2022

Requirement Analysis and Specification Document

Authors:

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

Deliverable: RASD

Title: Requirement Analysis and Specification Document

Version: 1.0

Date: 20th of April, 2022

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

Revision history

Version	Date	Change
version 1.0	20th of April, 2022	first submitted version
version ...	...	...

Table of Contents

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 prices and so on. The goals of this system are to integrate, process and analyze the real estate price data, by combining WebGIS technology and algorithm analysis, and to complete the prediction model analysis of the real estate prices 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.

When analyzing real estate prices, the main influencing factors considered are traffic, greenery, medical, commercial, educational and related geographic conditions. 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: Real Estate Industry Practitioners
- Administrators

b. Database

- Urban real estate price dataset
- Real Estate Periphery Feature Dataset

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
Fault tolerance mechanism for user registration information	Shared	Machine
User Login	Shared	World
User login information verification mechanism	Shared	Machine
Display system main interface (including functional module options and map visualization display)	Shared	Machine
Users use functions on demand	Machine	Machine

3 Project feasibility analysis

3.1 Technical feasibility

The analysis of the technical feasibility of the system can be carried out in two main aspects. First, the hardware equipment of the system must be analyzed, for example, whether the capacity or speed of the computer can meet the needs of the user. Secondly, the software aspect of the system must be analyzed to examine whether the development platform and database required for the system are in line with the current technology and whether the existing functions can meet the needs of the system.

Based on the above two aspects, the hardware equipment required for the development of this research system does not have special requirements, and the current mainstream computer equipment and server equipment can meet the needs. Secondly, it is the software of the system. The development platform of this system is Anaconda, the development languages are Python, HTML, CSS, Javascript and Web development tools, and the required GIS secondary development interfaces are Openlayer and Echarts.js.

Through the preliminary research on the system, we have already had a deep understanding of the research content and now have the perfect technical requirements. Therefore, this system is technically feasible.

3.2 Economic feasibility

The system developed in this study mainly requires a computer, and the requirements for the computer are not too high, and the mainstream computers in the market are able to meet the performance requirements of this system. At the same time, the development of the system also requires some auxiliary equipment, the cost of these devices is not high, the development of this system is economically feasible.

3.3 Operational feasibility

The system is also user-friendly in terms of interface, and the system can be operated by ordinary Internet users without the need for specialized knowledge. Therefore, this system is operationally feasible.

4 Project operational requirements analysis

4.1 Interface Design Overview

a. Adaptability: The operating interface of the system can adapt itself to the browser front-end of various mainstream laptops and desktop computers.

b. Simplicity: The system is designed for the convenience of users of all ages and is designed to be simple, easy to use, familiar and easy to master, with humanized reminders, friendly language, rich in tips and inspiration.

c. Simplicity: The system has a concise interface and a beautiful visualization display that enables interoperability of users.

4.2 Design Goals

a. The web application will be designed to offer an easy to navigate in environment and will provide clear and understandable user interface (UI) design to ensure the best user experience (UX).

b. To introduce to the users a unique visualization tools, which will be the main feature of the web application; the ability to mold the provided data into something interesting.

c. The site will include an interactive map which in turn would also have a variety of features including filters, sorting, list views, and more.

User Cases & Scenarios

In order to explain the software functionalities, interactions between the components and possible exceptions, 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.
3. Administrator: They can modify the relevant APIs, inspect the exception handling feedback and check the usability of each functionality.

User Case 1: Registration and Login

This Use Case below illustrates the User registering for the first time within the application and the response given. It also illustrates the exception, if the details are not correct. This Use case also illustrates the Login process after registration and the exception, if details are incorrect.

Actors: Visitors.

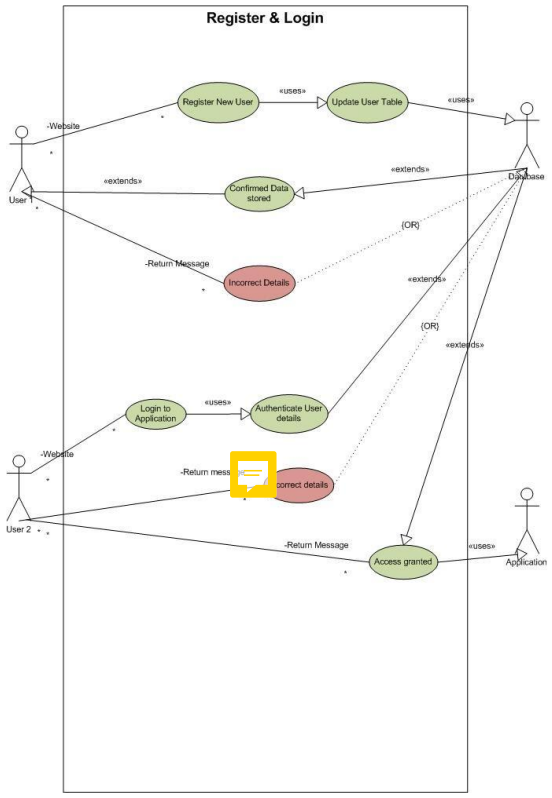
For registration, the flow is:

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.

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.
4. Alert the login result, redirect to main page.

Exit condition: The user successfully logs in or registers.



User Case 2: Discover about Rental Standards - Current Issues

Providing users with a platform to learn about the existing rental standards and current issues is one of the purposes of our project.

Actors: Registered Users.

Flow:

1. User enters this page.
2. Website send a request to the server, then server responds with existing rental standards and current issues.
3. Website loads the JSON response, then displays it by charts and tables.

Exit condition: When existing standards are showed on the web page correctly.

User Case 3: Data Visualization by Map

Through the interactive map, users can obtain the details about each measurement points from designed APIs.

Actors: Registered Users.

Flow:

1. The user enters the user interface for interactive mapping.
2. The whole available raw data is retrieved from the host server, cleaned up (remove strings from numerical values and setting as missing the invalid ones) and placed in the DBMS.
3. From the DBMS, a Geodata-frame with the whole available data is generated by the Application Server.
4. The Geodata-frame is input in the Mapping Tool, which adds it to the Interactive Map.
5. The user places the indicator on a certain measurement point (its geographical position in the map).
6. The pop-up function (tabular data-frame) is activated. Therefore, once the cursor is placed in the user can do the same with each available measurement point.

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

User Case 4: Map Data with a Custom Visualization

The Interactive map is one of the capabilities which is provided on the web-application in the aim of visualizing spatial data. This user case describes how a user can obtain a customized visualization of spatial data.

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, cleaned up and placed in the DBMS) is obtained by the Data retrieval and pre-processing 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, in a certain time window 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 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 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).
7. The user clicks the Visualization Options Menu and selects a certain style to plot the spatial

8. data (heat maps, etc.). The user request is sent to the Data Analysis Tools.
9. 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.
10. 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 5: Data Visualization by Charts and Plots

The user will be able to discover information about Rental standards - current issues also by means of interactive analytical tools (e.g. histograms and other charts). These tools are meant to be interactive
in the sense that the user can choose which data to visualize, for example filtering data by specific land use or time of the day of the surveying. This use case describes the main analytical tools which will be provided by the web application.

Actors: Registered Users.

Flow:

1. The user enters the user interface for interactive analytical tools.
2. The whole available raw data which has already been retrieved from the host server, cleaned up and placed in the DBMS, is obtained by the Data retrieval and pre-processing 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. Different operations and filters are available for personalizing these charts, for example the user can filter the data by land use or by the time of day when the noise pollution measure was taken. In this case the user clicks in the Filter Manager to display just the subset of needed 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 6: Repeatedly Visualization - Filtering Requesting

Now we consider a case similar to User Case 5, 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.
Given that each geospatial chart visualization requests are going to be added to the client-side map, this is an interesting user case. If we consider a scenario in which for each filtering request, each map layer is updated with the new filtering, then it would be very computationally expensive, especially when the number of layers is high. For handling this problem, the previously generated Geospatial Visualizations will not be updated with each filtering request.
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 7 Exception Handling

Exceptions, being an unstructured process, increase the possibility of resource leakage (e.g., escaping from locked code, escaping away when opening a file) and may also lead to inconsistent state. Therefore, centralized exception handling is needed.

Actors: Registered Users

Flow:

7. When an exception is detected, the program will catch it and enter the handling phase. Distinguish the exception type.
8. 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.

Technical requirements:

- The website will be developed in python, HTML, CSS.
- The web application will use map visualization using OpenStreetMap
- The website will use a rest API from Epicollect5 dataset.

Functional requirements:

- The housing prices should be used as model building data or for cross validation.
- The user can compare houses based on location, price, and rating.
- The user should be able to compare real prices with prices provided in the website.
- The housing prices on the website can be classified into clusters for forecasting purposes.
- The users can visualize the data on the map.
- The users can sign in as buyer or seller.
- The users will have different visualization options based on the maps (filters, heat maps…etc.).
- The users will have the option to visualize the data with graphs and plots to forecast different attributes based on preference.
- The visualization techniques will have different filtering options based on time, volume, price…etc.
- People can see if the price they are offered is higher or smaller than the price based on forecasts.

Domain Assumptions:

- The house ratings must be provided by domain experts.
- The housing prices are assumed to be final to be used for forecasting models.