
Software Requirements Specification

for

SarajeWar

Version 1.2

Prepared by Oleksandr Cherednychenko

International University of Sarajevo

5.11.2017

Table of Contents

Introduction	1
User Requirement Definition	1
System Architecture.....	2
System Requirements Specification	3
Functional Requirements.....	3
Non-Functional Requirements	6
System Evolution.....	7

Revision History

Name	Date	Reason For Changes	Version
Rev1.0	5.11.17		Initial version
Rev1.1	10.11.17	Functional requirements updated, release plan added	Second draft
Rev 1.2	9.01.18	Adjusted according to the actual implementation	Final draft

Introduction

Information technology is rapidly changing our lives, redefining the way we interact with the outside world. This could be especially noticeable in the ways educational institutions, such as schools, universities, or museums, fulfill their goals in the era of information. As for the museums, they are at the forefront of change, augmenting traditional exhibitions with digital devices. Overall, it could be said that museums are now to cater to the new generations of users, focused not on observing exhibits, but on interacting with them whenever possible.

This project is addressing the present-day lack of visitor engagement at the Historical Museum of BiH (Historijski Muzej BiH) by proposing an interactive map of Sarajevo Siege (Opsada Sarajeva). The system is designed to serve its visitors both on museum's website and on premises, depicting key locations on the interactive map. Since it could be said that the museum primary focuses on foreign visitors, the interactive map should be available in English language, and it should provide a comprehensive cartographic narrative of the events that have taken place during the Siege of Sarajevo. The system is to be developed as a standalone application, and is expected to improve the overall intuitiveness of the Historical Museum of BiH and raise awareness of the designated historical period.

Implementation of the system should further equip the museum with the tools, instrumental for transitioning to the digital age. Additionally, it is also expected to improve the visitors experience and make the exhibition more compelling.

User Requirement Definition

SarajeWar is expected to deliver a comprehensive visual narrative of the Siege of Sarajevo. It is intended to do so by offering users an interactive map of Sarajevo, with map markers, identifying locations of key events, crucial locations, damaged residential blocks and Sarajevo Roses. Markers are to be designed clickable, with pop-up windows being available to users after clicking on designated markers. Pop-up windows are to contain extensive textual and visual information, relevant to selected location. Additionally, the application is expected to allow search of locations, so that users would be able to navigate around Sarajevo via the geocoding service. As for the markers, they are also supposed to be clustered to avoid potential visual overlap and to ensure the clean look of the map. Markers are expected to be managed by the museum staff, and CUD (create, update, delete) operations should be restricted only to authorized users through basic HTTP authentication.

For the system to be easily accessible to the users, it is expected to be developed as a web application, hosted on a cloud platform. SarajeWar should support up to 50 simultaneous users, and is to be available with 98% SLA (30 minutes/day of potential downtime). System also should be developed as an open source app, and could handed over to the Historical Museum upon completeness.

System Architecture

The system is to be designed as a web application, suitable for deployment to the cloud computing platform. Thus, it should follow a standard 3-tier architecture, with clients (web browsers) acting as thin clients and interacting with the deployed application. It should also be noted that the application is to be accessed with both mobile and desktop versions of web browsers. Since hosting is to be provided by a PaaS (platform-as-service) solution, no specific hardware constraints are present. As for the nature of the data, location details are supposed to be stored in the database and obtained in the JSON format. The architecture diagram is represented on Figure 1 – Architecture.

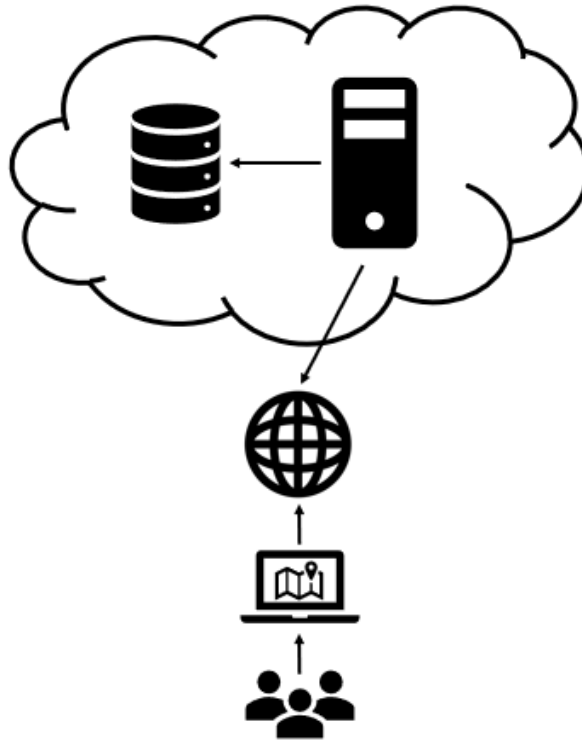


Figure 1 – Architecture Diagram

System Requirements Specification

Functional Requirements

Code	FR1
Name	Open the map
Description	The system shall display a map to the user
Pre-condition	Application is opened
Post-condition	Map of Sarajevo is displayed
Basic flow	1. User opens the browser and navigates to homepage URL 2. Map of Sarajevo is loaded and centered
What can go wrong	If the title provider is unavailable, map would not load properly.
Priority	High

Code	FR2
Name	Display markers
Description	The system shall display markers on the map
Pre-condition	Map of Sarajevo is loaded
Post-condition	User sees markers on the map
Basic flow	1. User opens the application 2. Locations and icons are loaded from the database 3. Markers are displayed on the map
What can go wrong	If marker data could not be obtained from the database and encoded in JSON, empty map is going to be displayed instead.
Priority	High

Code	FR3
Name	Open the pop-up window
Pre-condition	Markers are loaded and displayed on the map

Post-condition	User is presented with a pop-up window
Basic flow	<ol style="list-style-type: none"> 1. User clicks on the required marker 2. Pop-up window is displayed 3. Textual and visual materials, relevant to the location, are available in the pop-up window 4. Map is centered on the clicked marker
What can go wrong	Pop-up window could be not attached to a certain marker, textual/visual information might not load or be unavailable
Priority	High

Code	FR4
Name	Close the pop-up
Pre-condition	Pop-up window is opened
Post-condition	Pop-up window is closed, only markers are available to the user.
Basic flow	<ol style="list-style-type: none"> 1. User clicks on the area of the map, not taken by the pop-up window 2. Pop-up window disappears
What can go wrong	It is possible for the window not to disappear but to still occupy certain region of the map
Priority	High

Code	FR5
Name	Admin login
Pre-condition	Application is loaded, user clicks on “manage markers” button
Post-condition	Admin is authorized and has access to CUD functionality of markers
Basic flow	<ol style="list-style-type: none"> 1. User opens application 2. User enters admin credentials (login and password) 3. User is granted access to the administrator interface and is allowed to perform CUD (create, update, delete) operations
What can go wrong	If user enters wrong credentials, he is offered a re-login functionality. If user cancels the authentication, “Access denied” message is shown.
Priority	High

Code	FR6
-------------	-----

Name	Manage markers
Pre-condition	User has access to the admin interface (is authorized)
Post-condition	Data about markers is altered
Basic flow	<ol style="list-style-type: none"> 1. User utilizes admin interface 2. User selects marker to be added, updated or deleted 3. User creates, adds or deletes the marker 4. Locations (markers) on the map are updated accordingly
What can go wrong	To eliminate accidental deletion, user is shown a confirmation window prior to the marker being deleted
Priority	High

Code	FR7
Name	Search for real-world locations (via geocoding)
Pre-condition	Application is opened, search bar is displayed
Post-condition	Map is centered on a required location
Basic flow	<ol style="list-style-type: none"> 1. User types the required location's name 2. Search is performed via geocoding plugin 4. Map is centered on the most relevant location (matching the entered one)
What can go wrong	If the location is not present, map focus and zoom are not changed
Priority	Medium

Code	FR8
Name	Cluster markers
Pre-condition	Application is opened, markers are displayed
Post-condition	Markers are grouped in clusters
Basic flow	<ol style="list-style-type: none"> 1. Markers are obtained from the database 2. Markers are displayed on the map and combined into clusters when they are located too close to each other
What can go wrong	Clusterization is not operational – markers are not in clusters
Priority	Medium

Code	FR9
-------------	-----

Name	Center map
Pre-condition	Application is opened and focused on specific location
Post-condition	Map is centered on initial location (43.8562586, 18.4130763 with zoom of 14)
Basic flow	1. User clicks on the “Center” button 2. Zoom and focus of the map are changed 3. Map is centered on the initial location
What can go wrong	If map is already centered on initial location – nothing happens
Priority	Medium

Non-Functional Requirements

Code	NF1
Name	Cloud deployment
Category	Deployment
Description	The system should be deployed to the cloud-based computing platform, preferably Heroku

Code	NF2
Name	Open-source license
Category	Licensing
Description	The system should be provided with a non-restrictive open-source license

Code	NF3
Name	Uptime
Category	Availability
Description	The system should be available for usage with 98% SLA (30 minutes of downtime per day)

Code	NF4
Name	User capacity
Category	Load

Description	This system must perform adequately with maximum 50 concurrent users during peak periods.
--------------------	---

Code	NF5
Name	Resource constraints
Category	Hardware
Description	The system should be adequately working on a server with the following characteristics: 512 MB RAM, 20 Gb HDD, 1CPU, Linux-based OS

Code	NF6
Name	Map views per month
Category	Scalability
Description	The system should utilize a free mapping service and generate up to 75.000 map views per month.

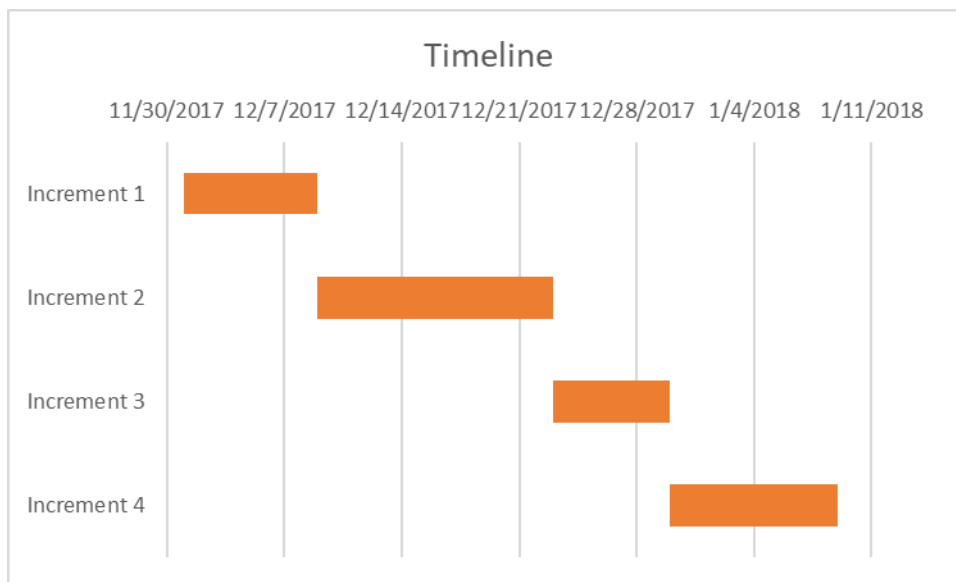
Code	NF7
Name	Intuitiveness
Category	Usability
Description	The system should be suitable for non-experienced users without any prior training

Code	NF6
Name	Database
Category	Software
Description	The system should be using an open-source relational DBMS

Release Plan

Requirement	Duration	Increment	Priority	Dependencies	Release
FR1	2 days	1	High		1
FR2	2 days	1	High	FR1	1

FR3	2 days	1	High	FR2	1
FR4	2 days	1	High	FR3	1
FR5	1 week	2	High		1
FR6	1 week	2	High	FR5	1
FR7	1 week	3	Medium		1
FR8	5 days	4	Medium	FR2	1
FR9	5 days	4	Medium	FR1	1



System Evolution

Since the system is expected to be generating up to 75.000 map views per month, free-tier cartographic service could be used without limitations. However, if the applications is to generate more map views, selection of map titles should be performed with pricing and response time constraints in mind.

Currently, the system is expected to be operating on a free-tier cloud computing platform, with certain limitations on performance and bandwidth. If the application is about to expand, free-tier cloud platform should be upgraded according to the anticipated usage.