

21 Questions - Requirements Specification

Gabriel Lopez de Leon -
Curtis Milo - milocj - 1305877
Max Moore - moorem8 - 1320009
Alex Rahman - rahmaa25 - 1305735
Connor Sheehan - sheehacg - 1330964

February 8th, 2016

Abstract

Requirements Specification for the 21 Questions phenomenon identifier project in SFWR 3A04 at McMaster University.

1 Introduction

This section of the SRS should provide an overview of the entire SRS.

1.1 Purpose

- a) Delineate the purpose of the SRS
- b) Specify the intended audience for the SRS

1.2 Scope

- a) Identify the software product(s) to be produced by name (e.g., Host DBMS, Report Generator, etc.)
- b) Explain what the software product(s) will, and, if necessary, will not do
- c) Describe the application of the software being specified, including relevant benefits, objectives, and goals
- d) Be consistent with similar statements in higher-level specifications (e.g., the system requirements specification), if they exist

1.3 Definitions, Acronyms, and Abbreviations

- **Area of Interest** - The phenomenon initially unknown to the system but known to the user. Throughout this document it will be referred to as the *AOE*.
- **API** - Short form for application programming interface.
- **Questions** - The method by which each expert receives information for analysis. These *questions* will be answered primarily with yes or no answers.

1.4 References

- a) Provide a complete list of all documents referenced elsewhere in the SRS
- b) Identify each document by title, report number (if applicable), date, and publishing organization
- c) Specify the sources from which the references can be obtained

1.5 Overview

- a) Describe what the rest of the SRS contains
- b) Explain how the SRS is organized

2 Overall Description

This section of the SRS should describe the general factors that affect the product and its requirements. It does not state specific requirements; it provides a background for those requirements and makes them easier to understand.

2.1 Product Perspective

- a) Put the product into perspective with other related products, i.e., context
- b) If the product is independent and totally self-contained, it should be stated here
- c) If the SRS defines a product that is a component of a larger system, as frequently occurs, then this subsection should relate the requirements of that larger system to functionality of the software and should identify interfaces between that system and the software
- d) A block diagram showing the major components of the larger system, interconnections, and external interfaces can be helpful

2.2 Product Functions

- a) Provide a summary of the major functions that the software will perform.
 - **Example:** An SRS for an accounting program may use this part to address customer account maintenance, customer statement, and invoice preparation without mentioning the vast amount of detail that each of those functions requires.
- b) Functions should be organized in a way that makes the list of functions understandable to the customer or to anyone else reading the document for the first time
- c) Textual or graphical methods can be used to show the different functions and their relationships
 - Such a diagram is not intended to show a design of a product, but simply shows the logical relationships among variables

2.3 User Characteristics

- a) Describe those general characteristics of the intended users of the product including educational level, experience, and technical expertise
- b) Do not state specific requirements, but rather provide the reasons why certain specific requirements are later specified

2.4 Constraints

The following constraints are to limit the design of the software to be:

- a) The system must have three experts modules in which will determine an aspect about the location to identify.
- b) The systems software experts are not able to communicate between each other.
- c) The system should use the Google maps API.
- d) The should must be designed so that we can swap out experts.
- e) The system must have Internet access.
- f) The system must use cryptography for sending messages from experts.
- g) The system must follow all legal rules of the Canadian laws.
- h) The system must follow all rules that the Google maps API.

2.5 Assumptions and Dependencies

a) Assumptions

- The system can only find locations within a set domain.
- The user will only use the system to identify real locations.

b) Dependencies

- The system is dependent on the functionality the Google Maps API.
- The system's performance is dependent on the speed of the user's Internet.
- The system's performance is dependent on the operating systems overhead.

2.6 Apportioning of Requirements

a) Letting the user add locations to list of locations

b) Using social media to allow the user to share the location that the system predicted.

c) Trying to guesses base on events from there face book event list and large news stories.

3 Functional Requirements

This section of the SRS should contain all of the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. Throughout this section, every stated requirement should be externally perceivable by users, operators, or other external systems. These requirements should include at a minimum a description of every input (stimulus) into the system, every output (response) from the system, and all functions performed by the system in response to an input or in support of an output.

You normally have two options for organizing your functional requirements:

1. Organize first by *business events*, then by *viewpoints*
2. Organize first by *viewpoints*, then by *business events*

Choose the one which makes the most sense.

For example, if you wish to organization by business events:

BE1. Business Event

VP1.1 Viewpoint

- i. Requirement
- ii. Requirement
- iii. ...

VP1.2 Viewpoint

- i. Requirement
- ii. Requirement
- iii. ...

VP1.3 ...

BE2. Business Event

VP2.1 Viewpoint

- i. Requirement
- ii. Requirement

iii. ...

VP2.2 Viewpoint

i. Requirement

ii. Requirement

iii. ...

VP2.3 ...

OR, if you wish to organization by viewpoints:

VP1. Viewpoint

BE1.1 Business Event

i. Requirement

ii. Requirement

iii. ...

BE1.2 Business Event

i. Requirement

ii. Requirement

iii. ...

BE1.3 ...

VP2. Viewpoint

BE2.1 Business Event

i. Requirement

ii. Requirement

iii. ...

BE2.2 Business Event

i. Requirement

ii. Requirement

iii. ...

BE2.3 ...

4 Non-Functional Requirements

4.1 Look and Feel Requirements

1. The system shall be usable on the first use by any user over the age of ten.
2. The system shall display any information in a visual way.
3. The system shall show a graphical view of locations and **AOEs**.
4. The system shall include a tutorial to teach new users correct usage of the application.

4.1.1 Appearance Requirements

LF1. The game will appear in a pleasant manner suitable for all demographics.

LF2. The game will appear innovative, and it should be apparent the application was developed in 2016.

4.1.2 Style Requirements

- LF1. The style of the game's GUI will not distract from the essence of the game.
- LF2. The game will be playable by colorblind persons.
- LF3. The game will be playable by deaf persons.

4.2 Usability and Humanity Requirements

4.2.1 Ease of Use Requirements

- UH1. The system shall provide questions that are easy to understand.
- UH2. The system shall be easy to navigate and understand the layout.

4.2.2 Personalization and Internationalization Requirements

- UH1. The system should be able to present information in a variety of languages.
- UH2. The system shall be able to set fonts and colors to the users preference.

4.2.3 Learning Requirements

- UH1. The system will provide a method of learning the game's rules and objectives.

4.2.4 Understandability and Politeness Requirements

- UH1. The questions asked in the game will be simple and focus on one specific point.
- UH2. The questions asked in the game will not include offensive, insensitive or immature remarks.

4.2.5 Accessibility Requirements

- UH1. The system will be usable by colour blind persons.
- UH2. The system will be usable by deaf persons.
- UH3. The system will be able to read questions aloud to the user.

4.3 Performance Requirements

4.3.1 Speed and Latency Requirements

1. Any operation that does not require use of the internet will respond in 2 seconds.
2. Any operation that does require use of the internet will respond within 30 seconds.

4.3.2 Safety-Critical Requirements

There are no safety critical requirements for this application.

4.3.3 Precision or Accuracy Requirements

1. The system shall predict the correct location with 75% accuracy.
2. The system shall predict the user's environment 90% of the time.
3. The system shall predict landmarks 80% of the time.

4.3.4 Reliability and Availability Requirements

PR1. The game will remain online to play

4.3.5 Robustness or Fault-Tolerance Requirements

PR1.

4.3.6 Capacity Requirements

PR1. The system should be able to respond to at least 20 client queries at a time.

4.3.7 Scalability or Extensibility Requirements

1. The system should be able to add new **AOEs** easily.
2. The system should be able to be implemented on new systems without changing

4.3.8 Longevity Requirements

PR1.

4.4 Operational and Environmental Requirements

1. The system shall check to ensure that there is sufficient internet connectivity.
2. The system shall ensure that any servers are up for 80% of the time.

4.4.1 Expected Physical Environment

The application's expected physical environment will be anywhere that a mobile phone can operate. Typically users will not interact with the system in an emergency or other circumstantial event.

4.4.2 Requirements for Interfacing with Adjacent Systems

OE1. The system should provide an interface to connect with adjacent systems

4.4.3 Productization Requirements

OE1.

4.4.4 Release Requirements

OE1.

4.5 Maintainability and Support Requirements

4.5.1 Maintenance Requirements

1. The system shall be easy to update.
2. The system shall

4.5.2 Supportability Requirements

1. The system shall be able to run on at least 90% of Android devices.

4.5.3 Adaptability Requirements

1. The system shall be

4.6 Security Requirements

1. Any information that enters or exits the system shall be encrypted.
2. The system will not store or transmit information related to user location.
3. The system will not store usernames or passwords.

4.6.1 Access Requirements

SR1. The system should be able to run anywhere that a data or wifi connection is available.

4.6.2 Integrity Requirements

SR1.

4.6.3 Privacy Requirements

SR1. The system shall not store any user's location data.

SR2. The system shall not store any user passwords.

4.6.4 Audit Requirements

SR1.

4.6.5 Immunity Requirements

SR1.

4.7 Cultural and Political Requirements

4.7.1 Cultural Requirements

1. The system shall ensure that culturally significant **AOEs** are presented in a respectful manner.

4.7.2 Political Requirements

1. The system shall show no bias towards any political party or related organization.
2. The system will not endorse or associate with any political group or government.

4.8 Legal Requirements

1. The system shall operate within the laws of Canada and the United States of America.

4.8.1 Compliance Requirements

LR1.

4.8.2 Standards Requirements

LR1.

A Division of Labour

Include a Division of Labour sheet which indicates the contributions of each team member. This sheet must be signed by all team members.