Software Architecture Description for
Afet Bilgi

Emre Geçit, Baran Yancı

May 31, 2023

Contents

List of Figures

List of Tables

1   Introduction

1.1   Purpose and Objectives

The purpose of afetbilgi.com is to provide a centralized platform for users to access information about natural disasters and emergency response procedures. The website aims to be a reliable source of information that can help users prepare for, respond to, and recover from disasters.

1.2   Scope

The scope of afetbilgi.com includes the following:

- Provide up-to-date information about natural disasters, including their causes, impacts, and possible response procedures.

- Provide important phone numbers, or the locations of important places or services to the people affected by the earthquake.

- Provide information about donation centers; such as blood, stem cells, money and so on to the people who want to help the people affected by the earthquake.

- Offer people to create help points and share the locations of the help points, with the help of the system.

  The scope of afetbilgi.com does not include the following:

- Providing direct and instant communication services during a disaster event (this is the responsibility of official emergency response organizations).

- Offering financial assistance or physical aid during a disaster event (this is the responsibility of the official relief organizations).

1.3 Stakeholders and Concerns

The stakeholders of afetbilgi.com include the following:

- **Users:** They are concerned with accessing accurate information about natural disasters, preparing for emergencies, and communicating with others during a disaster event.

- **Emergency responders:** They are concerned with having access to timely and accurate information about disaster events to respond effectively.

- **Information providers:** People who voluntarily provide information about the earthquake. They can reach out to the maintainers of the system to provide information.

- **Maintainers / Developers:** People who maintain and develop the system. They develop the software and maintain it by verifying new information and adding it to the system.

2 References

The following references were consulted during the development of this SAD:

- ISO/IEC/IEEE 42010:2011, Systems and software engineering – Architecture description

- Rozanski, N., & Woods, E. (2011). Software systems architecture: working with stakeholders using viewpoints and perspectives. Addison-Wesley Professional.

3 Glossary

The following terms are used throughout this SAD and have the following meanings:

- **Disaster:** A sudden, catastrophic event that causes widespread damage or loss of life. Examples include earthquakes, hurricanes, floods, and wildfires.

- **Emergency responder:** A person who is trained to respond to emergencies and provide assistance to those in need. Examples include police officers, firefighters, and paramedics.

- **ISO/IEC/IEEE 42010:** A standard for software architecture description that provides a framework for describing the architecture of a software-intensive system.

- **Relief organization:** An organization that provides aid and support to communities affected by disasters. Examples include the Red Cross and UNICEF.

- **Scope:** The boundaries of a project or system, including what is included and what is excluded from consideration.

- **Stakeholder:** A person or group who has an interest in the success of a project or system. Examples include users, customers, and investors.

- **Viewpoint:** A perspective on the architecture of a system that emphasizes certain aspects or concerns. Examples include functional, data, and deployment viewpoints.

- **AWS:** Amazon Web Services, a cloud computing platform that provides on-demand access to computing resources.
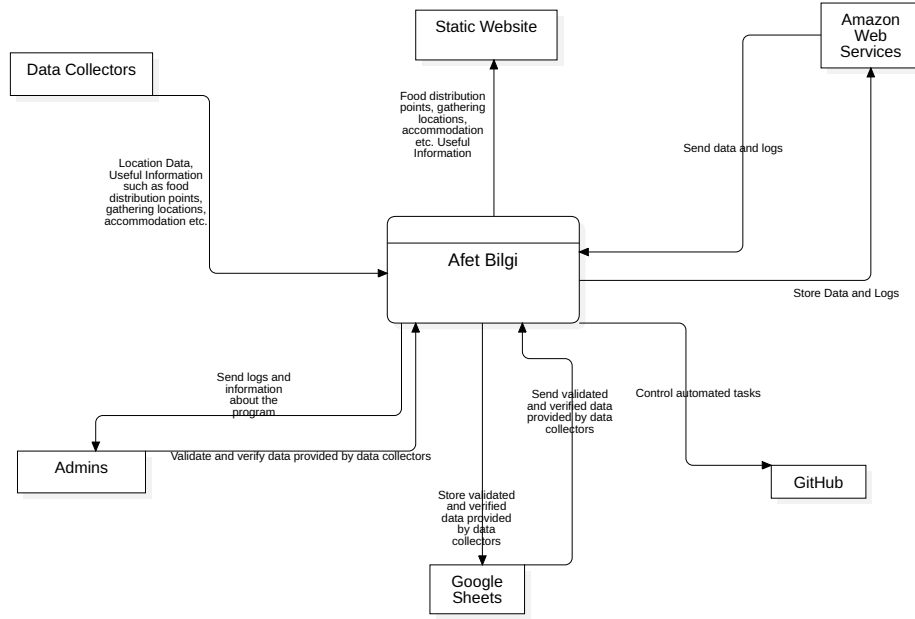
Figure 1: Context Diagram

4 Architectural Views

4.1 Context View

4.1.1 Stakeholders' Uses

There are four main stakeholders of the system: the users, emergency responders, information providers and the programmers. The users use this view to understand the general structure of their interaction with the system. The emergency responders use it to understand the general structure of the system to plan their response to the disaster. Information providers use it to supervise the processes their data goes through. Programmers use it to understand the general structure of the system to develop and maintain it.

4.1.2 Context Diagram

As shown in the diagram, users can access the website through a web browser or mobile device. They can view information about natural disasters, emergency response procedures, and other related content. Administrators have

7
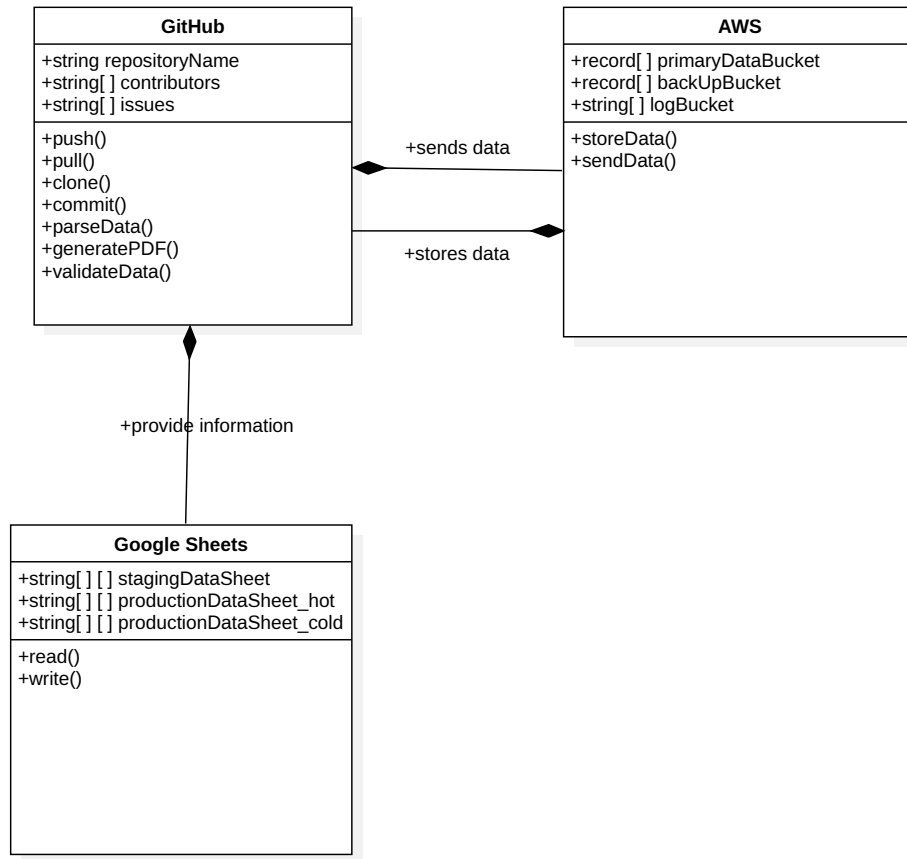
Figure 2: External Interfaces

additional privileges and can manage user accounts, update content on the website, and monitor site analytics.

Third-party services are also used to provide additional functionality on the website. For example, a map API is used to display maps of affected areas and help users locate nearby emergency resources.

Overall, the Context Diagram provides a high-level overview of how afetbilgi.com interacts with its external environment and stakeholders. This information is useful for understanding the scope of the system and identifying potential risks or dependencies that may impact its operation.

4.1.3 External Interfaces

As it can be observed from the figure above, afetbilgi.com has multiple external interfaces. GitHub, Information Provider, Information Validator, Developer, Google Sheets, and AWS. The operations given in the diagram can be summarized as follows:

Operation	Description
registerInformation()	Data providers can register new pieces of information to the Google Sheets-hosted database.
push()	Collaborators can change the code hosted in GitHub servers.
pull()	Collaborators can fetch the latest version of the source code hosted on GitHub servers.
clone()	Collaborators can download and work on the code.
commit()	Collaborators can submit their changes on the code.
parseData()	GitHub server parses the data stored on AWS.
generatePDF()	Automated scripts hosted on GitHub servers generate PDF files based on the latest information.
validateData()	Automated scripts hosted on GitHub servers validate the data stored on AWS.
storeData()	Automated scripts hosted on GitHub servers store the data on AWS.
sendData()	Automated scripts hosted on GitHub servers request data from AWS and AWS sends the data.
read()	Users can read the information stored on the Google Sheets.
write()	Users can write the information stored on the Google Sheets.

Table 1: Operations on External Interfaces

4.1.4 Interaction Scenarios

4.2 Functional View

4.2.1 Stakeholders' Uses

There are four main stakeholders of the system: the users, emergency responders, information providers and the programmers. The users use this view to understand the capabilities and limitations of the system. The emergency
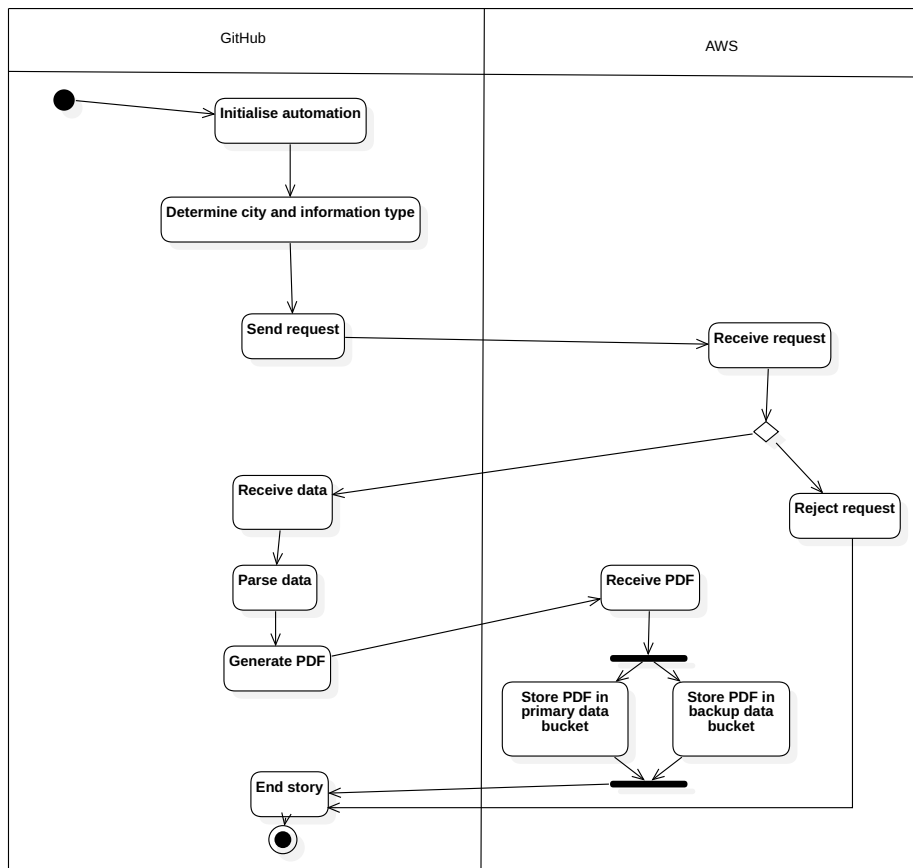
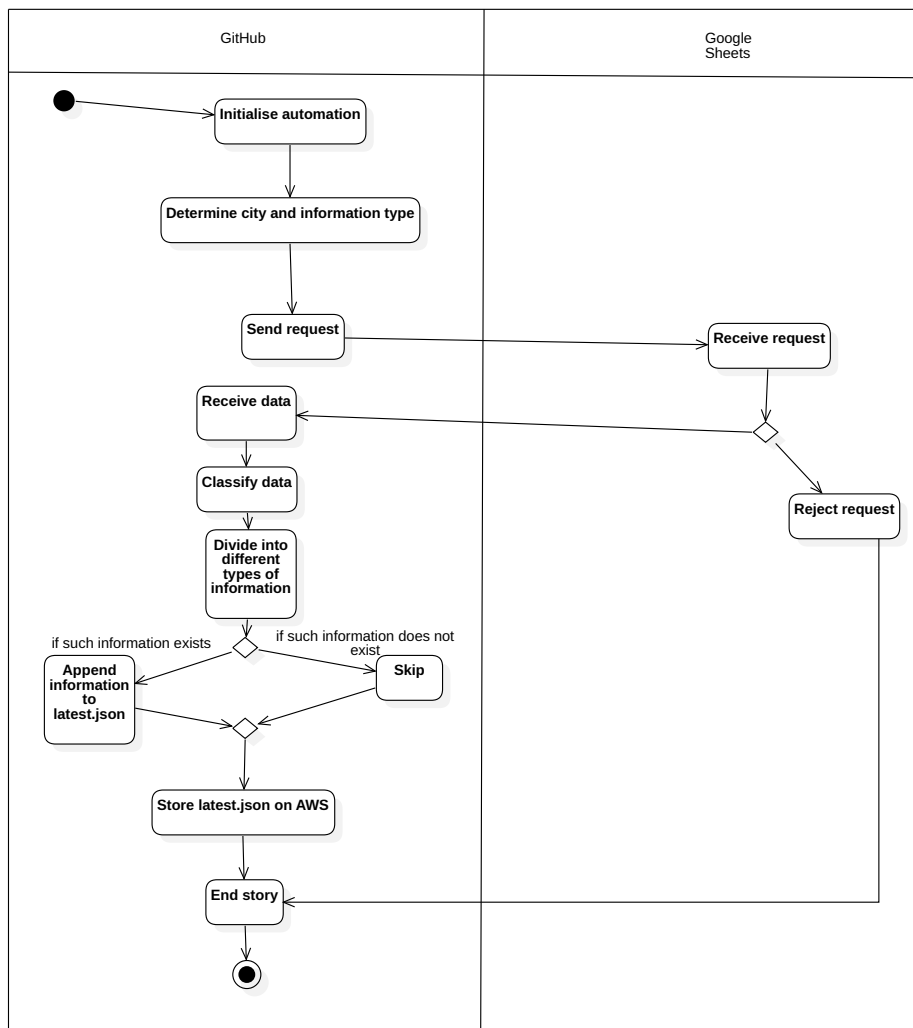**Figure 3:** Activity Diagram for GitHub - AWS Interactions

Figure 4: Activity Diagram for GitHub - Google Sheets Interactions

Figure 5: Component Diagram

responders use it to determine how the system can be used to respond to the disaster. Information providers use it to understand what procedures are used and how they operate on their data. Programmers use it to understand the capabilities and limitations of the system to develop and maintain it.

4.2.2 Component Diagram

Our system consists of two main subsystems: the Frontend and the PDF Generator.

4.2.3 Internal Interfaces

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra

metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

4.2.4   Interaction Patterns

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

4.3   Information View

4.3.1   Stakeholders' Uses

There are four main stakeholders of the system: the users, emergency responders, information providers and the programmers. The users use this view to understand the data that is stored in the system. The emergency responders use it to understand the data that is stored in the system to plan their response to the disaster. Information providers use it to understand what data is stored

in the system and how it is used. Programmers use it to understand the data that is stored in the system to develop and maintain it.

4.3.2 Database Class Diagram

4.3.3 Operations on Data

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

4.4 Deployment View

4.4.1 Stakeholders' Uses

There are four main stakeholders of the system: the users, emergency responders, information providers and the programmers. The users use this view to understand how the system is deployed and how it reacts to changes, in other words, how dynamic it is. The emergency responders use it to understand how the system is deployed and how it reacts to changes to plan their response to the disaster. Information providers use it to understand how the system is deployed and how the data is processed and the changes are reflected. Programmers use it to understand how the system is deployed and how it reacts to changes to develop and maintain it.

4.4.2 Deployment Diagram

4.5 Design Rationale

4.5.1 Context View

The rationale behind the context view is to provide a high-level overview of how afetbilgi.com interacts with its external environment and stakeholders. This information is useful for understanding the scope of the system and identifying potential risks or dependencies that may impact its operation.

4.5.2 Functional View

The rationale behind the functional view is to provide a detailed description of the capabilities and limitations of the system. This information is useful for understanding how the system can be used to respond to the disaster.

4.5.3 Information View

The rationale behind the information view is to provide a detailed description of the data that is stored in the system. This information is useful for understanding what data is stored in the system and how it is used.

4.5.4 Deployment View

The rationale behind the deployment view is to provide a detailed description of how the system is deployed and how it reacts to changes. This information is useful for understanding how dynamic the system is.

5 Architectural Views for Suggestions to Improve the Existing System

5.1 Context View

5.1.1 Stakeholders' Uses

There is no change in the stakeholders' uses.

5.1.2 Context Diagram

As shown in the diagram, users can access the website through a web browser or mobile device. They can view information about natural disasters, emergency response procedures, and other related content. Administrators have additional privileges and can manage user accounts, update content on the website, and monitor site analytics.

Third-party services are also used to provide additional functionality on the website. For example, a map API is used to display maps of affected areas and help users locate nearby emergency resources.

Overall, the Context Diagram provides a high-level overview of how afet-bilgi.com interacts with its external environment and stakeholders. This information is useful for understanding the scope of the system and identifying potential risks or dependencies that may impact its operation.
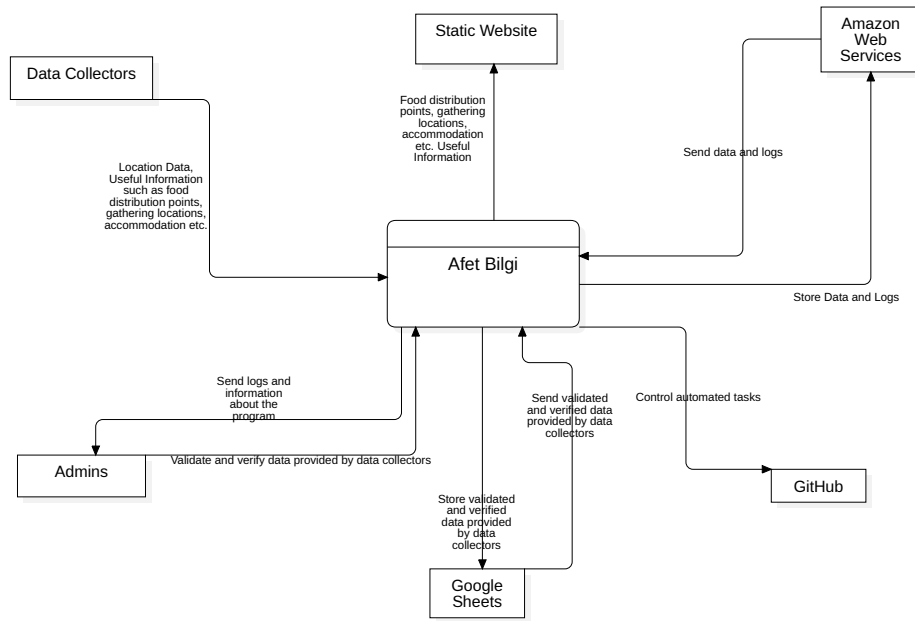
Figure 6: Context Diagram
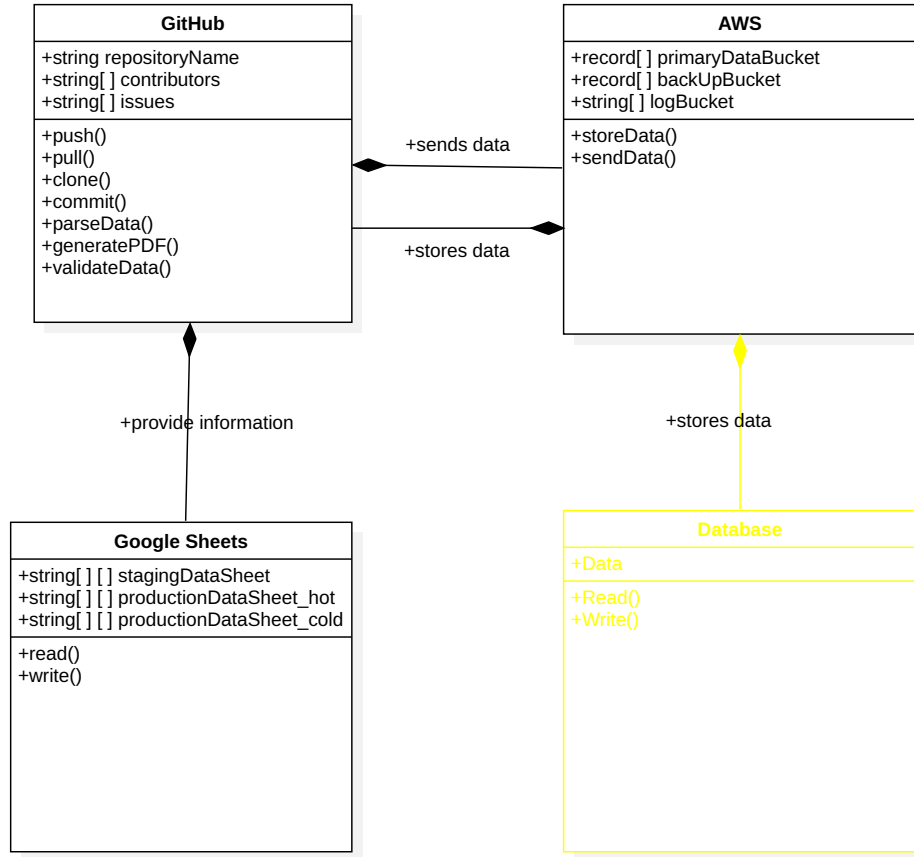
5.1.3 External Interfaces

Figure 7: External Interfaces

Operation	Description
read()	Read data from the database
write()	Write (or delete) data to the database

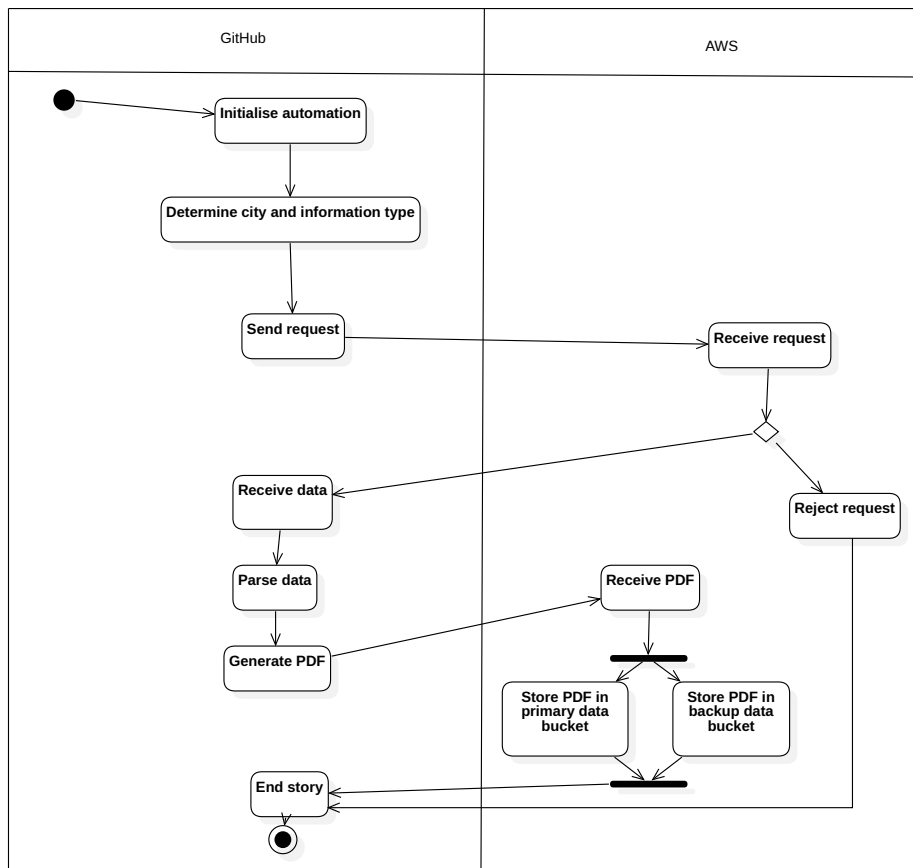**Table 2:** Operations on External Interfaces

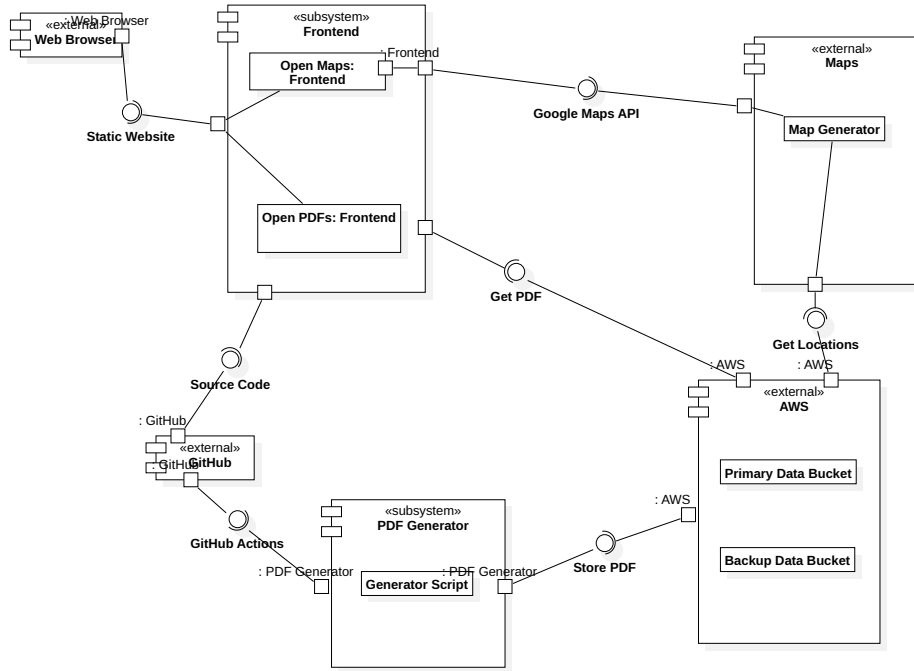**Figure 8:** Activity Diagram for GitHub - AWS Interactions

Figure 9: Component Diagram

5.1.4 Interaction Scenarios

5.2 Functional View

5.2.1 Stakeholders' Uses

There are four main stakeholders of the system: the users, emergency responders, information providers and the programmers. The users use this view to understand the capabilities and limitations of the system. The emergency responders use it to determine how the system can be used to respond to the disaster. Information providers use it to understand what procedures are used and how they operate on their data. Programmers use it to understand the capabilities and limitations of the system to develop and maintain it.

5.2.2 Component Diagram

Our system consists of two main subsystems: the Frontend and the PDF Generator.

5.2.3   Internal Interfaces

This section should include Internal Interfaces Class Diagram for your suggestions. Descriptions of the operations given in the internal interface class diagram should also be given. You should aim for 2 internal interfaces.

5.2.4   Interaction Patterns

This section includes 1 Sequence Diagram to show messaging sequences taking place among the system components over the internal interfaces for your suggestions. Choose the most complex interaction for sequence diagram. They must be different from those in your SRS document.

5.3   Information View

5.3.1   Stakeholders' Uses

There is no change in the stakeholders' uses of the information view.

5.3.2   Database Class Diagram

Database Class Diagram involving the key database or main memory objects for your suggestions. Complete with relevant associations. Descriptions of the non- obvious names (for classes, attributes, operations) should also be given.

5.3.3   Operations on Data

Descriptions of the operations given in the database class diagram. These operations may possibly deal with storage and handling of information regarding stores, customers, products and so on. Operations for your suggestions should be listed in a table or using bullets. These are usually CRUD (Create Read Update Delete) operations.

5.4 Deployment View

5.4.1 Stakeholders' Uses

There is no change in the stakeholders' uses of the deployment view.

5.4.2 Deployment Diagram

5.5 Design Rationale

5.5.1 Context View

The rationale behind the context view is to provide a high-level overview of how afetbilgi.com interacts with its external environment and stakeholders. This information is useful for understanding the scope of the system and identifying potential risks or dependencies that may impact its operation.

5.5.2 Functional View

The rationale behind the functional view is to provide a detailed description of the capabilities and limitations of the system. This information is useful for understanding how the system can be used to respond to the disaster.

5.5.3 Information View

The rationale behind the information view is to provide a detailed description of the data that is stored in the system. This information is useful for understanding what data is stored in the system and how it is used.

5.5.4 Deployment View

The rationale behind the deployment view is to provide a detailed description of how the system is deployed and how it reacts to changes. This information is useful for understanding how dynamic the system is.