maps API makes the data appear on the map when the Map Button is clicked

4.2.3 Internal Interfaces

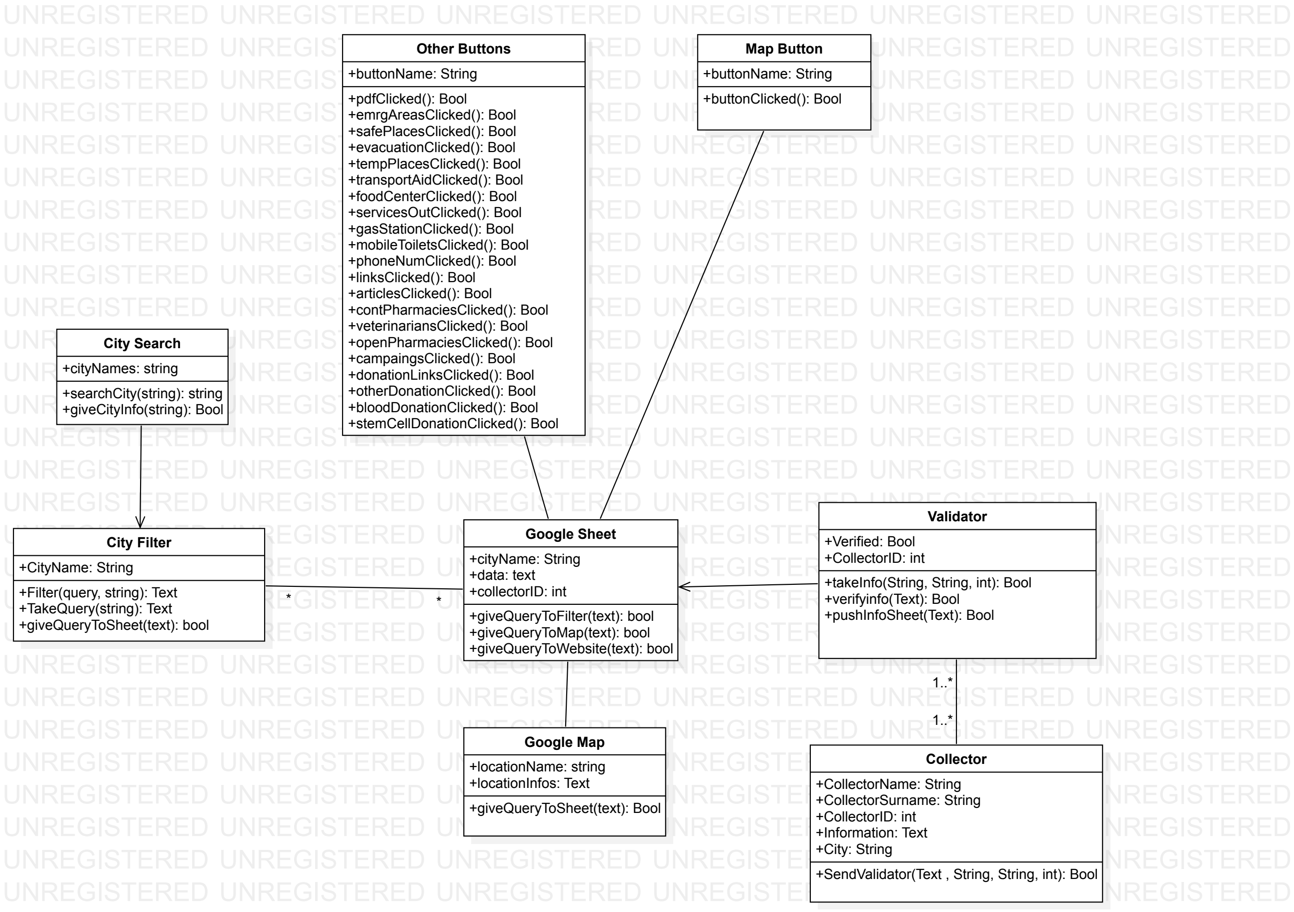
Other Buttons
+buttonName: String
+pdfClicked(): Bool
+emrgAreasClicked(): Bool
+safePlacesClicked(): Bool
+evacuationClicked(): Bool
+tempPlacesClicked(): Bool
+transportAidClicked(): Bool
+foodCenterClicked(): Bool
+servicesOutClicked(): Bool
+gasStationClicked(): Bool
+mobileToiletsClicked(): Bool
+phoneNumClicked(): Bool
+linksClicked(): Bool
+articlesClicked(): Bool
+contPharmaciesClicked(): Bool
+veterinariansClicked(): Bool
+openPharmaciesClicked(): Bool
+campaingsClicked(): Bool
+donationLinksClicked(): Bool
+otherDonationClicked(): Bool
+bloodDonationClicked(): Bool
+stemCellDonationClicked(): Bool
Map Button
+buttonName: String
+buttonClicked(): Bool
City Search
+cityNames: string
+searchCity(string): string
+giveCityInfo(string): Bool
City Filter
+CityName: String
+Filter(query, string): Text
+TakeQuery(string): Text
+giveQueryToSheet(text): bool
*
*
Google Sheet
+cityName: String
+data: text
+collectorID: int
+giveQueryToFilter(text): bool
+giveQueryToMap(text): bool
+giveQueryToWebsite(text): bool
Validator
+Verified: Bool
+CollectorID: int
+takeInfo(String, String, int): Bool
+verifyinfo(Text): Bool
+pushInfoSheet(Text): Bool
1..*
1..*
Google Map
+locationName: string
+locationInfos: Text
+giveQueryToSheet(text): Bool
Collector
+CollectorName: String
+CollectorSurname: String
+CollectorID: int
+Information: Text
+City: String
+SendValidator(Text , String, String, int): Bool

Figure 6: Internal Interfaces Diagram

veterinariansClicked	Sends the information that Veterinarians was clicked to Google Sheet
openPharmaciesClicked	Sends the information that Open Pharmacies was clicked to Google Sheet
campaignsClicked	Sends the information that Digital Solidatary Campaigns was clicked to Google Sheet
donationLinksClicked	Sends the information that Monetary Donation Links was clicked to Google Sheet
otherDonationClicked	Sends the information that Other Donation was clicked to Google Sheet
bloodDonationClicked	Sends the information that Kızılay Blood Donation Places was clicked to Google Sheet
stemCellDonationClicked	Sends the information that Stem Cell Donation Places was clicked to Google Sheet
buttonClicked	Sends the information that Map Button was clicked to Google Sheet
giveQueryToSheet	Sends filtered data back to Google Sheet
Filter	By filtering the data received from Google Sheet according to the selected city, it ensures that only the data related to the selected city remains.
qiveQueryToFilter	Sends selected data according to selected button to city filter to apply city filter to data
qiveQueryToMap	Sends the data to Google Map.
giveQueryToWebsite	Sends the final version of the requested data to afetbilgi.com to be displayed on the website.
takeInfo	Gets the data that the collector wants to give from it

searchCity	Searches the desired city in all cities
giveCityInfo	Sends the selected city to the City Filter system
pdfClicked	Exports all information to PDF when the PDF button is clicked
emrgAreasClicked	Sends the information that Emergency Gathering Area was clicked to Google Sheet
safePlacesClicked	Sends the information that Safe Gathering Places was clicked to Google Sheet
evacuationClicked	Sends the information that Evacuation Points was clicked to Google Sheet
tempPlacesClicked	Sends the information that Temporary Accommodation Places was clicked to Google Sheet
transportAidClicked	Sends the information that Transportation Aid was clicked to Google Sheet
foodCenterClicked	Sends the information that Food Distribution Center was clicked to Google Sheet
servicesOutClicked	Sends the information that Services Outside The Disaster Area was clicked to Google Sheet
gasStationClicked	Sends the information that Gas Stations was clicked to Google Sheet
mobileToiletsClicked	Sends the information that Mobile Toilets was clicked to Google Sheet
phoneNumClicked	Sends the information that Crucial Phone Number was clicked to Google Sheet
linksClicked	Sends the information that Useful Links was clicked to Google Sheet
articlesClicked	Sends the information that Useful Articles was clicked to Google Sheet
contPharmaciesClicked	Sends the information that Container Pharmacies was clicked to Google Sheet

Table 2: Internal Interface Operations

verifyinfo	Evaluates the data received by the collector according to whether it is correct or not.
pushInfoSheet	Sends data validated by Validator to Google Sheets for use
SendValidator	Sends the data that the Collector wants in Google Sheets to the Validator to verify

4.2.4 Interaction Patterns

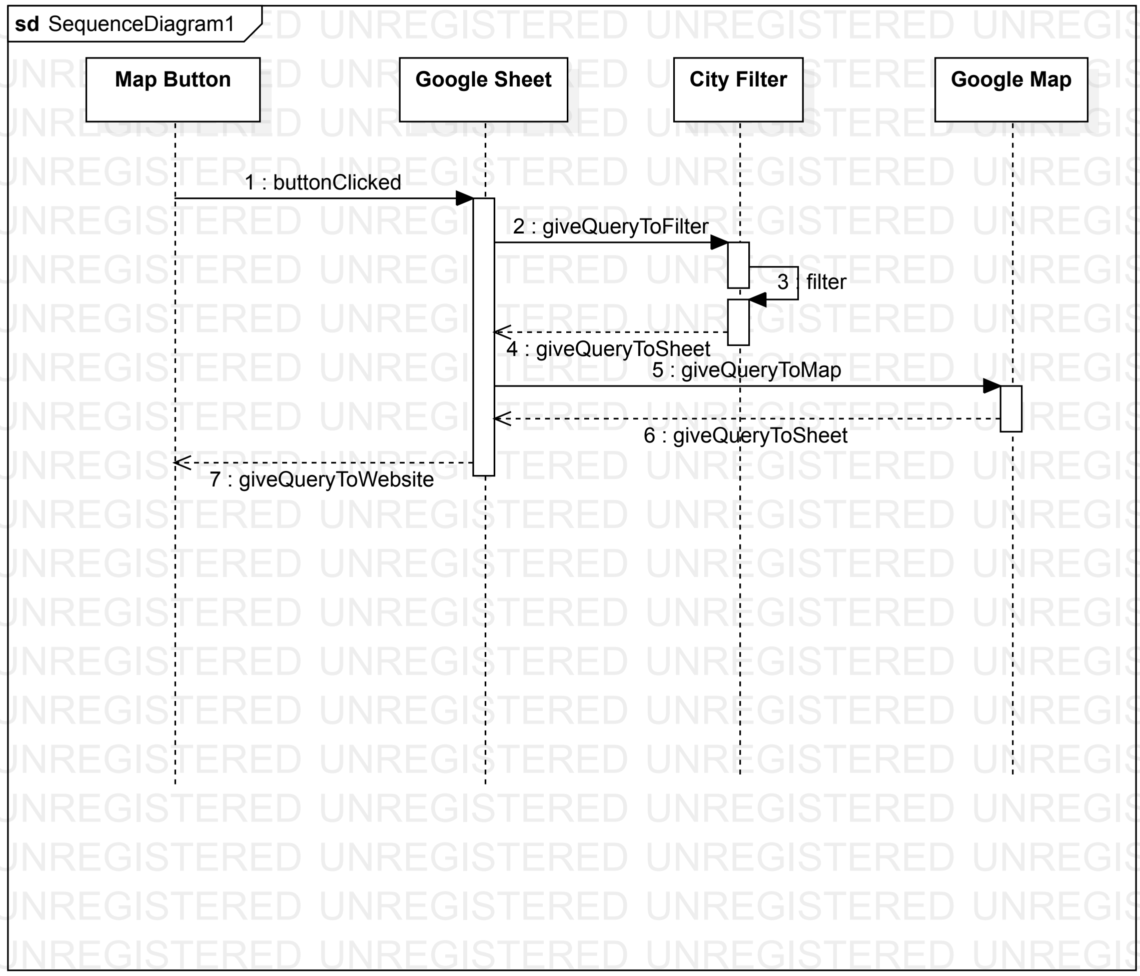

Figure 7: Sequence Diagram for Map Button and Google Map

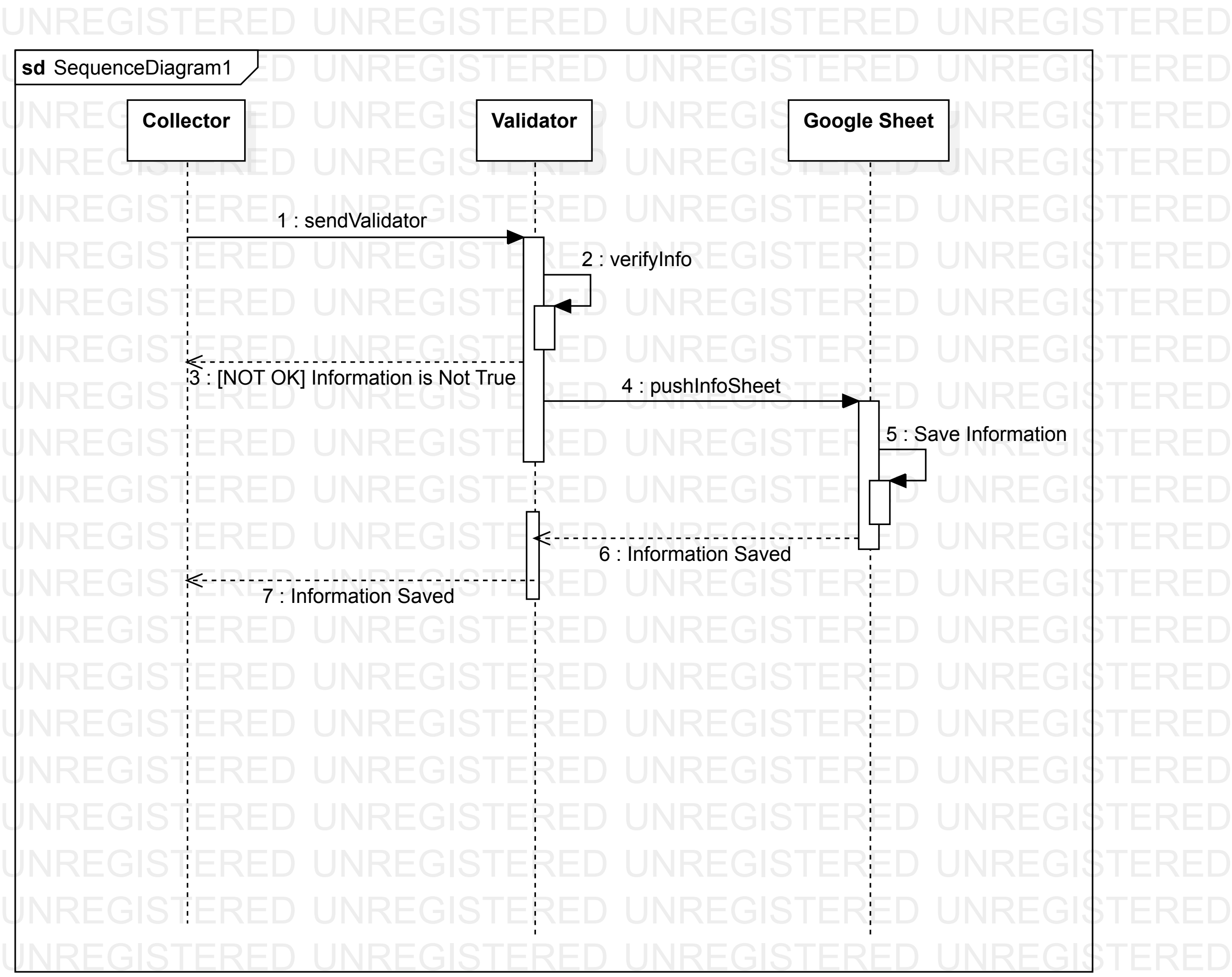

Figure 8: Sequence Diagram for Collector and Google Sheet

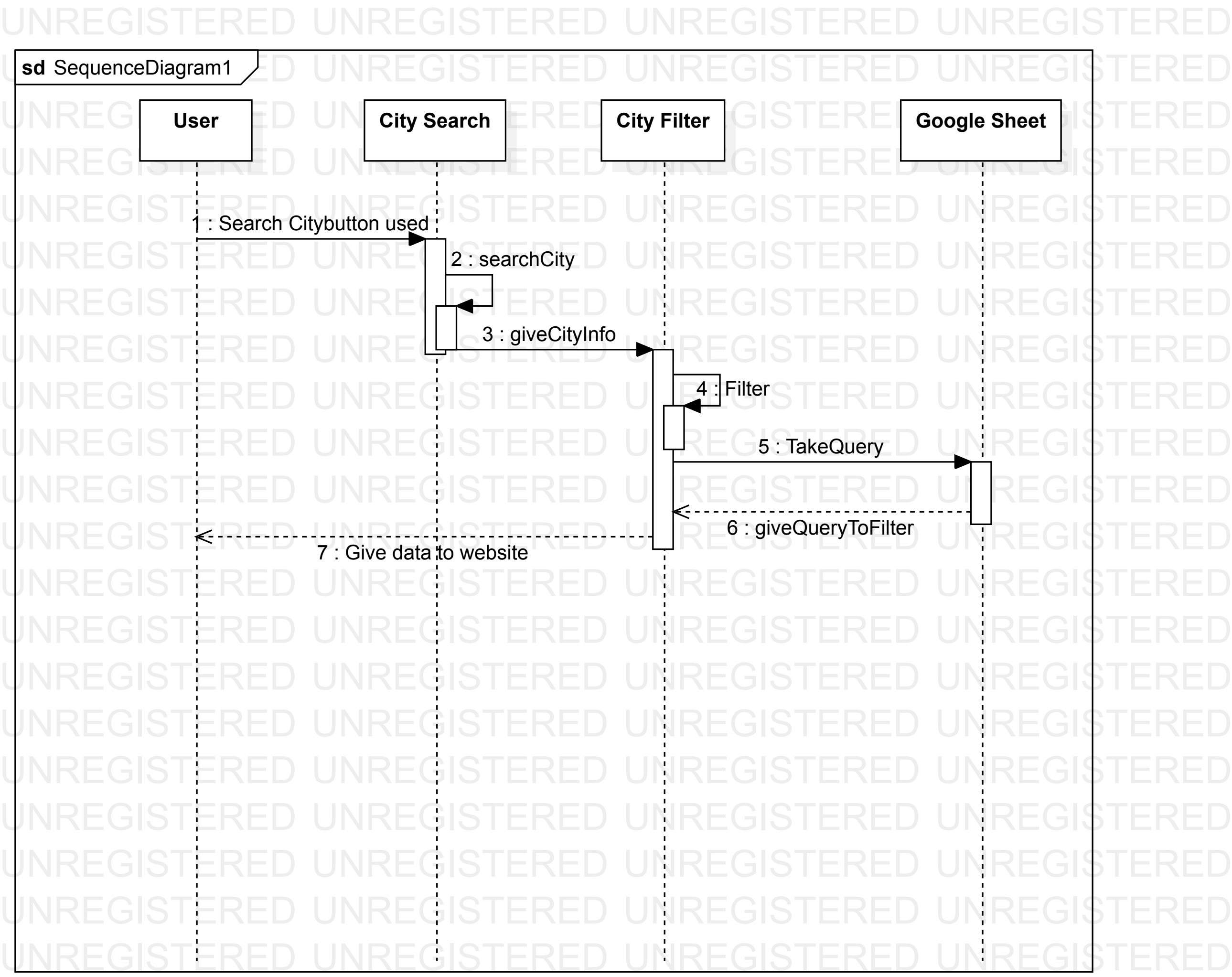

Figure 9: Sequence Diagram for Other Buttons and Google Sheet

4.3 Information View

4.3.1 Stakeholders' uses of this view

Stakeholders depend on the Information View to access accurate and up-to-date earthquake-related data, including affected areas, safety guidelines and victim needs.

4.3.2 Database Class Diagram

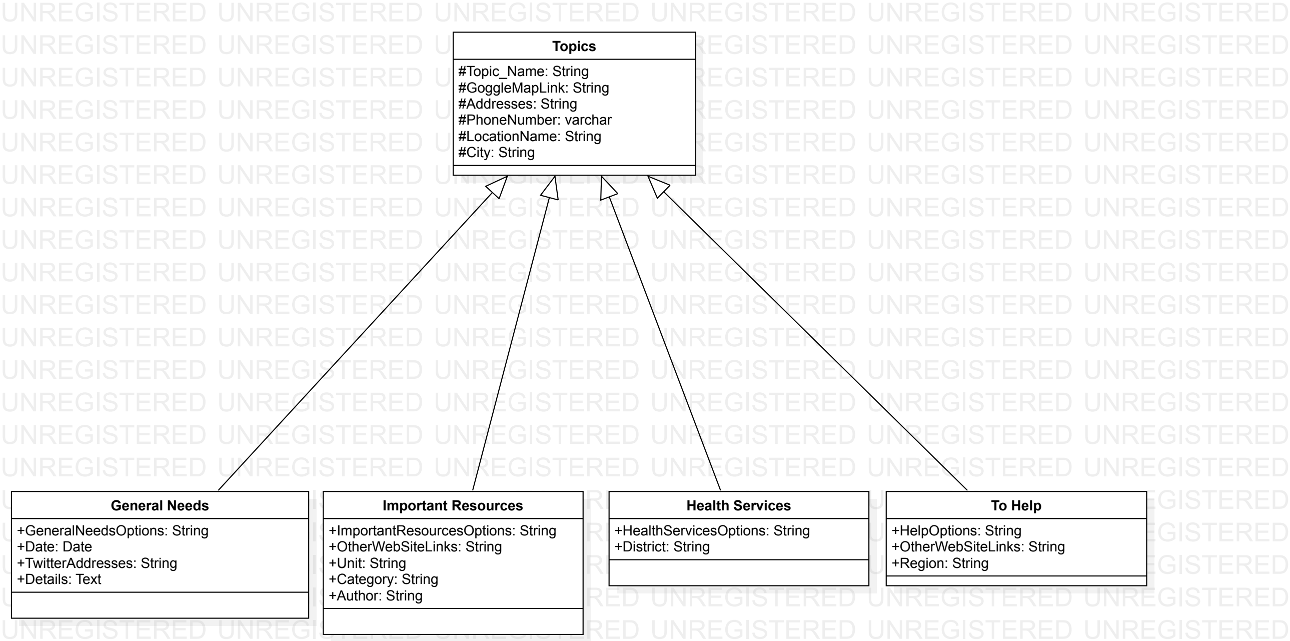

Figure 10: Class Diagram for Database

4.3.3 Operations on Data

Term	Definition
searchCity	Create: Read: City Update: LocationName, City, PhoneNumber, Addresses Delete:
giveCityInfo	Create: Read: Update: City Delete:
pdfClicked	Create: Read: TopicName Update: Delete:
emrgAreasClicked	Create: Read: GeneralNeedsOptions Update: LocationName, City, PhoneNumber, Addresses Delete:
safePlacesClicked	Create: Read: GeneralNeedsOptions Update: LocationName, City, PhoneNumber, Addresses Delete:
evacuationClicked	Create: Read: GeneralNeedsOptions Update: LocationName, City, PhoneNumber, Addresses Delete:
tempPlacesClicked	Create: Read: GeneralNeedsOptions Update: LocationName, City, PhoneNumber, Addresses Delete:

Table 3: Operations on Data

transportAidClicked	Create: Read: GeneralNeedsOptions Update: LocationName, City, PhoneNumber, Addresses Delete:
foodCenterClicked	Create: Read: GeneralNeedsOptions Update: LocationName, City, PhoneNumber, Addresses Delete:
servicesOutClicked	Create: Read: GeneralNeedsOptions Update: LocationName, City, PhoneNumber, Addresses Delete:
gasStationClicked	Create: Read: GeneralNeedsOptions Update: LocationName, City, PhoneNumber, Addresses Delete:
mobileToiletsClicked	Create: Read: GeneralNeedsOptions Update: LocationName, City, PhoneNumber, Addresses Delete:
phoneNumClicked	Create: Read: ImportantResourcesOptions Update: LocationName, City, PhoneNumber, Addresses Delete:
linksClicked	Create: Read: ImportantResourcesOptions Update: LocationName, City, PhoneNumber, Addresses Delete:
articlesClicked	Create: Read: ImportantResourcesOptions Update: LocationName, City, PhoneNumber, Addresses Delete:
veterinariansClicked	Create: Read: HealthServicesOptions Update: LocationName, City, PhoneNumber, Addresses Delete:

contPharmaciesClicked	Create: Read: HealthServicesOptions Update: LocationName, City, PhoneNumber, Addresses Delete:
openPharmaciesClicked	Create: Read: HealthServicesOptions Update: LocationName, City, PhoneNumber, Addresses Delete:
campaignsClicked	Create: Read: HelpOptions Update: LocationName, City, PhoneNumber, Addresses Delete:
donationLinksClicked	Create: Read: HelpOptions Update: LocationName, City, PhoneNumber, Addresses Delete:
otherDonationClicked	Create: Read: HelpOptions Update: LocationName, City, PhoneNumber, Addresses Delete:
bloodDonationClicked	Create: Read: HelpOptions Update: LocationName, City, PhoneNumber, Addresses Delete:
stemCellDonationClicked	Create: Read: HelpOptions Update: LocationName, City, PhoneNumber, Addresses Delete:
buttonClicked	Create: Read: TopicName, GoogleMapLink Update: Delete:
giveQueryToSheet	Create: GoogleMapLink, Addresses, PhoneNumber, LocationName, City Read: Update: Delete:

Filter	Create: Read: Update: Delete: City
qiveQueryToFilter	Create: Read: GoogleMapLink, Addresses, PhoneNumber, LocationName, City Update: Delete:
qiveQueryToMap	Create: GoogleMapLink Read: Addresses, PhoneNumber, LocationName, City Update: Delete:
giveQueryToWebsite	Create: Read: GoogleMapLink, Addresses, PhoneNumber, LocationName, City Update: Delete:
takeInfo	Create: GoogleMapLink, Addresses, PhoneNumber, LocationName, City Read: Update: Delete:
verifyinfo	Create: Read: Update: GoogleMapLink, Addresses, PhoneNumber, LocationName, City Delete: GoogleMapLink, Addresses, PhoneNumber, LocationName, City
pushInfoSheet	Create: Read: Update: GoogleMapLink, Addresses, PhoneNumber, LocationName, City Delete:
SendValidator	Create: Read: Update: GoogleMapLink, Addresses, PhoneNumber, LocationName, City Delete:

4.4 Deployment View

4.4.1 Stakeholders' uses of this view

Stakeholders, especially volunteer developers, utilize this view to understand the website's infrastructure, technical requirements, and deployment strategies, help them to effectively utilize and contribute to the website's services.

4.4.2 Deployment Diagram

The website is deployed and hosted on the Vercel platform and met with clients. The GitHub is used for version control and storing system files in the cloud. Updates in the GitHub triggers automatic deployment in Vercel. Database is hosted on the Amazon Web Services. It is stored as data bukcets in there and this data get retrieved whenever needed. Interaction between every node is via internet.

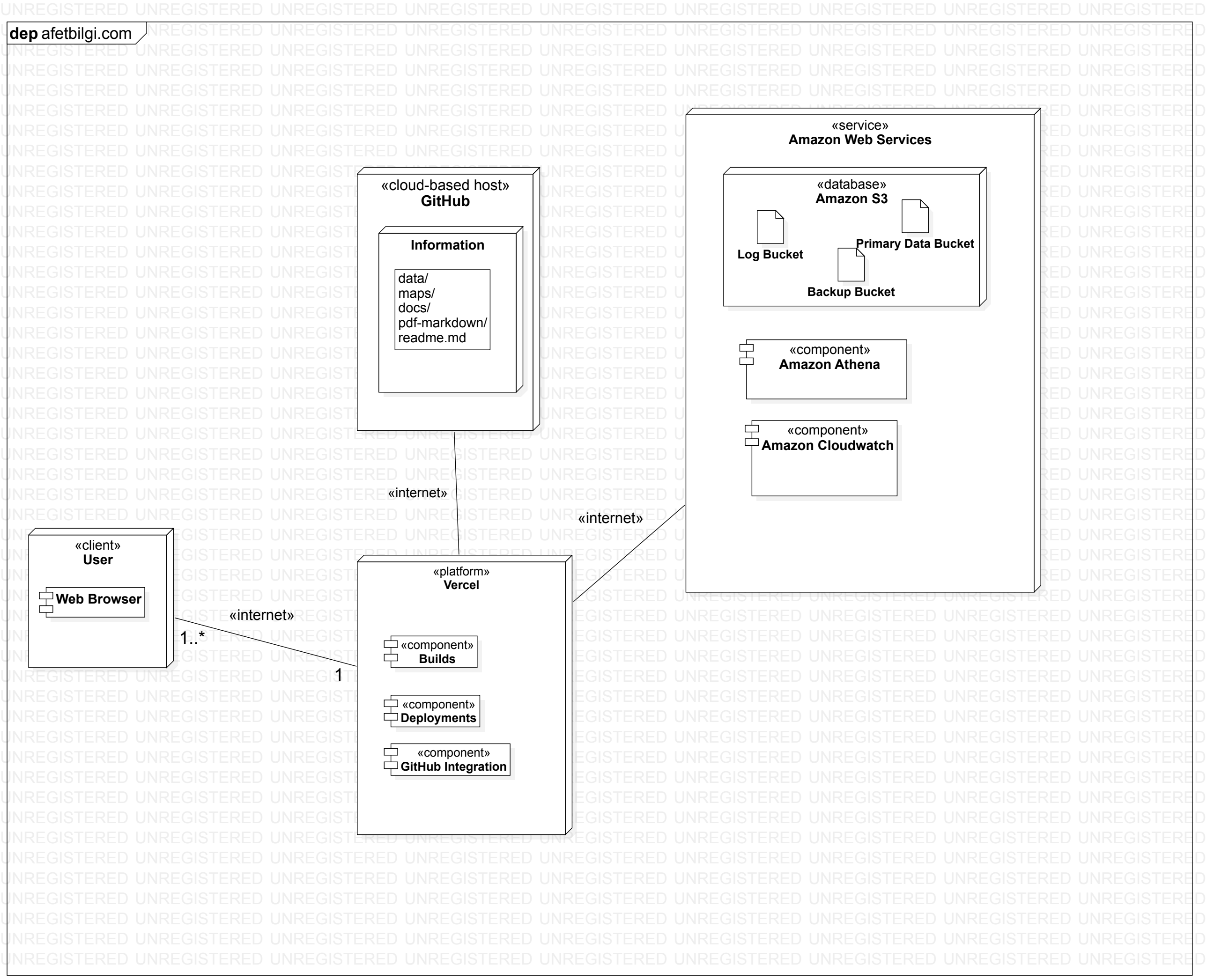

Figure 11: Deployment Diagram

4.5 Design Rationale

- **Rationale for Context View:**The reason for designing the Context View is to help different people access important information about earthquakes. They can use this view to stay safe or understand what's happening in the affected area. The design should be easy to use and understand so that anyone can find the information they need quickly during an emergency.
- **Rationale for Functional View:** The purpose is to make it easy for people to use the website's features. They can search for information, donate, volunteer, or find emergency contacts. The design should be user-friendly, with clear buttons and menus, so that people can easily do what they want to do on the website.
- **Rationale for Information View:** The goal is to give people accurate and up-to-date information about earthquakes. They can learn about the size of the earthquake, which areas are affected, safety guidelines, relief efforts, and what victims need. The design should show this information clearly to help people understand it better. It should also be updated regularly to make sure the information is reliable.
- **Rationale for Deployment View:** The reason for creating the Deployment View is to help people understand how the website works behind the scenes. This is important for institutions, volunteer developers, and media organizations who want to use or contribute to the website. The design should explain the technology and how everything is set up so that people can use and maintain the website effectively. It should be easy to find and understand the information about the website's infrastructure and technical requirements.

5 Architectural Views for Suggestions to Improve the Existing System

5.1 Context View

5.1.1 Stakeholders' uses of this view

Integrating an AI assistant, search bar and a few more features would allow stakeholders to interact with the system more effectively.

5.1.2 Context Diagram

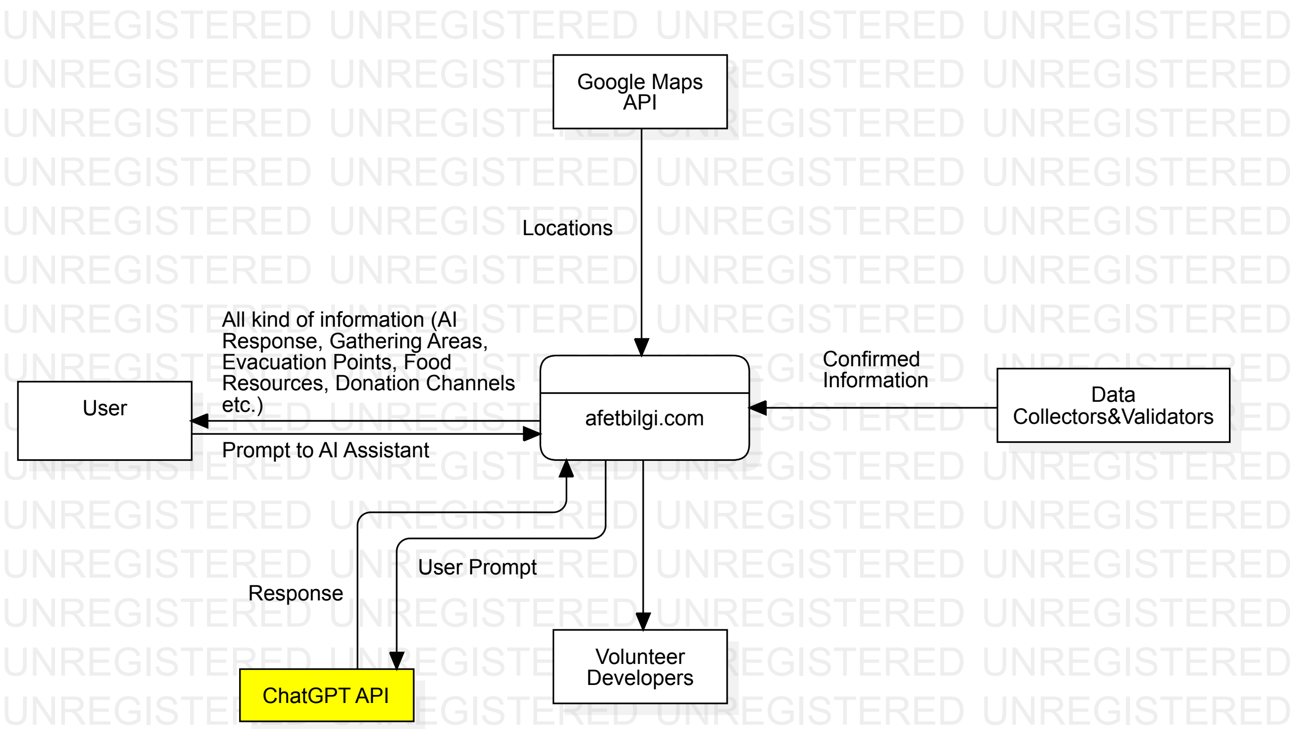
Google Maps API
Locations
All kind of information (AI Response, Gathering Areas, Evacuation Points, Food Resources, Donation Channels etc.)
User
Prompt to AI Assistant
afetbilgi.com
Confirmed Information
Data Collectors&Validators
User Prompt
Response
ChatGPT API
Volunteer Developers

Figure 12: Suggested System Context Diagram

5.1.3 External Interfaces

The diagram now includes the ChatGPT API interface, which represents the connection to an AI chatbot system. It has a connection status and enables the system to send responses. The main server interface has been updated with methods for sending text to the chatbot and receiving responses. The human user interface now includes methods for connecting to the AI assistant and starting conversations with it.

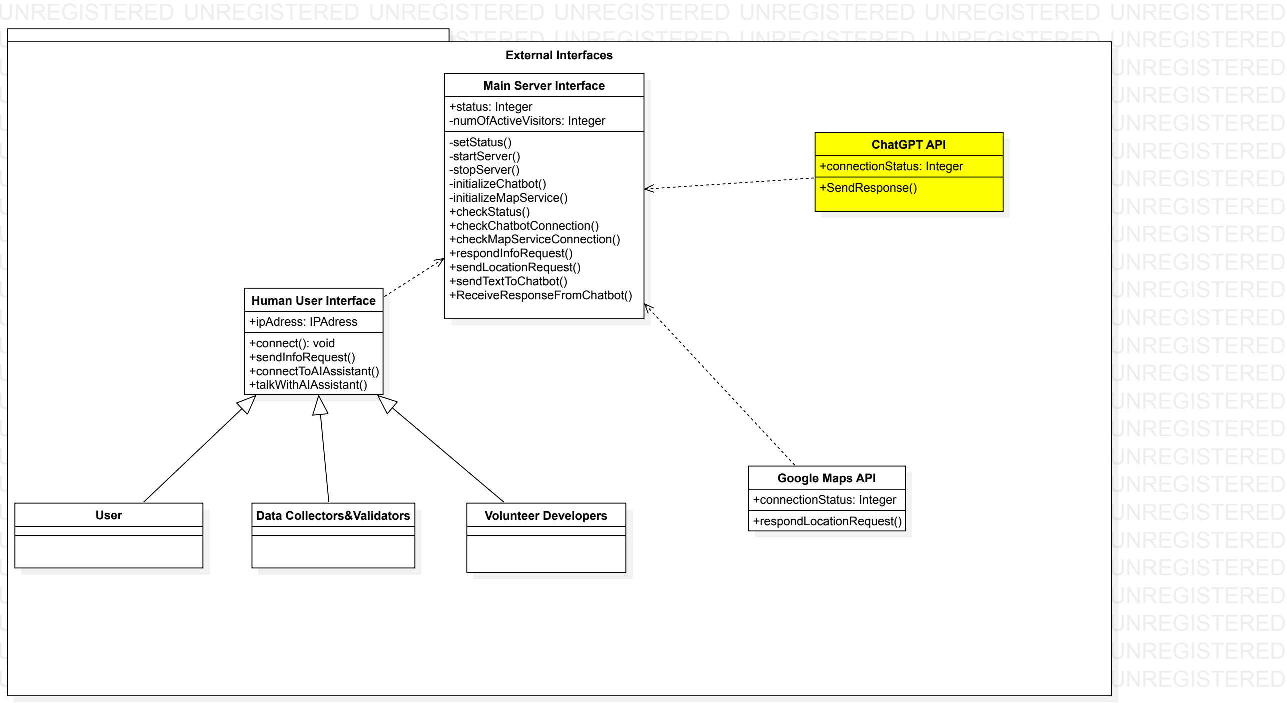

Figure 13: Suggested External Interfaces Class Diagram

5.1.4 Interaction scenarios

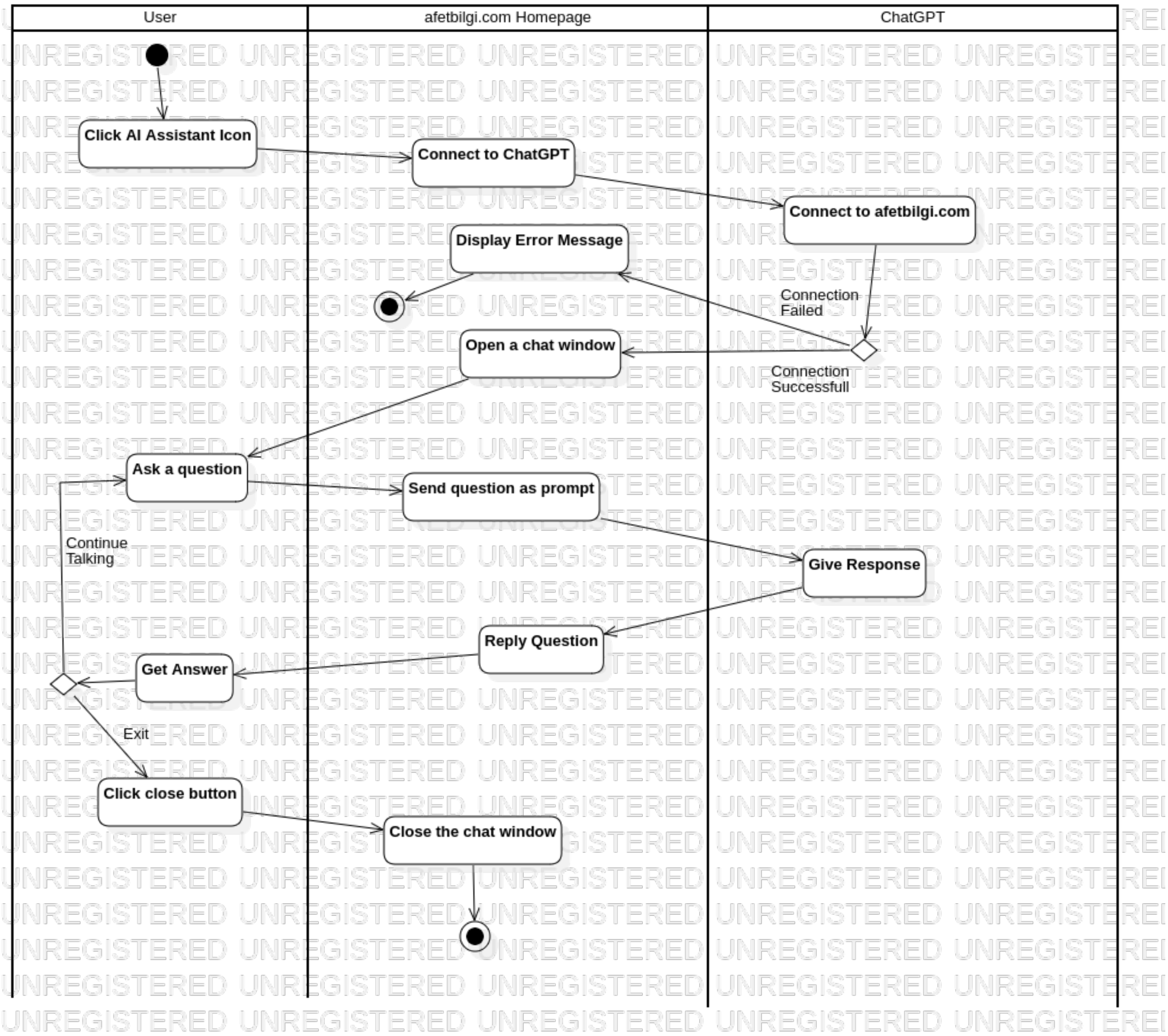

Figure 14: Activity Diagram for Talk With AI Assistant

5.2 Functional View

5.2.1 Stakeholders' uses of this view

The AI assistant and search bar would streamline stakeholders' interaction with the system's features, enabling them to easily navigate and interact.

5.2.2 Component Diagram

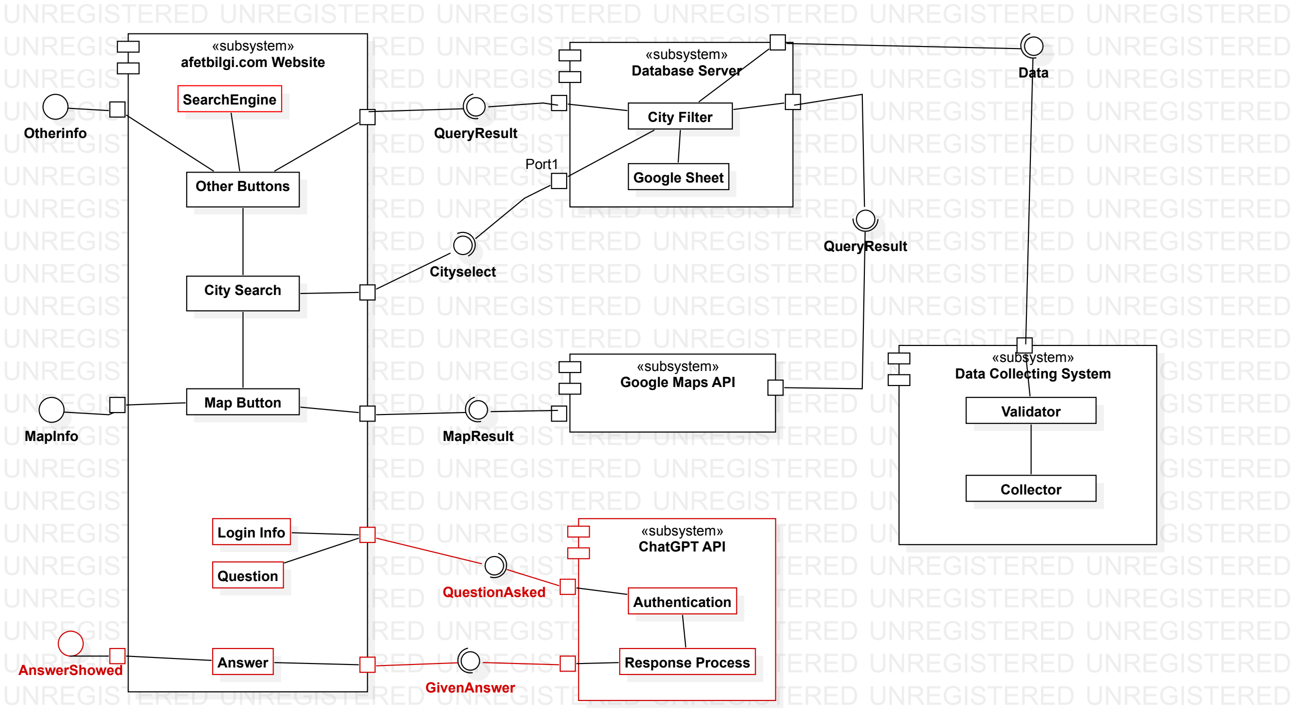

Figure 15: Component Diagram for Suggestions to Improve

5.2.3 Internal Interfaces

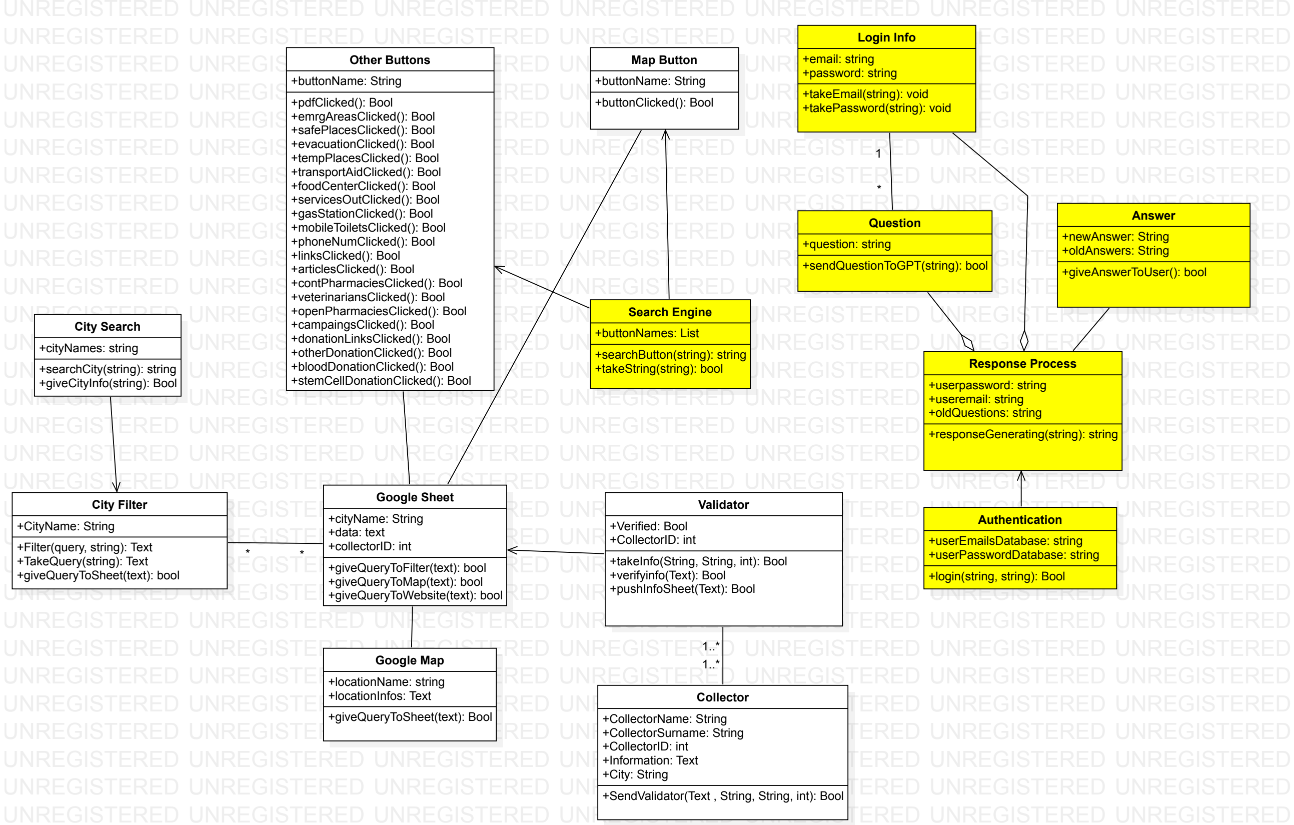
Other Buttons
+buttonName: String
+pdfClicked(): Bool
+emrgAreasClicked(): Bool
+safePlacesClicked(): Bool
+evacuationClicked(): Bool
+tempPlacesClicked(): Bool
+transportAidClicked(): Bool
+foodCenterClicked(): Bool
+servicesOutClicked(): Bool
+gasStationClicked(): Bool
+mobileToiletsClicked(): Bool
+phoneNumClicked(): Bool
+linksClicked(): Bool
+articlesClicked(): Bool
+contPharmaciesClicked(): Bool
+veterinariansClicked(): Bool
+openPharmaciesClicked(): Bool
+campaingsClicked(): Bool
+donationLinksClicked(): Bool
+otherDonationClicked(): Bool
+bloodDonationClicked(): Bool
+stemCellDonationClicked(): Bool
Map Button
+buttonName: String
+buttonClicked(): Bool
Login Info
+email: string
+password: string
+takeEmail(string): void
+takePassword(string): void
1
*
Question
+question: string
+sendQuestionToGPT(string): bool
Answer
+newAnswer: String
+oldAnswers: String
+giveAnswerToUser(): bool
City Search
+cityNames: string
+searchCity(string): string
+giveCityInfo(string): Bool
Search Engine
+buttonNames: List
+searchButton(string): string
+takeString(string): bool
Response Process
+userpassword: string
+useremail: string
+oldQuestions: string
+responseGenerating(string): string
City Filter
+CityName: String
+Filter(query, string): Text
+TakeQuery(string): Text
+giveQueryToSheet(text): bool
*
*
Google Sheet
+cityName: String
+data: text
+collectorID: int
+giveQueryToFilter(text): bool
+giveQueryToMap(text): bool
+giveQueryToWebsite(text): bool
Validator
+Verified: Bool
+CollectorID: int
+takeInfo(String, String, int): Bool
+verifyinfo(Text): Bool
+pushInfoSheet(Text): Bool
Authentication
+userEmailsDatabase: string
+userPasswordDatabase: string
+login(string, string): Bool
1..*
1..*
Google Map
+locationName: string
+locationInfos: Text
+giveQueryToSheet(text): Bool
Collector
+CollectorName: String
+CollectorSurname: String
+CollectorID: int
+Information: Text
+City: String
+SendValidator(Text , String, String, int): Bool

Figure 16: Internal Interfaces for Suggestions to Improve

5.2.4 Interaction Patterns

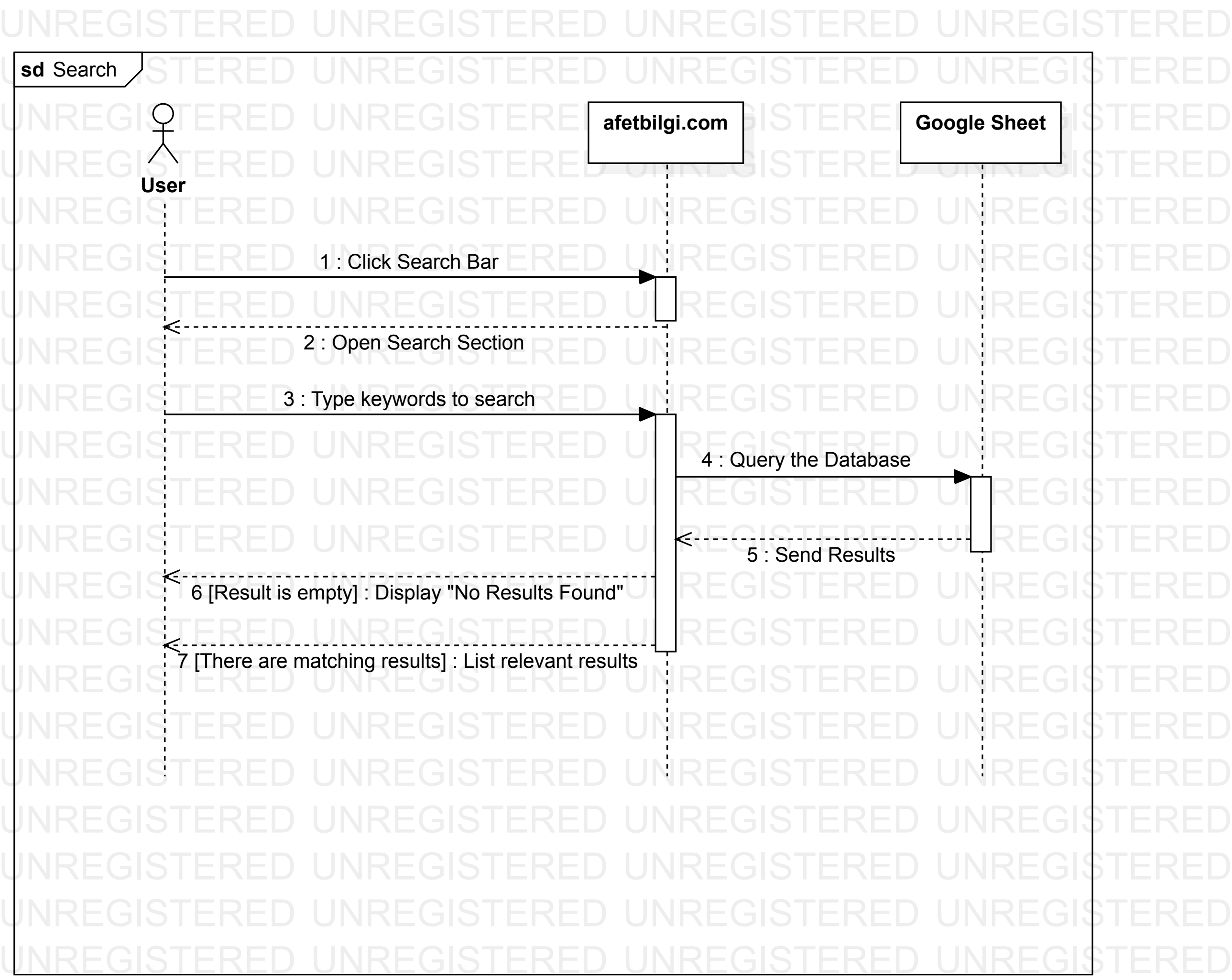
sd Search
User
afetbilgi.com
Google Sheet
1 : Click Search Bar
2 : Open Search Section
3 : Type keywords to search
4 : Query the Database
5 : Send Results
6 [Result is empty] : Display "No Results Found"
7 [There are matching results] : List relevant results

Figure 17: Sequence Diagram for Search

5.3 Information View

5.3.1 Stakeholders' uses of this view

New features enhance stakeholders' access to accurate and up-to-date earthquake-related data, providing immediate responses to their specific questions and allowing them to search for information they require.

5.3.2 Database Class Diagram

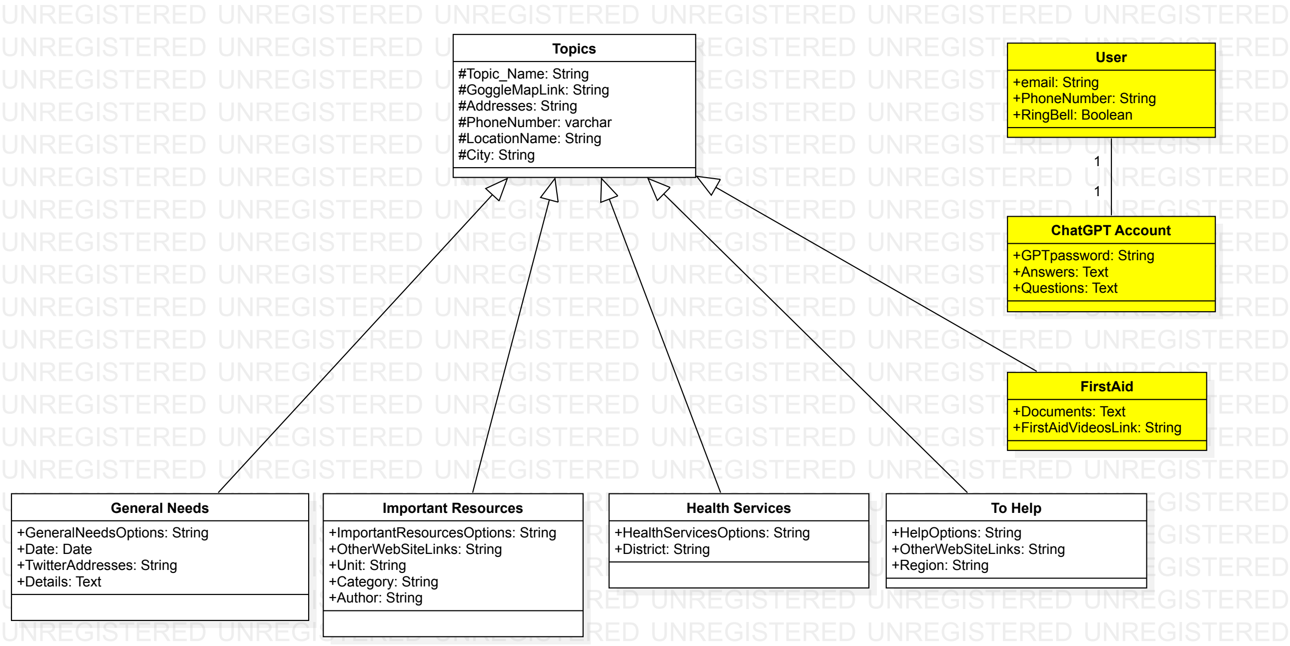

Figure 18: Class Diagram for Database

5.3.3 Operations on Data

searchButton	Create: Read: TopicName Update: Delete:
takeString	Create: TopicName Read: Update: Delete:
takeEmail	Create: email Read: Update: Delete:
takePassword	Create: GPTpassword Read: Update: Delete:
sendQuestionToGPT	Create: Question Read: Update: Question Delete:
giveAnswerToUser	Create: Read: Answer Update: Delete:
responseGenerating	Create: Answer Read: Update: Answer Delete:
Login	Create: Read: email, GPTpassword Update: Delete:

Table 4: Operations on Data for Suggested System

5.4 Deployment View

5.4.1 Stakeholders' uses of this view

New features increase technical guidance and assistance to stakeholders regarding the system's infrastructure and deployment strategies.

5.4.2 Deployment Diagram

When OpenAI's ChatGPT service is integrated to the system. The deployment process slightly changes as now the deployed system need to connect to ChatGPT system. It can be done by obtaining private API keys and credentials so that the ChatGPT services can be used with its APIs.

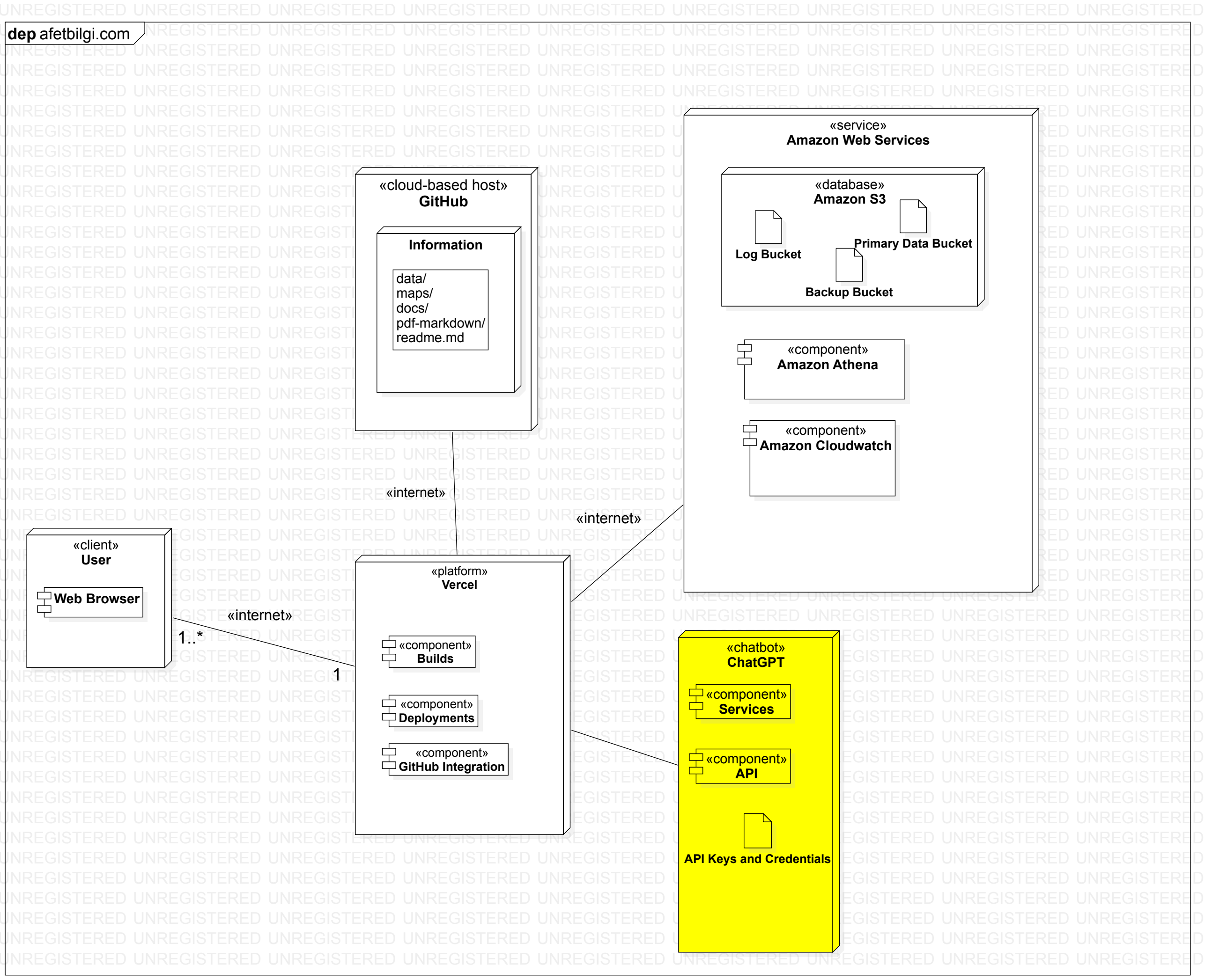

Figure 19: Suggested Deployment Diagram

5.5 Design Rationale

- **Rationale for Context View:**The addition of an AI assistant and search bar would enable stakeholders to interact with the assistant for personalized earthquake information and use the search bar to quickly find relevant details based on their preferences or location.

- **Rationale for Functional View:** With the new features stakeholders would have a more efficient way to navigate the website's features and functionalities. For instance, interacting with the assistant and using the search bar to find specific actions or information.
- **Rationale for Information View:** Stakeholders would be able to receive specialized information much more easily by using a search bar.
- **Rationale for Deployment View:** The Deployment View would benefit from the AI assistant, which could provide technical guidance and assistance to stakeholders.