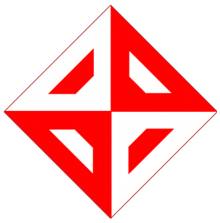

Software Requirements Specification for

Afet Bilgi

Emre Geçit, Baran Yancı

April 23, 2023

Contents

1 Introduction **3**

1.1 Purpose of the System . . . 3
1.2 Scope . . . 3
1.3 System Overview . . . 3
1.3.1 System Perspective . . . 3
1.3.2 System Functions . . . 4
1.3.3 Stakeholder Characteristics . . . 5
1.3.4 Limitations . . . 5
1.4 Definitions . . . 5

2 References **6**

3 Specific Requirements **6**

3.1 External Interfaces . . . 6
3.2 Functions . . . 6
3.3 Usability Requirements . . . 11
3.4 Performance Requirements . . . 11
3.5 Logical Database Requirements . . . 12
3.6 Design Constraints . . . 12
3.7 System Attributes . . . 12
3.7.1 Reliability . . . 12
3.7.2 Avaliability . . . 13
3.8 Supporting Information . . . 13

4 Suggestions for Future Work **13**

4.1 System Perspective . . . 13
4.2 External Interfaces . . . 13
4.3 Functions . . . 13
4.4 Usability Requirements . . . 14
4.5 Performance Requirements . . . 14
4.6 Logical Database Requirements . . . 14
4.7 Design Constraints . . . 14
4.8 System Attributes . . . 15
4.9 Supporting Information . . . 15

1 Introduction

This document is the Software Requirements Specification for the **Afet Bilgi** project, developed by a group of METU students to verify and deliver important information to fight against February 6, 2023, Pazarcık Earthquake.

1.1 Purpose of the System

The purpose of the system is to provide information to those who are affected by the earthquake.

1.2 Scope

- The system is used by two groups of people: the people who are affected by the earthquake and the people who want to help the people who are affected by the earthquake.
- People affected by the earthquake can reach important phone numbers, or the locations of important places or services.
- People who want to help the people affected by the earthquake can reach to donation centers; such as blood, stem-cells, money and so on. People can also create help points and share the locations of the help points, with the help of the system.
- People can also help fight against the earthquake by providing information about the earthquake. Information reaches by email, gets verified by the system managers, and then gets published on the website.

1.3 System Overview

1.3.1 System Perspective

Figure 1: Context Diagram

The `afetbilgi.com` product is not an element of a larger system. The project is split into two main parts. The first part is the front-end of the website. The second part is the cloud services that is used to store and process the data. The front-end is a web application that is developed using TypeScript and the ReactJS framework. The front-end uses packages like MUI and is hosted on the static website `afetbilgi.com`. For the cloud services, the project uses Amazon Web Services (AWS) and the serverless framework. Alongside AWS, GitHub Actions is used for continuous integration and continuous deployment (CI/CD). The cloud services process the data and store it in a database. The data comes from individuals who enter and/or validate the data. The data is collected in Google Sheets and then processed by the cloud services. The cloud services are hosted on AWS. GitHub actions are also responsible for generating PDF files including information about affected areas, from the data in the database. The PDF files are then stored in the cloud services and can be accessed by the front-end.

1.3.2 System Functions

Functionalities of `afetbilgi.com` are summarized below. More detailed information about the functionalities can be found in Section 3.2.

Function	Summary
See PDF About a City	User can see a PDF about an affected city, that includes important information about the city and the earthquake.
Kızılay Donation Centers	User can see the locations of Kızılay donation centers.
Generate PDF Documents	Automatic PDF generation. The PDF files include information about affected areas.
Show Maps	Shows a map of useful locations in the affected area.
Change Language	User can change the language of the website.
Reach to depremyardim.com	User can click a link to reach to depremyardim.com.
Reach to afetharita.com	User can click a link to reach to afetharita.com.
Reach to deprem.io	User can click a link to reach to deprem.io.
Join the Discord Server	User can join the Discord server using the website.
City Selection	User can select a city to see the information about the city.

Table 1: System Functions

1.3.3 Stakeholder Characteristics

Stakeholders of the system can be divided into four main groups:

- Victims of the disaster: People who are affected by the earthquake. They can reach information using the system.
- People who want to help: People who want to help the people who are affected by the earthquake. They can reach information using the system. They can also share the information of their helps.
- 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 to the system.

1.3.4 Limitations

The limitations of the system are listed below:

- The system should be available on mobile devices, as mobile devices are the most used devices on disaster times.
- The system should not be demanding on network as the network may be unavailable on disaster times.
- The system should response quickly as the people who are affected by the earthquake may need the information quickly.

1.4 Definitions

- **Earthquake**: Refers to the earthquake that happened on 6th of February 2023, Kahramanmaraş, Turkey.

- **Disaster**: For the time, earthquake is the only disaster that is considered in this project, although the system can be used for other disasters as well in the future.
- **Victim**: A person who is affected by the disaster.

2 References

This document is prepared with respect to **IEEE 29148-2018 standard**: *Systems and software engineering — Life cycle processes — Requirements engineering*

3 Specific Requirements

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam eget libero sollicitudin justo vehicula venenatis quis ut eros. Proin vitae.

3.1 External Interfaces

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam eget libero sollicitudin justo vehicula venenatis quis ut eros. Proin vitae.

3.2 Functions

Figure 2: Use Case Diagram

Table 2: See PDF About City Function

Table 3: Kızılay Blood Donation Places Function

Use case name	See PDF About a City
Actors	User, Static Website, PDF Viewer
Description	If a user wants to see the PDF document containing information about a city, the city selection dialog is shown, then the desired is picked on the dialog.
Data	PDF file about the city
Preconditions	The PDF for the city file must be refreshed previously.
Stimulus	User clicks on the "PDF" button and picks a city
Basic flow	Step 1: User clicks the "PDF" button Step 2: The pop-up dialog is shown Step 3: User picks the desired city on the city selection dialog
Alternative flow	-
Exception flow	-
Post conditions	User is redirected to the PDF viewer

Use case name	Kızılay Blood Donation Places
Actors	User, Static Website, Kızılay Website
Description	The user wants to see the blood donation locations.
Data	Blood Donation Locations
Preconditions	-
Stimulus	User clicks on the "Kızılay Blood Donation Places" button
Basic flow	Step 1: User clicks the "Kızılay Blood Donation Places" button
Alternative flow	-
Exception flow	-
Post conditions	User is redirected to the Kızılay website

Table 4: Generating PDF Documents Function

Table 5: Showing Maps Function

Use case name	Generating PDF Documents
Actors	GitHub Actions, automation code, Data Providers & Validators
Description	The PDF files are generated by the automated code running on GitHub Actions. Those PDF files include information about evacuation points, food distribution centers, pharmacies, gas stations and more, based on the districts of the given city. All the information comes from Google Sheets, which holds the data coming from voluntary individuals.
Data	Open pharmacies, evacuation points, food distribution centers, gas stations, acommodation places, veterinarians.
Preconditions	The information should be on Google Sheets prior to generation.
Stimulus	GitHub actions runs this task periodically.
Basic flow	Step 1: The data from sheets are fetched Step 2: The PDF file is generated using the Python script Step 3: The file is stored in AWS Cloud.
Alternative flow	-
Exception flow	-
Post conditions	The updated PDF file is stored on the cloud, and is ready to be seen on the front-end.

Use case name	Showing Maps
Actors	Leaflet, Data Providers & Validators
Description	The map is generated by the map provider Leaflet. The map includes information about donation centers, temporary acommodation places, food distribution places, pharmacies, gas stations and more, and where they are on the map. All the information comes from from voluntary individuals.
Data	Open pharmacies, evacuation points, food distribution centers, gas stations, acommodation places, veterinarians.
Preconditions	The information should be available on the sources prior to generation.
Stimulus	User clicks the "map" button on the page.
Basic flow	Step 1: The user clicks the "map" button Step 2: The user is redirected to maps.afebilgi.com Step 3: The map is shown.
Alternative flow	Step 1: The user opens maps.afetbilgi.com.
Exception flow	-
Post conditions	-

Table 6: Changing Language Function

Table 7: Reaching to depremyardim.com Function

Use case name	Changing Language
Actors	User, Static Website
Description	There are millions of people living in the affected ares belonging to different ethnicities and background, and although the main language of the site is Turkish, support for different languages is a must. The project comes in four different languages which the users can choose from: Turkish, English, Kurdish and Arabic.
Data	Visual messages
Preconditions	The translations are done previously.
Stimulus	User clicks the language button on the site
Basic flow	Step 1: The user clicks the language button Step 2: A dropdown menu for language selection is shown Step 3: The desired language is selected
Alternative flow	-
Exception flow	-
Post conditions	The site is now in the desired language.

Use case name	Reaching to depremyardim.com
Actors	User, Static Website
Description	The user wants to help the people affected by the earthquake.
Data	URL of the website.
Preconditions	-
Stimulus	User clicks the related button.
Basic flow	Step 1: The user hovers over the button. Step 2: A description about the website is shown. Step 3: The user clicks the button. Step 4: The user is redirected to depremyardim.com.
Alternative flow	-
Exception flow	-
Post conditions	The user is redirected to depremyardim.com.

Use case name	Reaching to afetharita.com
Actors	User, Static Website
Description	Afetharita.com is a website that provides map based information about the earthquake. The map includes information about the earthquake, the aftershocks, the shelters, the hospitals, the schools and more.
Data	URL of the website.
Preconditions	-
Stimulus	User clicks the related button.
Basic flow	Step 1: The user hovers over the button. Step 2: A description about the website is shown. Step 3: The user clicks the button. Step 4: The user is redirected to afetharita.com.
Alternative flow	-
Exception flow	-
Post conditions	The user is redirected to afetharita.com.

Table 8: Reaching to afetharita.com Function

Use case name	Reaching to deprem.io
Actors	User, Static Website
Description	Deprem.io is a website that users can use to help earthquake victims.
Data	URL of the website.
Preconditions	-
Stimulus	User clicks the related button.
Basic flow	Step 1: The user hovers over the button. Step 2: A description about the website is shown. Step 3: The user clicks the button. Step 4: The user is redirected to deprem.io.
Alternative flow	-
Exception flow	-
Post conditions	The user is redirected to deprem.io.

Table 9: Reaching to deprem.io Function

Use case name	Join the Discord server
Actors	User, Static Website
Description	The Discord server is where the developers of the project develop their projects and communicate. User can join the Discord server.
Data	Link for the Discord server.
Preconditions	-
Stimulus	User clicks the button.
Basic flow	Step 1: The user hovers over the button. Step 2: A description about the Discord server is shown. Step 3: The user clicks the button. Step 4: The user is redirected to join the Discord server.
Alternative flow	-
Exception flow	-
Post conditions	The user can join the Discord server after redirected.

Table 10: Join the Discord server Function

Use case name	City Selection
Actors	User, Static Website
Description	Users may need information about only one city. In order to eliminate unnecessary information about other cities and focus on the desired city could save time for users.
Data	Information about the given city
Preconditions	Information should be available beforehand.
Stimulus	User interacts with the dropdown menu.
Basic flow	Step 1: The user clicks "Select a city" button. Step 2: A dropdown menu is shown. Step 3: The user selects the desired city. Step 4: The information is filtered for the selected city.
Alternative flow	-
Exception flow	-
Post conditions	All the information shown on the home page is about the selected city.

Table 11: City Selection Function

3.3 Usability Requirements

- A user shall use the system when a network connection is available.
- A user shall be able to find the needed information in at most 3 steps.
- All users shall be able to navigate through the system without basic computer knowledge.
- A user shall be able to use the system on any device and any browser.
- A user shall always be provided with the latest information.
- afetbilgi.com shall be available 24/7.
- afebilgi.com shall be available in the 3 major languages spoken in Turkey plus English.

3.4 Performance Requirements

- All operations shall be completed in less than 3 seconds.

- Web GUI shall use less than 100 MB memory to ensure that application can be run from majority of modern devices.
- Users shall be redirected to a required page such as another website or a form in 3 seconds after clicking a button.
- The user interface shall be interactable for users while another action is being performed.
- The GitHub actions workflow shall be able to continue functioning indefinitely (without any external factors) to ensure that automated plugins can run without problems.

3.5 Logical Database Requirements

There are no database.

3.6 Design Constraints

System concerns to serve users in a cheap and free way with reliable information. Therefore, open source hardware designs and open source software development methods are chosen.

3.7 System Attributes

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam eget libero sollicitudin justo vehicula venenatis quis ut eros. Proin vitae

3.7.1 Reliability

- The system shall be able to provide reliable information to users.
- All of the hardware and software code of the system must be open-source.
- All the data sources must be verified and reliable.

3.7.2 Avaliability

- The site should be available for all platforms.
- The site should be available 24/7.

3.8 Supporting Information

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam eget libero sollicitudin justo vehicula venenatis quis ut eros. Proin vitae.

4 Suggestions for Future Work

4.1 System Perspective

This system can be improved and held ready for future earthquakes and any possible disasters. New data-sources can be added to the system. These data-sources should better be verified and reliable. With reliable data-sources, and good programming interfaces, the need for manual data entry and human verification may be eliminated and the system can be made more reliable and quick.

4.2 External Interfaces

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam eget libero sollicitudin justo vehicula venenatis quis ut eros. Proin vitae.

4.3 Functions

- **Get data from verified information source**: System fetches data from the verified information sources automatically.

4.4 Usability Requirements

- **User-friendly interface**: The system should have a better and more user-friendly interface. The interface should be easy to use and understand.
- **Easy to use**: The system should be easy to use. The user should be able to use the system without any prior knowledge.
- **Offline mode**: In disaster situations, the network can totally be down. The system should be able to work in offline mode. A PWA (Progressive Web App) can be used to make the system work offline. Native mobile apps can also be developed.

4.5 Performance Requirements

- **Low network usage**: The system should only transfer data when necessary. In disaster situations, the network may be restricted, the system should be able to work with limited network usage.
- **Fast response time**: In disaster situations, time is a crucial factor. The system should be able to respond quickly to the user's requests.

4.6 Logical Database Requirements

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam eget libero sollicitudin justo vehicula venenatis quis ut eros. Proin vitae.

4.7 Design Constraints

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam eget libero sollicitudin justo vehicula venenatis quis ut eros. Proin vitae.

4.8 System Attributes

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam eget libero sollicitudin justo vehicula venenatis quis ut eros. Proin vitae.

4.9 Supporting Information

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam eget libero sollicitudin justo vehicula venenatis quis ut eros. Proin vitae.