Requirements document Date planner Application

By: Dominic Crofoot, Travis Early, Jordan Fok, Sean Hill, & Nicholas Ho

October 26, 2018

**Revisions:**

* **26Oct2018: First draft**
  + First draft of SRS document created.

**Table of Contents**

**1. Scope**

**1.1 Identification**

**1.2 System overview**

**2. Reference Documents**

**3 . Specific Requirements**

**3.1 Required states and modes**

**3.2 Capability requirements**

**3.3 External Interface requirements**

**3.4 Internal Interface requirements**

**3.5 Internal Data requirements**

**3.6 Adaptation requirements**

**3.7 Safety requirements**

**3.8 Security and privacy requirements**

**3.9 Environment requirements**

**3.10 Computer resource requirements**

**3.11 Software quality factors**

**3.12 Design and implementation constraints**

**4. Qualification Provisions**

**5. Requirements Traceability**

**6. Notes**

1. **Scope**

**1.1 Purpose -**

This document supplies the software requirements for an unnamed date planning website intended for use by anyone who wants to go and do an activity.

**1.2 Scope -**

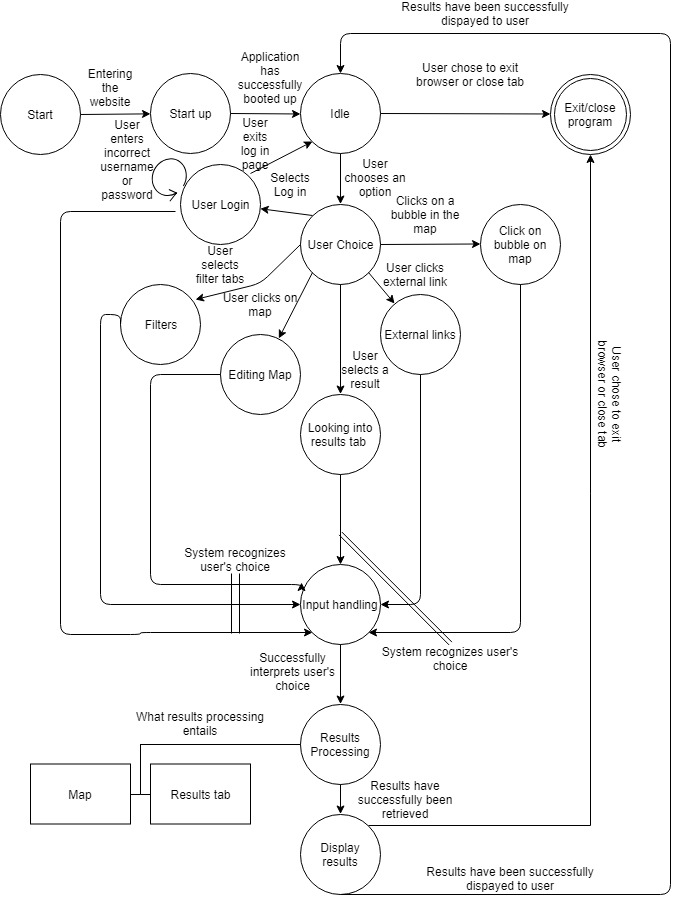
Our yet unnamed web app serves as an all in one location for people to find a location or event to spend time based on a variety of user supplied conditions.

**1.3 Overview -**

Section 1 provides the system overview, which includes details about the web app. It also includes the document overview, detailing what each section contains. Section 2 contains any documents that were referenced when making this document. Section 3 contains all of the requirements for

1. **References**
   1. **Government -**
      1. MIL-STD 498, 5 December 1994
         1. Template for entire SRS document
         2. Link: [MIL-STD 498](https://blackboard.umbc.edu/bbcswebdav/pid-3246270-dt-content-rid-25947413_1/courses/CMSC447_8367_FA2018/SRS.htm)
   2. **Non-Government -**
      1. Loopback
         1. A node.js framework used for back-end services. Loopback will be used by the application for user authentication and retrieving of the users filter history.
         2. Link: <https://loopback.io/>
      2. Google Maps API
         1. Data supplied by google that will be used for filter, results tab, and map.
         2. Link: <https://cloud.google.com/maps-platform/>

1. **Requirements**
   1. **Required States and Modes -**

****

* 1. **Capability Requirements -**

Capability requirements will be expressed in terms of their description, their required input to the software, the processing upon that input, and the output the software will produce as a result of the processing.

**Functional Requirement 1 (FR1):**

The application shall provide users with a variety of date activities when given the required input by the user.

**Input:** A combination of predetermined filters selected by the user.

**Processing:** Uses the input to create a search query within the google API to develop a table of results that align with the given input.

**Output:** A list of activities that satisfy the input filters.

**Functional Requirement 2 (FR2):**

The application shall draw a map with pins denoting the location of activities returned by the software.

**Input:** A combination of predetermined filters selected by the user.

**Processing:** Uses the input to create a search query within the google API to develop a map with pins of the results that align with the given input.

**Output:** A Google maps representation of the results of processing the user input filters through the Google API.

**Functional Requirement 3 (FR3):**

The application shall store the search history of users who are logged into an account.

**Input:** A combination of predetermined filters selected by the user.

**Processing:** Uses SQL procedures to store the values given by the user into an SQL Database located on the UMBC servers.

**Output: N/A**

* 1. **External Interface Requirements -**

**External Interface Requirement 1 (EIR1):**

The application shall query the Google API upon a set of given filters to develop results that align with the filters.

**Input:** A combination of predetermined filters selected by the user.

**Processing:** Uses the input to create a search query within the google API to develop a table of results that align with the given input.

**Output:** A list of activities that satisfy the input filters.

**External Interface Requirement 2 (EIR2):**

The application shall utilize the IBM Loopback framework to develop user accounts and store data using MySQL.

**Input:** A username and password, or a token.

**Processing:** This will be sent to our REST services that interfaces with our MySQL database and query the user’s information with the provided input.

**Output:** profile information and a list of the user’s filter history.

* 1. **Attributes -**

**Availability** The application will be available through the public internet 24 hours a day.

**Security**

The provided security will be a token based system, which will identify a user and retrieve data based off the token.

**Maintainability**

Our services are scoped to a limited amount. This allows us to focus on the few services we provide and if needed we can easily adjust and add new services.

* 1. **Internal Data Requirements -**

The application must maintain the data related to the user. Since the application interfaces with the Google Maps API, it will not be maintaining the search result data. Instead our design revolves around reducing the load of data stored. The application will be maintaining the users profile information, and a filter history of the users searches. The system revolves around the user retrieving their own information. So a transaction should follow all the properties of a database transaction (Atomicity, Consistency, Isolation, Durability).

* 1. **N/A**
  2. **N/A**
  3. **N/A**
  4. **CSCI Environment Requirements**

The CSCI must run on the two most used currently supported web browsers, chrome and firefox.

* 1. **Computer Resource Requirements** 
     1. **Computer Hardware Requirements**

The CSCI can be run on any computer capable of running the web browsers detailed in section 10.9 as well an internet connection.

* + 1. **Computer Hardware Resource Utilization Requirements**

The CSCI is bound by the hardware resource utilization requirements of the web browsers detailed in 10.9.

* + 1. **Computer Software Requirements**

The CSCI requires being connected to a MYSQL database server.

* + 1. **Computer Communications Requirements**

All computer communications will be handled through the web browsers detailed in section 10.9.

* 1. **Software Quality Factors**
     1. All generated lists shall be created with the most up-to-date information from Google’s API.
     2. The CSCI will be available for use at any time.
  2. **Design and Implementation Constraints**

The CSCI is limited by the Google API. It will not be able to manually add search results that are not already available through Google. Conversely, as the Google API improves, our search functionality will provide better results.

Our development is limited by the number of languages supported by Google’s API.

1. **Qualification Provisions -**
   1. **Demonstration**

**This qualification method is carried out by executing the program and observing how the program changes visually without the use of special equipment or instruments**

**4.2 Test**

**A qualification method that is carried out by executing both specific parts of the application and the entire application under standard or specified conditions.**

**4.3 Analysis**

**This qualification method by reviewing and processing collected data such as API data, runtime of application, etc.**

**4.4 Inspection**

**This qualification method is done by visual examination, physically manipulating, or measuring to ensure the requirements have been met.**

**4.5 Special qualification methods**

**This qualification method may be used if inspection within the system and/or application is not enough.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Demonstration** | **Test** | **Analysis** | **Inspection** | **Special Qualification methods** |
| **FR1** | **✔** | **✔** |  | **✔** |  |
| **FR2** | **✔** | **✔** |  |  |  |
| **FR3** |  | **✔** |  |  |  |
| **EIR1** |  |  | **✔** | **✔** |  |
| **EIR2** |  | **✔** | **✔** |  | **✔** |
| **EIR3** | **✔** |  |  | **✔** |  |
|  |  |  |  |  |  |

1. **Requirements Traceability**
   1. This section addresses how various requirements will be tracked throughout the rest of the other documents (Development, testing, etc.) and a simplified explanation of the functional requirements.

**FR1:** The application shall display the same results in the list as are displayed on the map. Every result shall have a map pin associated with it, and vice versa.

**FR2:** Same as FR1.

**FR3:** Logged-in users shall be able to see the same filters they have used previously and shall be able to generate results lists from their previous searches.

1. **Notes (Everyone [as needed], Jordan)**
   1. Acronyms:
      1. CSCI Computer Software Configuration Item
      2. API Application Program interface
      3. MIL-STD Military Standard
      4. N/A Not applicable
      5. REST Representational State Transfer
      6. TBD To Be Determined
   2. Reach Goals:
      1. At the moment, this application is not meant for financial gain, but it has been considered for the future in the form of commercials endorsement or sponsorships.
      2. A feature we considered is a time table/list. The idea is that if someone wants to plan an entire day, they can add results to a time table and create an entire itinerary for a date.
   3. Sections marked N/A:
      1. 3.6 - No extra data will be provided by the system,
      2. 3.7 - No realworld damage will be caused by the application,
      3. 3.8 - Although we have user profiles, until the use of sponsorships is implemented, there is no reason to have safe guards. The login will only be a username and password with no other personal information attached.
   4. Although date planner has already been designed, in our research, we found ways to ask to go on a date, how to ask someone on a date, and a date planner. The date planner we found though required lots of personal information in the form of an email to be sent to the company. After some time, the customer would receive back results based on the personal questionnaire. How we are choosing to implement our date planner is new and has not been done so far.
   5. Our user profiles is only meant to save searches at the moment. No other personal information will be required by the user besides a username and a password.
   6. At the moment, this application is not meant for financial gain, but it has been considered for the future in the form of commercials endorsement or sponsorships.