21 Questions - Requirements Specification

Gabriel Lopez de Leon - lopezdg - 1310514
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 ENG 3A04 at McMaster University.

Contents

1 Introduction

21 Questions game designed to ask a set of yes or no questions from that determine the location, area and establishment in which the user wanted to identify. The following document is a detailed description of the requirements and specifications that are essential in order to design 21 questions.

1.1 Purpose

The software system to be is a game in order for people to try too guess an area of interest in which the user was thinking of. This system is designed to be played as a game, however the functionality will have a wide range of uses. Some people who are would benefit from using this software system are:

- **Users**- Who wish to play and enjoy the game. This will often prove useful for when the user remembers a specific characteristic about the area but not enough to be able too search on the Internet for it.

- **Travelers**- People may also wish find new locations such as restaurants nearby when they are planning to find a destination to travel too. This system will help them find new locations in which they may want to travel too.

- **Establishments**- This game is also intended to help people traveling to new locations help find establishments to visit. Establishment's will be able to ask too add there business into the system to allow people too find them when playing the game.

- **Builders**- People such as city planners who wish to find new areas, in order to build cities.

1.2 Scope

The name given system to be is 21 questions. It is a game designed so that it will ask the user a set of yes or no questions in order to determine an Area of interest; Specifically it will try to determine the location that the user is thinking of, the surrounding environment in which it is located, and the establishment itself. This is designed to be a game, but is also intended to give users an idea of the different establishments that are around the area in which they may not have known about; this helps grow local economies and help give tourists a better understanding of the culture within this specific area.

1.3 Definitions, Acronyms, and Abbreviations

Throughout the document the following abbreviations and words will be used:

- **API** - Short form for application programming interface.

- **Area of Interest** - The phenomenon initially unknown to the system but known to the user.

- **Google Play Store** - Google's digital distribution platform, as well as the official 'app store' for the Android Operating System.

- **IT** - Short form for information technology.

- **Query** - A set of carefully selected words used in order to describe or pin point a location on a map.

- **Questions** - The method by which each expert receives information for analysis. These *questions* will be answered primarily with yes or no answers.

- **SRS** - Short form for Software Requirements Specification, refers to this document.

1.4 References

1 Dr. Khedri - 3A04, "3A04 Project Outline", 2016. [Online]. Available: http://www.cas.mcmaster.ca/ khedri/wp-content/uploads/COURSES/3A04/Project2016.pdf. [Accessed: 01- Feb- 2016].

1.5 Overview

The remainder of this document contains overall product overview, any characteristics about the product, assumptions on the real world. The document will also explain the functionalities of the software based on different users and the overall qualities of the software.

2 Overall Description

2.1 Product Perspective

21 Questions is simple location identification application which asks the user a total of 21 questions to try and identify the **AOI**. 21 Questions is very similar to other identification applications such as *Akinator* or the game *20Q*. *Akinator* is a web based program where the user thinks of a character and answers a series of questions, leading to the program guessing the character the user has thought of. *20Q* or 20 Questions is the original inspiration for our program, like *Akinator*, the user thinks of something but in this case it can be any person, place or thing. The game would then ask the user a total of 20 questions to try and guess what the user is thinking of. To differentiate our application with *20Q*, 21 Questions focuses on locations only with an end goal of outputting the result through *Google Maps*. Overall, the system is independent and only uses the Google API as its only resource outside of the system.

2.2 Product Functions

The following is a list of product functions that the software will perform:

a) The system identifies an AOI using three different experts.

- This is one of the main requirements and functionality of 21 Questions, the system uses three different experts which are independent of each other to reach a resulting output to user inputs.

b) The system forms a query using the answers inputted by the user and is then outputted through Google Maps.

- From the users' answers to the preset questions, **queries** are formed and used to find the **AOI**. A set of queries are stored in a database which is referenced and used along with the Google maps API to display the resulting location on a map.

c) The system allows for users to add new queries.

- The ability to add new queries for new locations is an important functionality of the application as it allows the database to continuously grow.

d) The system shall allow for voice commands as a possible input to answer the questions.

- The use of voice commands will be our innovative feature and will allow for a wider range of users to be able to use the app. Adding the use of voice commands as an input improves the applications accessibility and allows for those with visual impairment, such as blind persons to be able to use 21 Questions.

2.3 User Characteristics

a) General Characteristics of Intended Users

- 21 Questions is a location identifier application whose intended use is for any user above the age of ten. The application requires minimal training, experience or technical expertise to use, and can be easily picked up and used by anyone. 21 Questions can be used as a game where users select a location and see if the software can identify it based on the user's answers to a set of predefined questions. The application can also be used to identify a location, structure, or establishment as answering the questions will output the result via Google maps.

b) Reasoning for Requirements

- As later stated in the non-functional requirements, any user can easily learn how to use the application. With game like aspects, 21 questions is meant for users of all ages and thus it should be simple enough that users of a younger audience will be able to play while users of an older audience will still find the game appealing.
- The application targets a wide variety of users with different ethnic backgrounds and thus must be respectful of everyones cultures, religious belief and political preferences.

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 one could swap out any expert for another.

e) The system must have access to the Internet.

f) The system must encrypt all transmitted messages.

g) The system must abide to all the Canadian laws.

h) The system must abide to all terms and conditions set by the Google Maps API.

2.5   Assumptions and Dependencies

a) Assumptions

- The system can only find locations within the 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 and quality 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 the existing set of locations.

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

c) Suggest locations based off of interest and large news stories through social media.

3 Functional Requirements

The system will react to a set of key stimuli based on the interests of a set of people. The system will provide output based on the following requirements:

BE1 User enters new search.

VP1.1 Users

i. The system shall be able to display the resulting locations on a map.
ii. The system shall be able to ask the user questions.
iii. The system shall allow the user to answer yes or no to questions.
iv. The system shall determine the closest locations, environment and place.
v. The system shall allow each expert to determine the answer given the set of questions and their answers.

VP2.2 Establishments

i. Not Applicable

VP3.3 TA/Prof (Management role)

i. Not Applicable

VP4.4 IT Maintenance

i. Not Applicable

VP5.5 Front End Developers

i. Not Applicable

VP6.6 Back End Developers

i. Not Applicable

BE2 An unlisted establishment requests to be included in the application.

VP2.1 Users

i. Not Applicable

VP2.2 Establishments

i. The system shall be able to display the resulting locations on a map.

VP2.3 TA/Prof (Management role)

i. The system shall allow developers to easily add or remove questions or locations.

VP2.4 IT Maintenance

i. The system shall allow developers to easily add or remove questions or locations.

VP2.5 Front End Developers

i. The system shall be able to display the resulting locations on a map.

VP2.6 Back End Developers

i. The system shall allow developers to easily add or remove questions and locations.

BE3 Updates or maintenance of the app is required.

VP3.1 Users

i. The system shall inform the user if an update or maintenance is required.

VP3.2 Establishments

i. Not Applicable

VP3.3 TA/Prof (Management role)

i. The system shall allow the user to flag or report any inappropriate answers or questions.

VP3.4 IT Maintenance

    i. The system shall developers to easily add or remove questions and locations.

VP3.5 Front End Developers

    i. The system shall developers to easily add or remove questions and locations.

    ii. The system shall have 3 experts that will be able to be easily swapped out.

VP3.6 Back End Developers

    i. The system shall developers to easily add or remove questions and locations.

BE4 Adding and Removing Functionality

VP4.1 Users

    i. Not Applicable

VP4.2 Establishments

    i. Not Applicable

VP4.3 TA/Prof (Management role)

    i. Not Applicable

VP4.4 IT Maintenance

    i. The system shall developers to easily add or remove questions and locations.

    ii. The system shall not effect the integrity of the existing information.

VP4.5 Front End Developers

    i. The system shall developers to easily add or remove questions and locations.

    ii. The system shall not effect the integrity of the existing information.

VP4.6 Back End Developers

    i. The system shall developers to easily add or remove questions and locations.

    ii. The system shall not effect the integrity of the existing information.

BE5 User flags an incorrect or inappropriate search or result.

VP5.1 Users

    i. The system shall allow the user to flag or report any inappropriate answers or questions.

VP5.2 Establishments

    i. Not Applicable

VP5.3 TA/Prof (Management role)

    i. The system shall allow the user to flag or report any inappropriate answers or questions.

VP5.4 IT Maintenance

    i. The system shall developers to easily add or remove questions and locations.

VP5.5 Front End Developers

    i. The system shall developers to easily add or remove questions and locations.

VP5.6 Back End Developers

    i. The system shall developers to easily add or remove questions and locations.

4 Non-Functional Requirements

4.1 Look and Feel Requirements

LF1. The system shall be usable on the first use by any user over the age of ten.

LF2. The system shall display any information in a visual way.

LF3. The system shall show a graphical view of locations and **AOI**s.

LF4. The system shall include a tutorial to teach new users correct usage of the application.

4.1.1   Appearance Requirements

LF5.   The game will appear in a pleasant manner suitable for all demographics.

LF6.   The system shall use modern graphical user interface libraries and techniques.

4.1.2   Style Requirements

LF7.   The style of the game's GUI will not distract from the essence of the game.

LF8.   The game will be playable by people affected by colour vision deficiencies.

LF9.   The game will be playable by people affected by hearing deficiencies.

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

UH3.   The system shall be able to set fonts and colours to the users preference.

4.2.3   Learning Requirements

UH4.   The system will provide a set of instructions describing the game's rules and objectives.

4.2.4   Understandability and Politeness Requirements

UH5.   The questions asked in the game will be simple and focus on one specific point.

UH6.   The questions asked in the game will not include offensive, insensitive or immature remarks.

4.2.5   Accessibility Requirements

UH7.   The game will be playable for people affected by colour vision deficiencies.

UH8.   The game will be playable for people affected by hearing deficiencies.

UH9.   The system will be able to read questions aloud to the user.

4.3   Performance Requirements

4.3.1   Speed and Latency Requirements

PR1.   Any operation that does not require use of the internet will respond within 2 seconds.

PR2.   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

PR3.   The system shall predict the correct location with 75% accuracy.

PR4.   The system shall predict the user's environment 90% of the time.

PR5.   The system shall predict landmarks 80% of the time.

4.3.4   Reliability and Availability Requirements

PR6.   The game requires internet access to play.

4.3.5   Robustness or Fault-Tolerance Requirements

PR7.   The system shall include a wide variety of locations and monuments.

4.3.6   Capacity Requirements

PR8.   The system should be able to respond to at least 20 client queries at a time.

4.3.7   Scalability or Extensibility Requirements

PR9. The system should be able to add new **AOI**s easily.

4.3.8   Longevity Requirements

There are no longevity requirements for this application.

4.4   Operational and Environmental Requirements

OE1.   The system shall check to ensure that there is a sufficient internet connection.

OE2.   The system shall ensure that any servers are up for 80% of the time.

4.4.1   Expected Physical Environment

OE3.   The application should run in any physical environment in which a mobile phone can operate.

4.4.2   Requirements for Interfacing with Adjacent Systems

OE4.   The system should provide an interface to connect with adjacent systems.

4.4.3   Productization Requirements

OE5.   There are no productization requirements for this application.

4.4.4   Release Requirements

OE6.   The system shall be fully functional and have been tested for security flaws before release.

OE7.   The system shall be available to download in the *Google Play Store*.

OE8.   The system will be available for direct download.

4.5   Maintainability and Support Requirements

4.5.1   Maintenance Requirements

MS1.   The system shall be easy to update.

MS2.   The system shall be able to easily swap out modules.

4.5.2   Supportability Requirements

MS3.   The system shall be able to run on at least 90% of Android devices.

4.5.3   Adaptability Requirements

MS4.   The system can easily be implemented on new operating systems with little change.

4.6   Security Requirements

SR1.   Any information that enters or exits the system shall be encrypted.

SR2.   The system will not store or transmit information related to the user's location.

SR3.   The system will not store usernames or passwords.

4.6.1   Access Requirements

SR4.   The system should be able to operate anywhere that a data or wifi connection is available.

4.6.2   Integrity Requirements

SR5.   The system shall not allow any data to be modified by any algorithms.

4.6.3   Privacy Requirements

SR6.   The system shall not store any user's location data.

SR7.   The system shall not store any user passwords.

SR8.   The system shall not release information to outside parties about a specific business.

4.6.4   Audit Requirements

SR9.   Following a security audit at least 80% of the changes necessary shall be implemented.

4.6.5   Immunity Requirements

SR10.   The system shall not lend itself vulnerable to attacks or intruders.

SR11.   The system shall only allow users with proper clearances to modify or access data.

4.7   Cultural and Political Requirements

4.7.1   Cultural Requirements

CP1.   The system shall ensure that culturally significant **AOI**s are presented in a respectful manner.

4.7.2   Political Requirements

CP2.   The system shall show no bias towards any political party or related organization.

CP3.   The system will not endorse or associate with any political group or government.

4.8   Legal Requirements

4.8.1   Compliance Requirements

LR1.   The system shall operate within the jurisdiction of Canada and the United States of America.

LR2.   The system shall comply with any libraries offered by Google Maps.

4.8.2   Standards Requirements

LR3.   The system shall meet the standards set by Management (Professor and TAs).

LR4.   The system shall meet the standards set by ISO/IEC 12207.

A Division of Labour

The following includes the division of labour for Deliverable 1:

Team Member	Contributions
Gabriel Lopez de Leon	Described product perspective, product functions and user characteristics.
Maxwell Moore	Please enter your stuff
Curtis Milo	Generated and refined business events, view points, functional and non-functional requirements as well as constraints. Created list of assumptions, dependancies and apportioning of requirements.
Alexandra Rahman	Generated and refined business events, view points, functional and non-functional requirements as well as edited the final document. Assigned each business event and viewpoint a functional requirement if applicable.
Connor Sheehan	Generated and edited non-functional requirements. Added some definitions. Edited styling.

Table 1: Division of Labour

_____   _____

Gabriel Lopez de Leon   Date

_____   _____

Curtis Milo   Date

_____   _____

Max Moore   Date

_____   _____

Alex Rahman   Date

_____   _____

Connor Sheehan   Date