



Software Engineering for Geoinformatics

Design Document and Test Plan

Web-based Visualization of Attraction Points of Milan

Authors:

Hakan Kınıklı

Rishikesh Miriyala

May 24, 2022

Table of Contents

Introduction	3
Purpose.....	3
Scope.....	3
Overview	3
Executive Summary.....	4
System Overview	4
Main User Activities	4
System Architecture and Architecture Design	5
Software Architecture	5
Hardware Architecture	5
Design Interface	7
Structure of Code	8
Future Contingencies	8
Test Plan	9
Introduction	9
Features to be Tested.....	9
Risks and Contingencies	12
References	12

Introduction

This project is concerning about providing information to users about major tourist attractions and to let them have access to sample datasets of our interest (place Italy), contribute on data collection and data validation, and to make custom visualization on these data and get used of some analysis tools offered by a **desktop web- based application**.

Purpose

The purpose of the application is to provide the technical and detailed information about the services delivered by the application. It is to provide the guidance to the architecture of the software. The Design document aims to present the implementation of the hardware and software components as well as the testing plan.

Scope

The aim of the project is to locate the major tourist attractions of the city and its description for the users, by providing a useful visualization and analysis tools in a desktop web-based application interface, using google maps or open street maps. It allows user to put on their own views about the attraction points by registering themselves.

Overview

The interface allows the user to visualize all the tourist attraction points in the city. It includes picturesque description of that place by using a web-based application,

The registered user has access to make changes in the location. Each modification allocates few points for task. In the end, will be rewarded by points or ranking if he reaches up-to some limit.

Executive Summary

The web-application will offer maps of dataset with the ability of the manipulation of the data and to add comment to database. Also, the web- application will allow user to access and find information about co-ordinate regarding to dataset analysis to be displayed as points (To that user).

System Overview

The general overview of the system provides the relation between the inter-relationship between the system components and our vision to relate everything to one system.

The system allows the user to identify the tourist locations in the Milan using a web interface. The registered user has the accessibility to update the description of the attraction point and a score is rewarded to them. The user searches the location in the web application then the existing data is mined from the database using APIs through web server. The entered data is collected and stored in the database for future references.

Main User Activities

- The client accesses the database using web browser and searches for a tourist location (could be historic, natural areas, beaches, or city destinations).
- An overview of the location sites is visible to the client along with the description and the map view.
- The client can register him/herself by using name, surname, and email id.
- The client visualizes the description and comments added by other users.
- The registered user can modify the description which is already in the database.

System Architecture and Architectural Design

The system architecture is the conceptual model that defines the structure, behavior, and views of the system. It consists of components and sub-components of system that work together to implement the entire system.

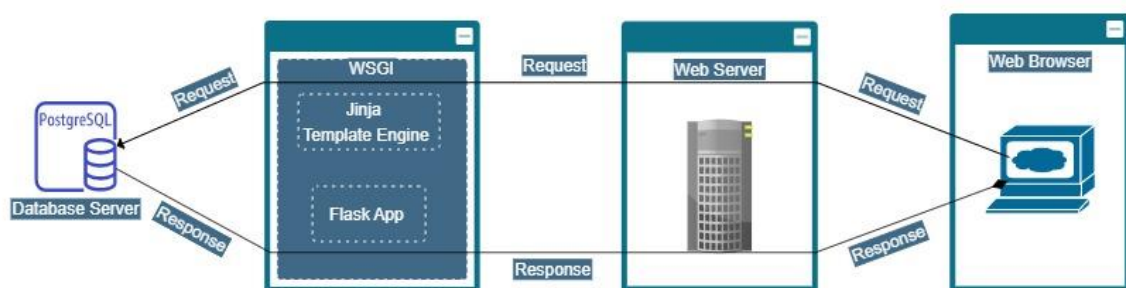


Fig 1: System components

Software architecture

Software architecture functions as the blueprint of the project. It refers to the structure of the software consisting of the relation among the elements.

As we can see a location relates to the subcategory of tourist attraction and as well as the city.

Hardware architecture

The hardware architecture refers to the identification of a system's physical components and their inter-relationships. It allows designers to understand how their components fit into a system architecture and provides to software component designers important information needed for software development and integration.

Structure of database

The database system that allows to data stored in a database and provides an easy and effective method of –

- Defining the information
- Storing the information
- Manipulating the information
- Protecting the information from system crashes or data theft.
- Differencing access permissions for different users.

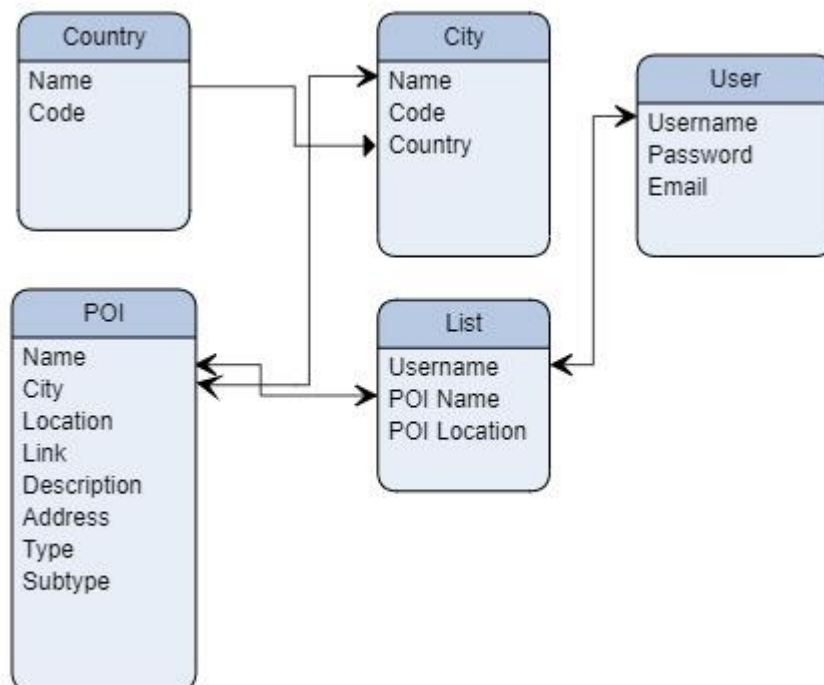
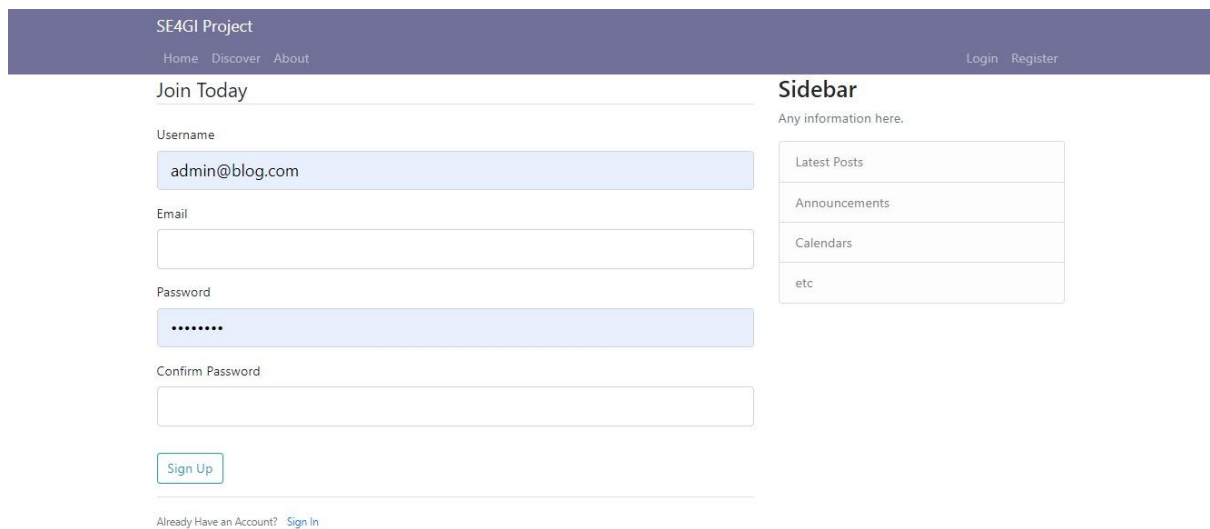


Fig 2: Database components

Design Interface

The sample interfaces are shown below:



The Admin Login page features a dark purple header with the text "SE4GI Project" on the left and "Login Register" on the right. Below the header, the page is divided into two main sections. On the left, under the heading "Join Today", there is a form with four input fields: "Username" (containing "admin@blog.com"), "Email", "Password" (with masked characters "*****"), and "Confirm Password". A "Sign Up" button is located below the "Confirm Password" field. At the bottom of this section, there is a link "Already Have an Account? Sign In". On the right, under the heading "Sidebar", there is a sub-header "Any information here." followed by a table with four rows: "Latest Posts", "Announcements", "Calendars", and "etc".

SE4GI Project	
Home	Discover About
Login Register	
Join Today	
Username	
admin@blog.com	
Email	
Password	

Confirm Password	
Sign Up	
Already Have an Account? Sign In	
Sidebar	
Any information here.	
Latest Posts	
Announcements	
Calendars	
etc	

Fig 3: Admin Login page



The Home page for common users features a dark purple header with the text "SE4GI Project" on the left and "Login Register" on the right. Below the header, the page is divided into two main sections. On the left, under the heading "Discover page", there is a large empty space. On the right, under the heading "Sidebar", there is a sub-header "Any information here." followed by a table with four rows: "Latest Posts", "Announcements", "Calendars", and "etc".

SE4GI Project	
Home	Discover About
Login Register	
Discover page	
Sidebar	
Any information here.	
Latest Posts	
Announcements	
Calendars	
etc	

Fig 4: Home page for common users

Structure of the Code

The Structure of the code includes the list of codes we have used in this project for various GUIs. The Description of Codes used for the interfaces are as follows:

- Home: It allows the user to search the locations. It has an option to login/ register.
- Register: This interface is used by the user to register him/herself with name, surname, and email id.
- Login: This interface is used for logging in by entering details like username and password. A message is displayed 'login successful'.
- Logout: This displays the user that he is logged out.
- Profile: The interface allows the user to check history of the recent searches as well as the individual score.
- View map: The user can access the location of a point using this mapping interface.
- Home_2: This interface is for the registered users for updating the description of desired location.

Future Contingencies

Future contingencies are contingent statements about the future – such as future events, actions, states etc. To qualify as contingent the predicted event, future event, state, action or whatever is at stake must neither be impossible nor inevitable.

The expected contingencies could be

- The user searched for a place that doesn't exist.
- The user enters the wrong email id while registering.
- The user forgets the login details.
- The description edited by the user could be false or inappropriate.
- The details mentioned the web application are not relevant.

Test Plan

Introduction

The implementation of the application service will be examined. The test to be carried out depends on various factors like the inter dependencies of the modules and complexities. It should be able to discover the errors of the design. It must identify the errors and the possibility to correct them as soon as possible.

Features to be Tested

The possible test cases are mentioned explaining the interaction between user input and expected output by understanding the hypothesis.

Test Case 1:

Input: Users access the registration page, inserts username, email, password and confirms password then clicks sign up button.

Hypothesis: the inserted mail is invalid.

Expected Result: the system informs the user that the email is invalid.

Test Case 2:

Input: User access the registration page, inserts username, email, password, and confirmation of password then clicks on sign up button.

Hypothesis: The inserted username or email belongs to on a pre-registered user.

Expected Result: The system informs user that the user already exists.

Test Case 3:

Input: User access the login page and inserts username xyz4567, password 23456.

Hypothesis: The user is registered in the system with the username hk11515, password 23456.

Expected Result: The system informs the user that the inserted username is not registered.

Test Case 4:

Input: User access the login page and inserts username xyz4567, password 23456.

Hypothesis: The user is registered in the system with the username xyz4567, password 987645.

Expected Result: The system warns user to check inserted username and password.

Test Case 5:

Input: User access the discover page and searches for a place named "zubi piazza".

Hypothesis: The place with the name "zubi piazza" does not exist in the system.

Expected Result: The system informs the user that the place with a given name does not exist.

Test Case 6:

Input: Registered user requests for updating the details of location 'xyz'.

Hypothesis: The description is updated along with the comments.

Expected Result: The system informs the user that the information is updated.

Test Case 7:

Input: User searches a location 'abc'.

Hypothesis: The system allows the user to access the database to show the details of the location 'abc'.

Expected Result: The system informs the user about the location 'abc' and link for the map view.

Test Case 8:

Input: After searching the location 'abc' user clicks on the map link.

Hypothesis: The web application connects the user from home page to mapping interface.

Expected Result: The system indicates the location 'abc' in the map.

Test Case 9:

Input: User registers with entering the name, surname, and email id.

Hypothesis: The registration is allowed, and data is stored.

Expected Result: The system informs the user that registration is successful.

Test Case 10:

Input: User searches a beach.

Hypothesis: The description of the beach is displayed with a link to access the maps.

Expected Result: The system displays the location of the beach and a link to open the map.

Test Case 11:

Input: User enters a place 'ghit' and tries to edit the description.

Hypothesis: The description of the place 'ghit is displayed without editing option.

Expected Result: The system informs the user to register.

Risks and Contingencies

- The system allows the registered users to update the information in the database.
- Although the data could be false or mis-leading. The investigation of the data provided by the registered users is not monitored.
- The Correction of entered data by the users is yet to be implemented.

References

- Lecture Notes SE4G
- You tube tutorials
- [yEd Live \(yworks.com\)](http://yworks.com)
- [Structure of Database Management System - GeeksforGeeks](http://www.geeksforgeeks.org/structure-of-database-management-system/)
- [Hardware architecture - Wikipedia](http://en.wikipedia.org/wiki/Hardware_architecture)