Software Requirements Specification (SRS)

CMSC 447 Group 4

Project Vesta

1 Scope

1 Scope

1.1 Identification

Vesta is a website that locates a user's ideal place to live.

1.2 System overview

Vesta is a web based application that will be utilized by an average internet user. This web application is intended to allow users to acquire potential living locations based on abstract specifications regarding the area. To garner said information, the user will specify quality of life attributes desired within a defined interval. These characters may include meteorological data (ie average temperature, average weather conditions), geographical information, etc. The application will compare the inputted information to specified open source databases to determine the locations that fit the criteria. Once the locations have been determined by the software on the back-end, they will be displayed to the user.

1.3 Document overview

The Software Requirements Specification document will be used to record the progression of this web application. All requirements, including those decided upon in the original meeting with the customer and added through the development of the software, will be recorded in this document. In addition to the requirements being written within the document, the original design decisions and the subsequent changes throughout the softwares development will be documented. There are no privacy or security concerns associated with the distribution of this document.

2 Referenced documents

None at this time.

3 Requirements

3.1 Required states and modes

Vesta will run in the following three states:

1. Input State
    a. During the Input State, Vesta will present a form for the user to fill out.
    b. The Input State is the starting state whenever the Vesta website is initially loaded.
2. Calculation State
    a. When the submit button is pressed by a user, Vesta moves into its calculation state.

b. During the calculation state, Vesta takes all of the user inputted information and finds every location in the United States that meets these requirements.

3. Results State
    a. During the Results State, Vesta displays a map and all locations found by the Calculation State.
    b. The user can interact with both the map and list of locations during this state.

3.2 CSCI capability requirements

3.2.1 (CSCI capability)

Inputs:

I. The application shall allow the user to input a search query through the web interface. (Input State)
    A. The application shall except a variety of inputs.
    B. The application shall except inputs with differing degrees of flexibility.
    C. The application shall except inputs with differing scales.
    D. The application should allow the user to input meteorological search information.
        1. The application should allow the user to select a range of temperatures.
        2. The application should allow the user to select prefered weather conditions.
            a. The user should be able to specify the number of sunny days per year.
            b. The user should be able to select areas with low hazardous weather conditions.
    E. The application should allow the user to input property values.
    F. The application should allow the user to input geographical information.
        1. The user should be able to select proximity of international airports.
        2. The user should be able to select proximity of sporting stadiums.
        3. The user should be able to select proximity of schools.
            a. The user should be able to select the prefered school ratings.
        4. The user should be able to select proximity of hospitals.
        5. The user should be able to select a range of population size.
        6. The user should be able to select proximity to religious landmarks.
        7. The user should be able to select proximity to ethnic communities.
    G. The application should allow the user to search for political parties.
        1. The user should be able to select Blue vs. Red states.
        2. The user should be able to select Blue vs. Red counties.
    H. The application should allow the user to search for health hazards.
        1. The user should be able to search for a history of contaminants in the water.
        2. The user should be able to search for a history of CDC outbreaks.

Outputs :

II. The application shall output the most optimal cities for the user to live based on inputted data. (Calculation State) (Results State)

Output Requirements:

III. The application should take no longer than 60 seconds to return. (Calculation State)

IV. The application should display a set of results for 95% of queries. (Calculation State) (Results State)

3.3 CSCI external interface requirements

3.3.1 Interface identification and diagrams

I. The application shall be able to incorporate multiple datasets as sources of information. (Calculations State)
   A. The application shall be able to incorporate json data.
   B. The application shall be able to incorporate csv data.
   C. The application should be able to incorporate xml data.
   D. The application should be able to incorporate rdf data.
   E. The application shall be able to easily incorporate new data sources.

3.4 CSCI internal interface requirements

There are no internal interface requirements. All internal interface requirements will be left to the design.

3.5 CSCI internal data requirements

There are no internal data requirements. All internal interface requirements will be left to the design.

3.6 Adaptation requirements

There are no current adaptation requirements.

3.7 Safety requirements

There are no requirements concerned with safety for this software.

3.8 Security and privacy requirements

There are no requirements concerned with maintain security and privacy for this software.

3.9 CSCI environment requirements

The software shall run on a Linux operating system.

3.10 Computer resource requirements

3.10.1 Computer hardware requirements

I. The computer should have an Intel Pentium 4 processor or later that's SSE2 capable.

3.10.2 Computer software requirements

I. The software shall be able to run on Google Chrome version 64.*.
II. The software shall be able to run on Mozilla Firefox 58.*.

3.11 Software quality factors

Requirements specific to quality of the software have been previously listed in 3.2:

I. Functions: The software shall be able to accept multiple inputs and produce several locations that match 95% of the search inputs within 60 seconds.
II. Maintainability/Flexibility: The software shall be able to easily updated to ensure that information collected from various databases is correct.
III. Usability: The software shall be intuitive to use, making it easy to learn and utilize.
IV. Availability: When the user has internet access and utilizes the specified web interfaces.

3.12 Design and implementation constraints

There are no current design and implementation constraints.

3.13 Personnel-related requirements

There are no current personnel-related requirements.

3.14 Training-related requirements

There are no current training-related requirements.

3.15 Logistics-related requirements

There are no current logistics-relate requirements.

3.16 Other requirements

There are no other requirements.

3.17 Packaging requirements

There are no current packaging requirements.

3.18 Precedence and criticality of requirements

There are no current precedences of requirements.

4 Qualification provisions

The following qualifications are defined and used below:

1. Demonstration
    a. The operation of Vesta that relies on observable functional operation not requiring the use of instrumentation, special test equipment, or subsequent analysis.
2. Test
    a. The operation of Vesta using test scripts to collect data for further analysis and benchmark realization.
3. Analysis
    a. The processing of accumulated data obtained from other qualification methods.
4. Inspection
    a. The visual examination of Vesta code and documentation.

Requirements	**Qualifications To Be Used**
3.2.1.I	Demonstration
3.2.1.II	Demonstration
3.2.1.III	Test, Analysis
3.2.1.IV	Test, Analysis
3.3.1.I.A-D	Inspection (Code)
3.3.1.I.E	Test (Adding new sources)
3.9	Demonstration
3.10.1	Demonstration
3.10.2	Demonstration