2022

Requirement Analysis and Specification Document

Authors:

- **Mohanad Diab**
- **Majed Alotaibi**
- **Shengshen Li**

Deliverable: RASD

Title: Requirement Analysis and Specification Document

Version: 1.1

Date: 25th of May, 2022

Download page: **github.com/MohanadDiab/SE4GI_Project**

Revision history

Version	Date	Change
version 1.0	20th of April, 2022	first submitted version
version 1.1	25th of May 2022	Version 1.1, based on the comments received from professor Di Nitto

Table of Contents

Project overview

This document's objective is to give technical overview about the requirements and specifications for the development of new web app called "Realty", This project has been made to deemed which Hills village/town has the best quality of living out of Crafers, Stirling and Aldgate. This will determine whether Prince's highway plays a role in the quality of housing.

Dataset used: Epicollect5 - Housing Quality Index Crafers Aldgate Stirling

It will also determine whether the distance to major junctions will increase or decrease housing quality.

The attributes provided for each house include the following:

- Geographic Location
- A photo of the house
- Type of dwelling
- Number of trees (greenery properties)
- Distance to major junctions
- Decibel reading (Measure of sound, describes how noisy a neighborhood is)
- Age of property
- Quality of housing (from domain experts)

Using the provided attributes of the housing dataset, many functionalities will be implemented to achieve the goal of the project: which is to study the effect of the different attributes of each house/neighborhood in the assigned quality of each house in each area.

Furthermore, an extra step will be to build a prediction model based on the existing data and their outcomes to predict the quality of the house/neighborhood for a user who wants to own a house in an area of the areas under the study

Different functionalities will be based to help the users to interact with the web app described in more details in this document.

1 Project Goals and Scope

1.1 Project Goals

The real estate industry is rapidly developing, and the influencing factors are so varied that it is difficult to evaluate the property quality and so on. The goals of this system are to process and analyze the real estate data, by combining WebGIS technology and algorithm analysis, and to complete the prediction model analysis of the real estate quality and visualize the data through front and back-end interaction and GIS spatial data analysis by combining various influencing factors.

1.2 Project Scope

Aiming at the current situation of backward information technology in the real estate appraisal industry, a WebGIS-based real estate appraisal system is designed using open-source GIS platform, Geopandas back-end interaction technology, machine learning algorithms, and OpenLayers visualization method, and the functional design and implementation methods are elaborated on the basis of overcoming the difficulties of traditional WebGIS. It is an attempt to combine GIS technology with real estate industry on the somewhat representative dataset that is available through epicollect5 open source projects.

When analyzing real estate attributes, the main influencing factors considered are the area(neighborhood), greenery, medical, commercial, educational and related geographic conditions (assumed to be the nearest conjunction in the chosen database).

Based on GIS technology, the system is developed by transforming geographic location information into quantitative data information and combining Python flask framework and related APIs.

2 Domain Analysis

2.1 Domain Entities

a. Users

- Citizens
- Specialized: domain experts

b. Database

- The “Housing quality index cafers, Aldgate, Stirling” dataset from epicollect5

2.2 Domain analysis table

Table 1: Domain Analysis

Phenomena	Location	Controlled by
Data Collection	Shared	World
User opens the browser	World	World
User inputs URL	Shared	World
User registration (including user name, password, email address, etc.)	Shared	World
If the username is shorter than 5 digits, an error message is displayed to the user	Shared	Machine
If the user inserts any spaces in the username, an error message will be displayed to the user.	Shared	Machine
If the password is shorter than 5 digits, an error message is displayed to the user	Shared	Machine

If the password includes any foreign characters different from letters and number, an error message is displayed to the user	Shared	Machine
If the user inserts any spaces in the password, an error message will be displayed to the user.	Shared	Machine
If the email provided by the user doesn't have '@' the user will receive an error message.	Shared	Machine
After the user presses the registration button, the system will query the DBMS to check if there are any similar username, if there are, a message will pop up and inform the user to use another username	Shared	Machine
After the registration is done, the user's username, password, and email will be stored in the DBMS.	Machine	Machine
Users Login	Shared	World
After the login process, the system will check and verify the user's credentials	Machine	Machine
If the user's credentials match the database, the system will start the session for the user and redirect the user to the welcome page	Machine	Machine
If the user's credentials do not match the database, the system will display an error message to the user.	Shared	Machine
For the administrators login, a different table in the database containing their credentials will be made.	Machine	Machine
The welcoming page will be rendered based on the function that allows the users to enter the site	Machine	Machine
The database will be queried to pull the housing data for displaying them on the WebGis map	Machine	Machine
The system will use the data along with other Geospatial data manipulation libraries (shapely, Geopandas…etc.) to plot the data.	Machine	Machine

The user presses on one of the points in the map	shared	Machine
Upon clicking, the point will be expanded to display the data specific to that point, via a query to the point ID.	shared	Machine
Display system main interface (including functional module options and map visualization display)	Machine	Machine
Users use functions displayed on the current screen on demand (filters, back to home button…etc.)	Shared	Machine
User presses on the graphs tab	Shared	Machine
The web app will invoke a function to redirect the user to a different url that renders different body for the interface but same page(via Jinja2 engine block inheritance)	Machine	Machine
The system will query the database for the data based on each attribute and plot the data based on their type	Machine	Machine
The system answers to the user's commands of filtering and plotting different types of data, querying the database for the specific data the user requests based on pre-defined filters and options	Shared	Machine
The user clicks on the forecast my location tab	Shared	Machine
The system queries the database for the data related to the machine learning model	Machine	Machine
The system asks the user to input the characteristics of his desired forecast	Shared	Machine
The system applies the clustering algorithm acquired from the forecasting library like scikit-learn	Machine	Machine
The forecasting model is built and stores the clusters in the database	Machine	Machine
The user's data is plugged into the model and receives the score based on the model	Machine	Machine
The final quality index of the user's housing based on their given parameters is shown t the users	Machine	Machine

The model's result is saved in the database in a special placeholder table, designed to keep temporary values until the admins decide what to do with it	Machine	Machine
The admin logs in following the log in process stated for normal users	Shared	Machine
The admin's welcome page is invoked and rendered	Machine	Machine
The data provided by the users is queried from the data base and displayed on the screen for the admin alongside their model results	Machine	Machine
The admins decide if they want to add the data to the permanent database for the model building or if they want to dump them based on domain knowledge	Shared	Machine
In case of approval, the instances are deleted from the temporary table in the PMDS and inserted into the permanent table	Machine	Machine
In case of rejection, the data are simply deleted from the temporary table and inserted into a junk table	Machine	Machine

3 User Cases & Scenarios

In order to explain the software functionalities, this section is going to address an explanation about the actions taken by the software and the user in a list of cases that are useful to explain the internal processes of the application. In this section we describe what is going on from server-side and client-side on when the user cases happen by specifying the different actions that take place in these situations.

Actors:

1. Visitors: To manage this website in an efficient and orderly manner, we don't allow unregister user to use this website. So, all visitors will be redirected to the register/login page.
2. Registered Users: They get the access to most of the functionalities of this website, including sending request to the server and visualize those responding data by map applications and chart/diagram, add their preferred location and see its score with regard to the prediction model.
3. Administrator (Domain experts): They can modify the relevant APIs, inspect the exception handling feedback and check the usability of each functionality.

User Case 1: Normal User Registration

Use case name: NormalUserRegistration.

Actors: Visitors.

For registration, the flow is:

1. User enters page of registration.
2. User fills the preferable username and password.
3. Website inspects username's uniqueness and display the result.
4. Website inspects the rules of password and display the result.
5. Information mentioned above are sent to the server then stored in the database.
6. Alert the registration result, redirect to login page.

Exit condition: The user successfully logs in or registers.

User Case 2: Registered User Login

Use case name: RegisteredUserLogin.

Actors: Registered Users.

For login, the flow is:

1. User enters page of login.
2. User fills the username and password.
3. Information mentioned above are sent to the server then check the correctness of username and password.

Exit condition: Alert the login result, redirect to main page

User Case 3: Administrator User Login

Use case name: AdministratorUserLogin.
Actors: Administrator User
For login, the flow is:

4. User enters page of login dedicated for admins.
5. User fills the username and password which are provided exclusively by the development team.
6. Information mentioned above are sent to the server then check the correctness of username and password.

Exit condition: Alert the login result, redirect to main page.

User Case 4: Data Visualization by Map

Use case name: DataVisualizationMap.
Actors: Registered Users.
Flow:

1. User enters this page.
2. Website send a request to the server, then server responds with the data of the housing dataset stored in the database, specifically the coordinates.
3. The Geodata-frame with the whole available data is generated by the Application Server.
4. Website loads the JSON response, then displays it by a map using WebGis.
5. The Geodata-frame is plotted on the WebGis by a Flask function, invoking the libraries used to plot geospatial data (GeoPandas for example).
6. The user can click on one point of the points displayed on the map representing a single real estate.
7. A request is sent to the server which in turn invokes a function that receives data from the database to pull the instance from the database related to the housing clicked by the user via a unique ID assigned to each housing (the unique ID is already present from the API)
8. The same request and function uses the instance to view a popup that displays the tabular instance of the housing in a user friendly user interface design.

User Case 5: Map Data with a Custom Visualization

Use case name: CustomDataVisualizationMap.
Actors: Registered Users.
Flow:

1. The user enters the user interface for interactive mapping.
2. The whole available raw data which has already been retrieved from the host server, is obtained by the Data retrieval function.
3. The whole available data points from the DBMS are visible in the interactive map by default (with a certain default base map).
4. The user wants to consider the data just within a specific area, and with a specific land use classification, so he/she does the corresponding settings in the friendly user-interface for filtering to keep just the needed data subset.
5. The filtering options depend on the attributes of the dataset used (distance to major junctions, housing type, age of house…etc.).
6. The settings request is taken by the Filter Manager, which performs the logical operations in order to obtain the indexes of the needed data subset.
7. The indexes of the filtered data points are delivered to the Mapping Tool, which use these to obtain only the needed data subset in order to add it to the Map. The page is rendered again with the updated Map (with the filtered data points).
8. The user clicks the Visualization Options Menu and selects a certain style to plot the spatial
9. data (heat maps, etc.). The user request is sent to the Data Analysis Tools.
10. The indexes of the filtered data points are also delivered to the Data Analysis Tools, which perform the logical operations to generate the required data to obtain the visualization with the data filtering.
11. The visualization data is input in the Mapping Tool, which computes and adds the custom visualization to the Figure. As before, the page is rendered again with the updated Map.

Exit condition: When every single factor relevant with geography is displayed on the map and the interactive function with the map is available.

User Case 6: Data Visualization by Charts and Plots

Use case name: DataVisualizationTools
Actors: Registered Users.
Flow:

1. The user enters the user interface for intera6ctive analytical tools.
2. The whole available raw data which has already been retrieved from the host server, placed in the DBMS, is obtained by the Data retrieval function.
3. The whole available data points from the DBMS are visible in the interactive map by default (with a certain default base graphs, plots…etc.).
4. Different operations and filters are available for personalizing these charts, for example the user can filter the data.
5. The settings request is taken by the Filter Manager, which performs the logical operations in order to obtain the indexes of the needed data subset.
6. The filtering information is input in the Data Analysis Tools.
7. Then, the user clicks in the Data Analysis service and selects the option to get Descriptive statistics.
8. The software calls the statistics function which computes and returns the general statistics of each data subset feature.
9. The statistics data is passed to the Template Engine. The page is rendered with the requested.

Exit condition: When charts and plots are correctly generated.

User Case 7: Repeatedly Visualization - Filtering Requesting

Now we consider a case similar to User Case 6, but supposing that the user requests a sequence of data analysis requests (geospatial chart visualization or statistics over filtered data) and filtering updates. If the user does many statistics-filtering requests it is not a problem, since the statistics display will be simply updated and overwritten.

Use case name: RepeatedVisualization
Actors: Registered Users.
Flow:

1. When the Mapping Tool is initialized, the only default layer is the measurement data points (the base map is also visible but not as a user-controlled layer).
2. The user can do a filtering request by setting up the filter manager controls. The Filtering Function will obtain the required data point indexes (As Boolean Series) and send these to the Data Analysis and Mapping Tools.
3. The Mapping Tool will update the Figure by modifying the Data Points layer according to the new filtering. The page is rendered again with the updated figure by the Template Engine.
4. The user can see the updated Data Points layer. Now he/she does a visualization request by specifying which type and submitting the layer name.
5. The Data Analysis Tools will receive the visualization request and the layer name. It will generate the figure data (taking into account the new filtering!) and send it to the Mapping Tool.
6. The Mapping Tool will update the Bokeh Figure by adding the new visualization layer according to the new filtering. The page is rendered again with the updated figure by the Template Engine.

Exit condition: When charts and plots are correctly generated.

User Case 8: Exception Handling

Use case name: ExceptionHandling
Actors: Can happen with all Actors.
Flow:

1. When an exception is detected, the program will catch it and enter the handling phase. Distinguish the exception type.
2. If this exception type was predicted in the program, run the corresponding exception handling function, otherwise redirect to 404 page.

Exit condition: When the exception is handled, and user has been redirected to the right page with clear instruction of dealing with this exception.

User Case 9: Finding rating of an item outside of the database

Use case name: QualityForecast
Actors: Registered User.
Flow:

1. This tool is used to predict the rating of a house entered by the user by means of machine learning algorithms implemented by the web application in Python.
2. When this tool is initialized the forecasting function is invoked.
3. The same data set retrieved for the previous steps will be retrieved for all the data.
4. The machine learning algorithms are imported from sci-kit-learn library and applied to the dataset, using supervised learning algorithms to build a classification model.
5. The user is greeted with a page to enter the data of his housing
6. The data are stored by the server to the database, the system then implements the model on the data to give the classification rank of the housing (the quality).
7. The webpage displays the results to the user.

User Case 10: Approving the housing data

Use case name: DataApproval
Actors: Administrator User.
Flow:

1- The Administrator user is welcomed by this page
2- The page displays the activity of the registered users who have entered data to be forecasted by the machine learning algorithm.
3- The administrator uses his domain experience to deem the quality of the data provided by the user.
4- The administrator also makes sure that the user didn't enter any false instances of the data, also based on his domain knowledge.
5- The domain expert reviews the score given by the model and makes sure it is coherent based on his domain knowledge, he can also change this score based on his view of the situation.
6- If the data is approved by the domain expert, it is entered to the database as a permanent data for the sole reason of increasing the accuracy of the predicting model for the future.

User Case 11: Logging out

Use case name: LogOut
Actors: Administrator User, Registered User
Flow:

1- A user who is already logged in the session presses the logout button.
2- A function in the web app is invoked to log out the user from the session.
3- The user is redirected to the login page.

4 Requirements

4.1 Technical requirements:

T1. The website will be developed in python, HTML, CSS.
T2. The web application will use map visualization using OpenStreetMap
T3. The website will use a REST API from Epicollect5 dataset.

4.2 Functional requirements:

F1. The housing quality index should be used as model building data for the forecasting process via the application of the algorithms on the dataset stored in the DBMS and the data inserted by the users.
F2. The user can compare houses based on the different attributes assigned to them based on the dataset (quality index, housing type…etc.).
F3. The housing prices on the website can be classified into clusters for forecasting purposes.
F4. The users can visualize the data on the map.
F5. The users can sign in as normal users or administrators.
F6. The users will have different visualization options based on the maps (filters, heat maps…etc.).
F7. The users will have the option to visualize the data with graphs and plots to forecast different attributes based on preference.
F8. The visualization techniques will have different filtering options based on time, volume, price…etc.
F9. People can see if the price they are offered is higher or smaller than the price based on forecasts.

4.3 Domain Assumptions:

D1. The house ratings are considered to be placed by domain experts.
D2. The housing ratings are assumed to be final, to be used for forecasting models.