"OUR Antiquity Land" Web Application

Software Requirements Specification (SRS)

Version 1.0

06/03/2023

Prepared by: Fatmeh Foqara

Table Of Content

(Click on the title to go to the content)

[1.0. Introduction](#introdoction)

[1.1. Purpose](#purpose)

[1.2. Glossary](#glossary)

[1.3 Stakeholder requirements](#stakeholderrequiremenst)

[1.4. User requirements](#userrequirements)

[2.0. Overall Description](#overalldescreption)

[2.1. System Environment](#SystemEnvironment)

[2.2. functional requirements](#functional)

[2.3. User Characteristics](#userCharacter)

[2.4. non-functional requirements](#nonFunctonal)

[2.5. User Interface requirements](#Userinterfaceequirements)

[2.6. User interaction requirements](#Userinteractionrequirements)

[2.7. System requirements](#systemrequiremenst)

[2.8. Technical requirements of the system](#Technicalrequirementsofthesystem)

[2.9. Performance requirements of the system](#Performancerequirementofthesystem)

[2.10. Society requirements](#Societyrequirements)

[2.11. Compliance requirements](#Compliancerequirements)

[2.12. Acceptance requirements](#Acceptancerequirements)

[3.0. Test Cases](#testcases)

[3.1. STD document](#stdDocument)

[3.2. Bug Report](#bugreportDocument)

**1.0. Introduction**

***1.1.*** ***Purpose***

The purpose of this document is to give a detailed description about the Web App "Our Antiquities Land". It explains the purpose and the features of the web, and what the system will do. This document is intended for both stakeholders and developers of the system.

***1.2. Scope***

This software system will be a Web of suggested archaeological sites in Israel. It will give the viewer information about those suggested sites like name, general information, and an instruction to search this site on waze. A picture of the site will be displayed too.

This web is so simple and attended to allow the user/viewer to add, remove or update a specific site. It also allows him to add his reviews, note and suggestions about a specific site.

***1.3. Glossary***

|  |  |
| --- | --- |
| **Term** | **Definition** |
| User | Reviewer or any one can reach the web link. |
| Stakeholder | Any person with an interest in the project who is not a  developer.  Any person who is interested with the project but not a developer. |
| Database | Collection of all the information monitored by the system. |
| Site | The main unit on the website. |
| SRS | A document that describes all the functions of the system. |
| Python | The programming language that used in this project. |
| Flask | Web Framework that used in the projects. |
| SQLite | Database |
| CSS | Style sheet language that used to design the web. |
| HTML | A markup language that used to build the skeleton of the web. |

***1.4. Stakeholder requirements***

Stakeholdersrequired the web application to be build with python programming language integrated with flask web framework. The user of the web must have the ability to add, delete, update items (sites) and to add their own reviews.

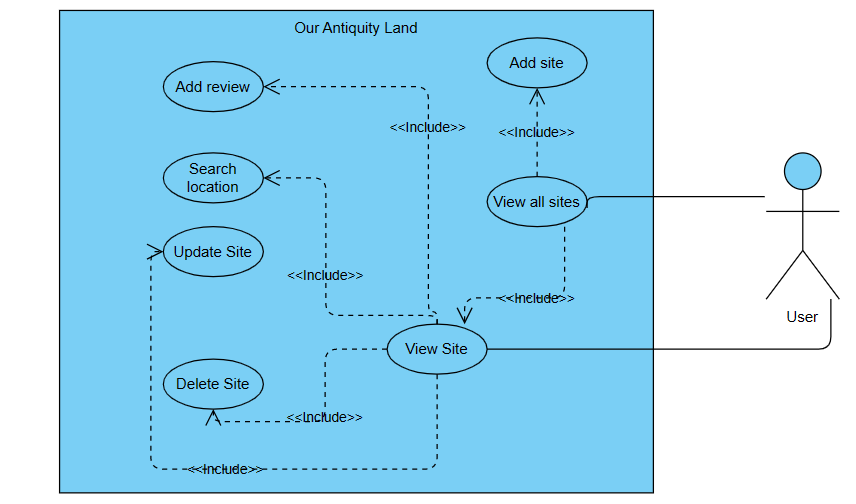
***1.5. User requirements***

User should be able add, delete, and update information about specific element (site) in the web, and to add his own review.

**2.0****. Overall Description**

***2.1.*** ***System Environment***

The web has only viewer/user role, who can add, update, delete and add reviews to a site. He can also view the site and search for its location.



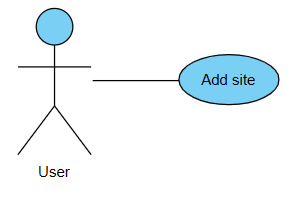
**Figure 1-System Environment**

***2.2.*** ***Functional Requirements Specification***

This section explains the previous use case step by step. In this case we have only one player in the system, which is the user, who can make many things.

2.2.1 User- Add site

**Use case: add site** 🡪 included in view sites (which is not a function, but it is simply viewing the site's main page).



**Description**: The user can add new site to the web with the data he wants.

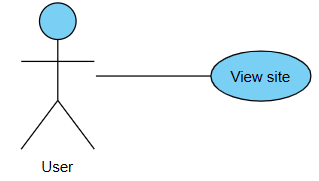
**Initial Step-By-Step Description:**

Before adding new site, the user already is viewing the main page which include all the sites:

1. user click add button in the navigation bar in the main page, then he will be delivered to another page with add form to fill.
2. user fill the form with the data he wants.
3. user click add button and will be delivered to the main page where he can see the sites with his new one.

2.2.2 User- View site

**Use case: view site** 🡪 included in view all sites, there the user can chose site to view.



**Description**: the user can view the site he wants and read about it and more.

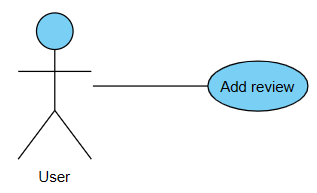
**Initial Step-By-Step Description:**

Before viewing the wanted site, the user is already in the main page:

1. User click on the picture of the site he wants.
2. User is delivered to another page, which is the page of the chosen site.

2.2.3 User- Add review

**Use case: add review**  🡪included in view site, user can add his review about a site only after entering to the site's page.



**Description**: user can add his own review/comment about a specific site.

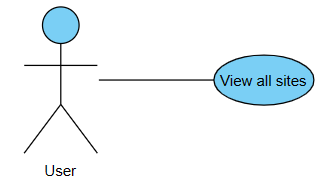
**Initial Step-By-Step Description:**

Before adding his review, the user will be in the chosen site's page:

1. User fill the form of reviews with his name and his comment.
2. User click add review button, and the page will be refreshed with the added review.

2.2.4 User- View all sites

**Use case: view all sites**  🡪 the user can see all the sites in the web in the main page, add site in extended from this use case.



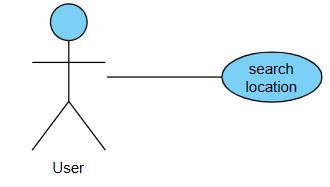
**Description**: the user can see all the sites in the web in the main page.

**Initial Step-By-Step Description:**

The only step in this use case is to run the project, the main page will be shown automatically, and the user can see all the sites in this web.

2.2.5 User- Search location

**Use case: search location**  🡪 This use case in included in view site's use case. User can search the location of the site he is viewing.



**Description**: The user can search the location of the specific site he want via the waze map.

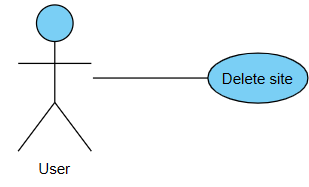
**Initial Step-By-Step Description:**

Before start searching, the user is already in the wanted site's page:

1. Users fill the waze search box with the data of the site's location from the data of the site.
2. The waze automatically will show the site's location on the map.

2.2.6 User- Delete site

**Use case: delete site**  🡪 This use case in included in the use case of view site, only after viewing the chosen site he can delete it.



**Description:** user can delete the site he is viewing by clicking on the delete button in the navigation bar.

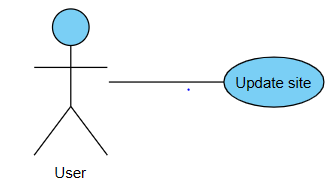
**Initial Step-By-Step Description:**

Before deleting a site, the user is already in the site's page:

1. The user clicks on delete button in the navigation bar.
2. The site will be deleted, and the user will be delivered to the main page where he can see that the site is really deleted. The reviews about the site will be deleted also.

2.2.7 User- Update site

**Use case: update site**  🡪 This use case in included in the use case of view site, only after viewing the chosen site he can update it



**Description:** user can update the site he is viewing by clicking on the delete button in the navigation bar.

**Initial Step-By-Step Description:**

Before updating a site, the user is already in the site's page:

1. The user clicks on update button in the navigation bar.
2. The user will be delivered to a new page with a form to update the site.
3. After filling the wanted updating data, the user will click on the alter/update button.
4. The web will take the user to the site's page again where he can see the new updated site with the new data.

***2.3.*** ***User Characteristics***

The only characteristic requires is that the user should be able to use the web even without technical ability.

***2.4.*** ***Non-Functional Requirements Specification***

* Performance: The web shall response in a short time, not more than five seconds.
* Security: The system shall be able to save the changes od the functionalities that occur in the web correctly.
* Scalability: The web shall be able to handle many users to doing any functionality.
* Availability: The web is available all the time to use.
* Usability: The system should be easy to use even for non-technical people.
* Compatibility: The web should be able to work on different browsers and systems.
* Reliability: The system should have no failures and to recover failures as quickly as possible if they found.
* Maintainability: The system must have the ability to be maintained by the developers if needed.
* Compliance: The system should meet the legal requirements like accessibility standards.

***2.5******. User interface requirements:***

* The user should have the ability to navigate between pages and in the same page.
* Consistency of elements like buttons, links of navigation bars, or logo link.
* The system is accessible for everyone.
* The web has a consist design to make it easer for the user to use it.

***2.6.*** ***User interaction requirements:***

The only interaction that users can make in the web are some major functions like add, update, delete and add review for a site.

***2.7.*** ***System requirements:***

*Note: Those* *requirements has been decided according to the development environment of the system.*

* Hardware: 8 GB and 64-bit to run smoother
* Operating system: Windows 10 or more.
* Database: SQLite
* Network: the web requires stable and strong networking.
* Performance: The system has to response in a short time, with a good performance.

***2.8.*** ***Technical requirements of the system:***

See: [system requirements](#systemrequiremenst)

***2.9.*** ***Performance requirements of the system:***

See: [Non-Functional Requirements](#nonfunctiona)

***2.10.*** ***Society requirements:***

The system should be accessible for anyone, the web application is simple and does not have more social aspects.

***2.11.*** ***Compliance requirements:***

The system should be accessible for anyone, the web application is simple and does not have more complianceaspects.

***2.12.*** ***Acceptance requirements:***

The system must meet all the [functional](#functional) and [non-functional](#nonfunctiona) requirements, and the [user requirements](#userRequirements).

**3.0. Test Cases**

***3.1.*** ***Test cases document (STD):***



***3.2.*** ***Bug Report document:***

