



# Software Requirements Specification for **Afet Bilgi**

Emre Geit, Baran Yancı

April 23, 2023

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
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 . . . . .	4
1.3.4	Limitations . . . . .	5
1.4	Definitions . . . . .	5
<b>2</b>	<b>References</b>	<b>5</b>
<b>3</b>	<b>Specific Requirements</b>	<b>5</b>
3.1	External Interfaces . . . . .	5
3.2	Functions . . . . .	5
3.3	Usability Requirements . . . . .	10
3.4	Performance Requirements . . . . .	11
3.5	Logical Database Requirements . . . . .	11
3.6	Design Constraints . . . . .	11
3.7	System Attributes . . . . .	12
3.8	Supporting Information . . . . .	12
<b>4</b>	<b>Suggestions for Future Work</b>	<b>12</b>
4.1	System Perspective . . . . .	12
4.2	External Interfaces . . . . .	12
4.3	Functions . . . . .	12
4.4	Usability Requirements . . . . .	12
4.5	Performance Requirements . . . . .	13
4.6	Logical Database Requirements . . . . .	13
4.7	Design Constraints . . . . .	13
4.8	System Attributes . . . . .	13
4.9	Supporting Information . . . . .	13

# 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

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 <code>depremyardim.com</code>	User can click a link to reach to <code>depremyardim.com</code> .
Reach to <code>afetharita.com</code>	User can click a link to reach to <code>afetharita.com</code> .
Reach to <code>deprem.io</code>	User can click a link to reach to <code>deprem.io</code> .
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**

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

### **1.3.4 Limitations**

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

## **1.4 Definitions**

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

## **2 References**

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

## **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

<b>Use case name</b>	See PDF About a City
<b>Actors</b>	User, Static Website, PDF Viewer
<b>Description</b>	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.
<b>Data</b>	PDF file about the city
<b>Preconditions</b>	The PDF for the city file must be refreshed previously.
<b>Stimulus</b>	User clicks on the "PDF" button and picks a city
<b>Basic flow</b>	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
<b>Alternative flow</b>	-
<b>Exception flow</b>	-
<b>Post conditions</b>	User is redirected to the PDF viewer

**Table 2:** See PDF About City Function

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

**Table 3:** Kızılay Blood Donation Places Function

<b>Use case name</b>	Generating PDF Documents
<b>Actors</b>	GitHub Actions, automation code, Data Providers & Validators
<b>Description</b>	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.
<b>Data</b>	Open pharmacies, evacuation points, food distribution centers, gas stations, accommodation places, veterinarians.
<b>Preconditions</b>	The information should be on Google Sheets prior to generation.
<b>Stimulus</b>	GitHub actions runs this task periodically.
<b>Basic flow</b>	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.
<b>Alternative flow</b>	-
<b>Exception flow</b>	-
<b>Post conditions</b>	The updated PDF file is stored on the cloud, and is ready to be seen on the front-end.

**Table 4:** Generating PDF Documents Function

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

**Table 5:** Showing Maps Function

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

**Table 6:** Changing Language Function

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

**Table 7:** Reaching to depremyardim.com Function



<b>Use case name</b>	Reaching to afetharita.com
<b>Actors</b>	User, Static Website
<b>Description</b>	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.
<b>Data</b>	URL of the website.
<b>Preconditions</b>	-
<b>Stimulus</b>	User clicks the related button.
<b>Basic flow</b>	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.
<b>Alternative flow</b>	-
<b>Exception flow</b>	-
<b>Post conditions</b>	The user is redirected to afetharita.com.

**Table 8:** Reaching to afetharita.com Function

<b>Use case name</b>	Reaching to deprem.io
<b>Actors</b>	User, Static Website
<b>Description</b>	Deprem.io is a website that users can use to help earthquake victims.
<b>Data</b>	URL of the website.
<b>Preconditions</b>	-
<b>Stimulus</b>	User clicks the related button.
<b>Basic flow</b>	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.
<b>Alternative flow</b>	-
<b>Exception flow</b>	-
<b>Post conditions</b>	The user is redirected to deprem.io.

**Table 9:** Reaching to deprem.io Function

<b>Use case name</b>	Join the Discord server
<b>Actors</b>	User, Static Website
<b>Description</b>	The Discord server is where the developers of the project develop their projects and communicate. User can join the Discord server.
<b>Data</b>	Link for the Discord server.
<b>Preconditions</b>	-
<b>Stimulus</b>	User clicks the button.
<b>Basic flow</b>	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.
<b>Alternative flow</b>	-
<b>Exception flow</b>	-
<b>Post conditions</b>	The user can join the Discord server after redirected.

**Table 10:** Join the Discord server Function

<b>Use case name</b>	City Selection
<b>Actors</b>	User, Static Website
<b>Description</b>	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.
<b>Data</b>	Information about the given city
<b>Preconditions</b>	Information should be available beforehand.
<b>Stimulus</b>	User interacts with the dropdown menu.
<b>Basic flow</b>	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.
<b>Alternative flow</b>	-
<b>Exception flow</b>	-
<b>Post conditions</b>	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.

### 3.8 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.