

High-Level Architectural Design

21 Questions

Gabriel Lopez de Leon - lopezdg - 1310514

Curtis Milo - milocj - 1305877

Maxwell Moore - moorem8 - 1320009

Alexandra Rahman - rahmaa25 - 1305735

Connor Sheehan - sheehacg - 1330964

March 7th, 2016

Contents

1	Introduction	2
1.1	Purpose	2
1.2	System Description	2
1.3	Overview	2
2	Use Case Diagram	3
3	Analysis Class Diagram	8
4	Architectural Design	9
4.1	System Architecture	9
4.2	Subsystems	9
5	Class Responsibility Collaboration (CRC) Cards	10
A	Division of Labour	15

List of Figures

1	Use Case for BE1	3
2	Use Case for BE2	4
3	Use Case for BE3	5
4	Use Case for BE4	6
5	Use Case for BE5	7
6	Analysis Class Diagram	8

List of Tables

1	CRC for Expert Controller	10
2	CRC for Question Controller	10
3	CRC for Graphics User Interface Controller	11
4	CRC for Start Screen	11
5	CRC for Settings Screen	11
6	CRC for Question Screen	11
7	CRC for Map Screen	12
8	CRC for Location Data	12
9	CRC for Environment Data	12
10	CRC for Establishment Data	13
11	CRC for Location Database	13
12	CRC for Environment Database	13
13	CRC for Establishment Database	14
14	Division of Labour	15

1 Introduction

This document is called the High-Level Architectural Design document and it will be giving the stakeholders a broad overview of how the project 21 Questions is designed and will be organized.

1.1 Purpose

The High-Level Architectural Design document outlines the functionality of the system through Use Case diagrams, and details the key classes of the system and how they relate via the use of an Analysis Class Diagram. The main purpose of this document is to explain in detail the software system to be developed, which in this case is the application, 21 Questions. Through the use of the various diagrams mentioned before, the main components of the product and their relationships with each other are shown. The main target audience for this document is the software developers as they need to see, in greater detail, how the system and its modules interact and how they are designed to work in relation to each other.

1.2 System Description

21 Questions is an android application that can be used as a location identifier 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 is a simple games that asks the user a series of twenty-one polar or binary questions to try to identify their area of interest. In this game the area of interest is limited to an establishment, building, place, or effigy with a focus on locations only with an end goal of displaying the result through Google Maps.

1.3 Overview

This document will outline the design of the *21 Questions* application from an architectural perspective. The document will begin from a use case outlook, outlining application functionality from a practical point of view and taking different actors and stakeholders into consideration. Next, an analysis class diagram and associated interpretation details is outlined, to specify application behaviours and resources in a modularized form. Following this section is a detailed architectural design as well as a set of class responsibility collaboration cards. These sections specify modules in greater detail, including interfaces to be implemented in the future. The order of these sections reflects a systematic progression from requirements to a more easily constructed application.

2 Use Case Diagram

a) User wants to enter new search (Figure 1).

BE1.1 The user presses the start button on the start screen.

BE1.2 The system will respond by bringing them to the question screen and asking them a question.

BE1.3 The user will answer yes, no or undecided to the question.

BE1.4 The system will ask the user another question.

BE1.5 After 21 questions, the system will respond by displaying a map to the user.

BE1.6 The user will hit done.

BE1.7 The system will ask if this was the location the user had in mind.

BE1.8 The user will answer yes it was or no it was not.

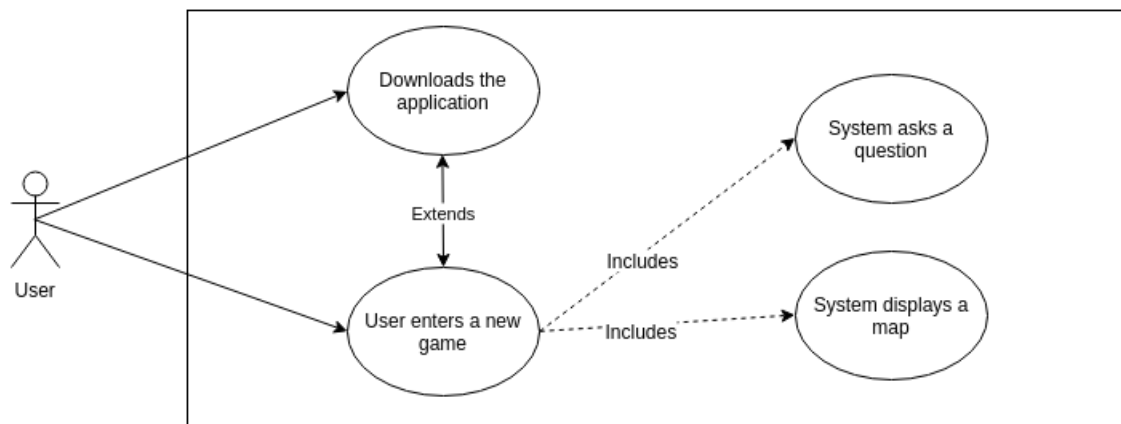


Figure 1: Use Case for BE1

b) An unlisted establishment requests to be included in the application (Figure 2).

BE2.1 A new business opens or a business opens a new location.

BE2.2 The business or establishment contacts the company to inform them they wish to be added to the system.

BE2.3 The IT specialists will send them a form for the business to fill out.

BE2.4 The business will return the form to the company.

BE2.5 The IT will verify that the information is valid and add it to the system.

BE2.6 The business is added.

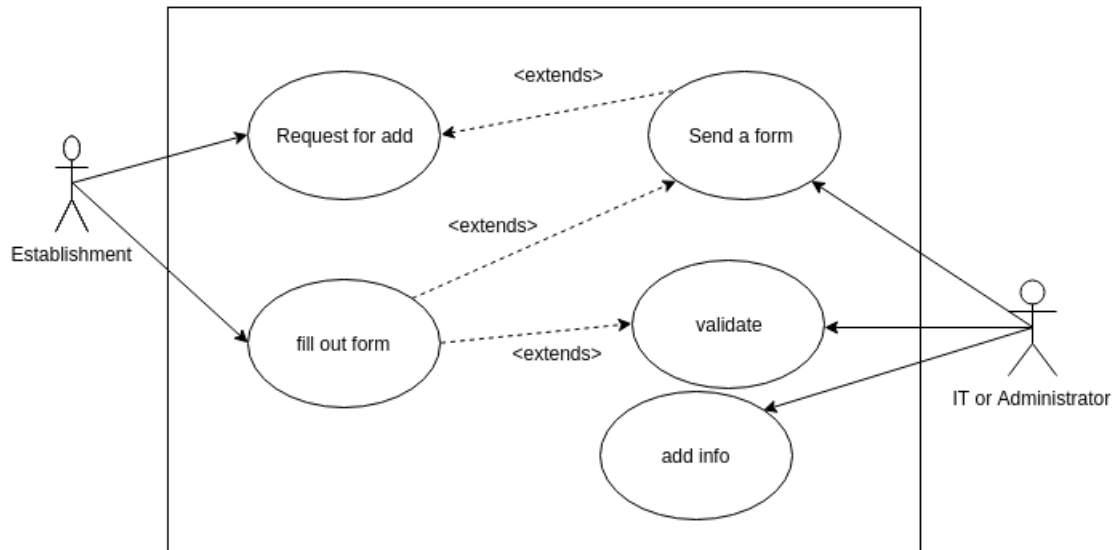


Figure 2: Use Case for BE2

c) Updates or maintenance of the application is required (Figure 3).

BE3.1 Internal management states an issue and requests an update/ maintenance.

BE3.2 Update/ maintenance is given a priority.

BE3.3 The IT specialist notifies the users that the system will update and be shut down for a certain period of time, if necessary.

BE3.4 The system will disconnect.

BE3.5 The necessary changes shall be made by the IT specialists.

BE3.6 The user will be notified if they need to update the application version.

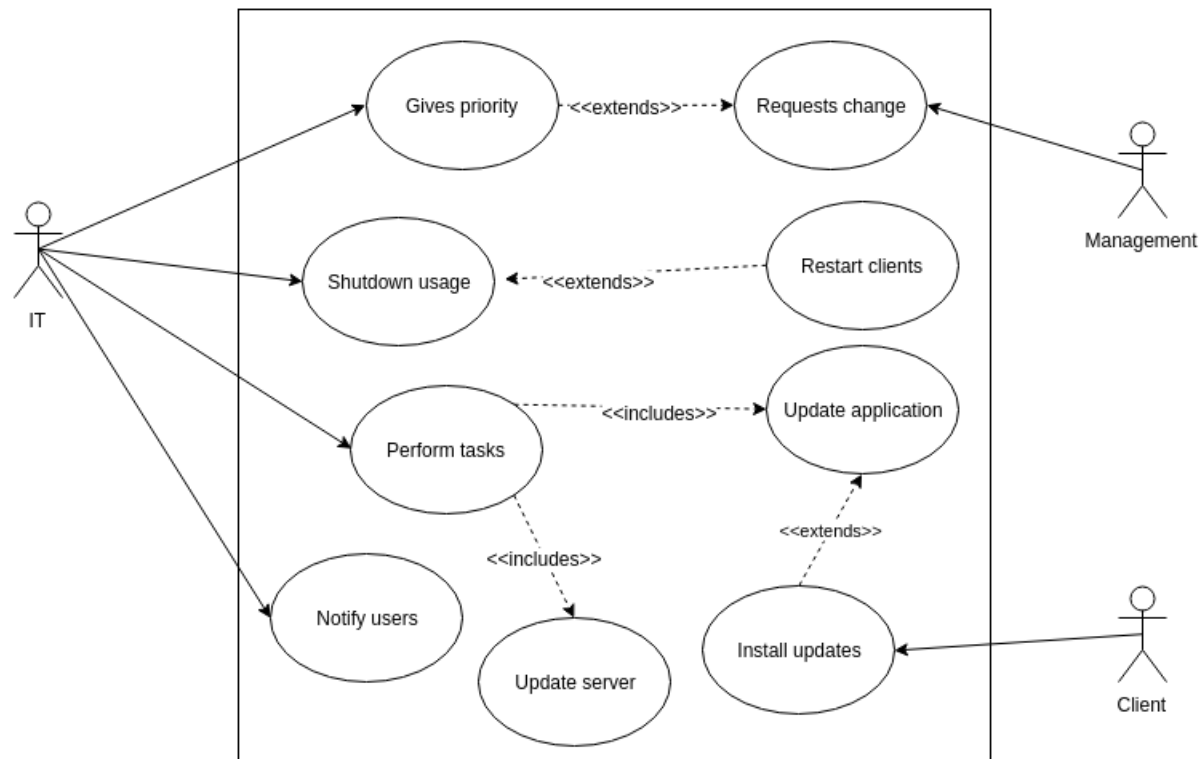


Figure 3: Use Case for BE3

d) Management requests implementation or change of experts (Figure 4).

BE4.1 Internal management states a function needs to be changed(added or removed).

BE4.2 The change is given a priority.

BE4.3 Function will be added or removed.

BE4.4 A small focus group is selected.

BE4.5 A survey is created.

BE4.6 Update is released to focus group.

BE4.7 Survey is sent to focus group.

BE4.8 The update will be released depending on the results of the survey.

BE4.9 If the survey results are not favourable, the function will be under review and released again to the focus group (repeat steps 5-9). Otherwise the update is released to the general public and the user is notified that they need to update app version.

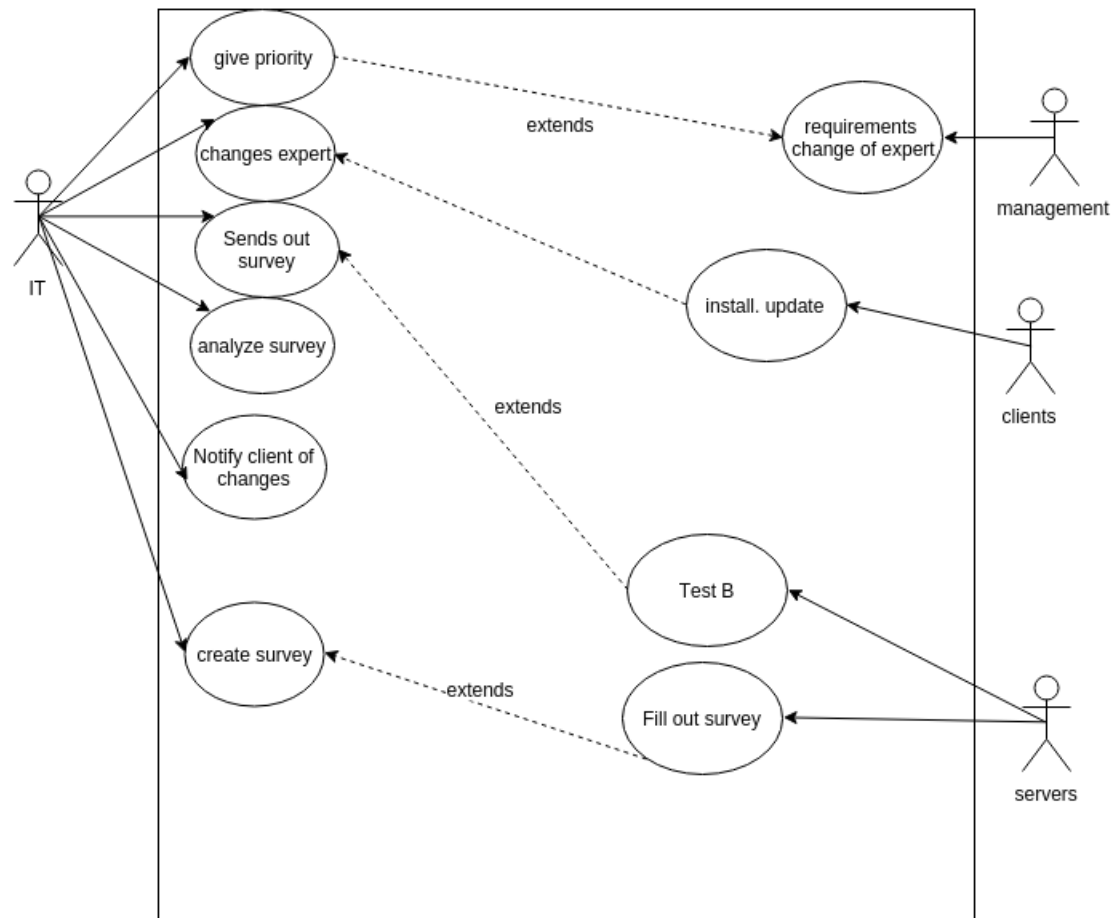


Figure 4: Use Case for BE4

e) User flags an incorrect or inappropriate search or result (Figure 5).

BE5.1 A business, user or internal management recognizes that the content is inappropriate.

BE5.2 The content shall automatically be hidden from other users.

BE5.3 The content is given a priority.

BE5.4 The content is put into a priority queue.

BE5.5 An IT will review the content and make appropriate changes (remove if necessary).

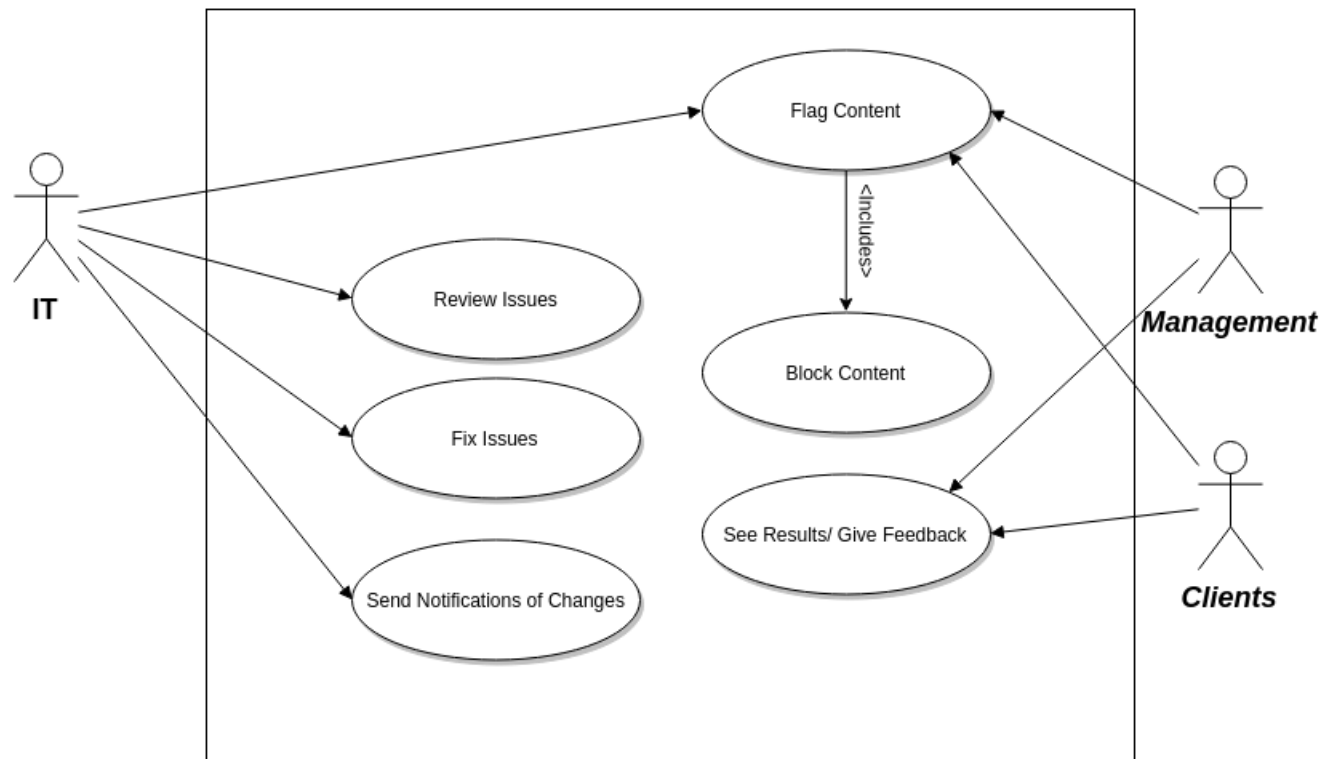


Figure 5: Use Case for BE5

3 Analysis Class Diagram

This section should provide an analysis class diagram for your application.

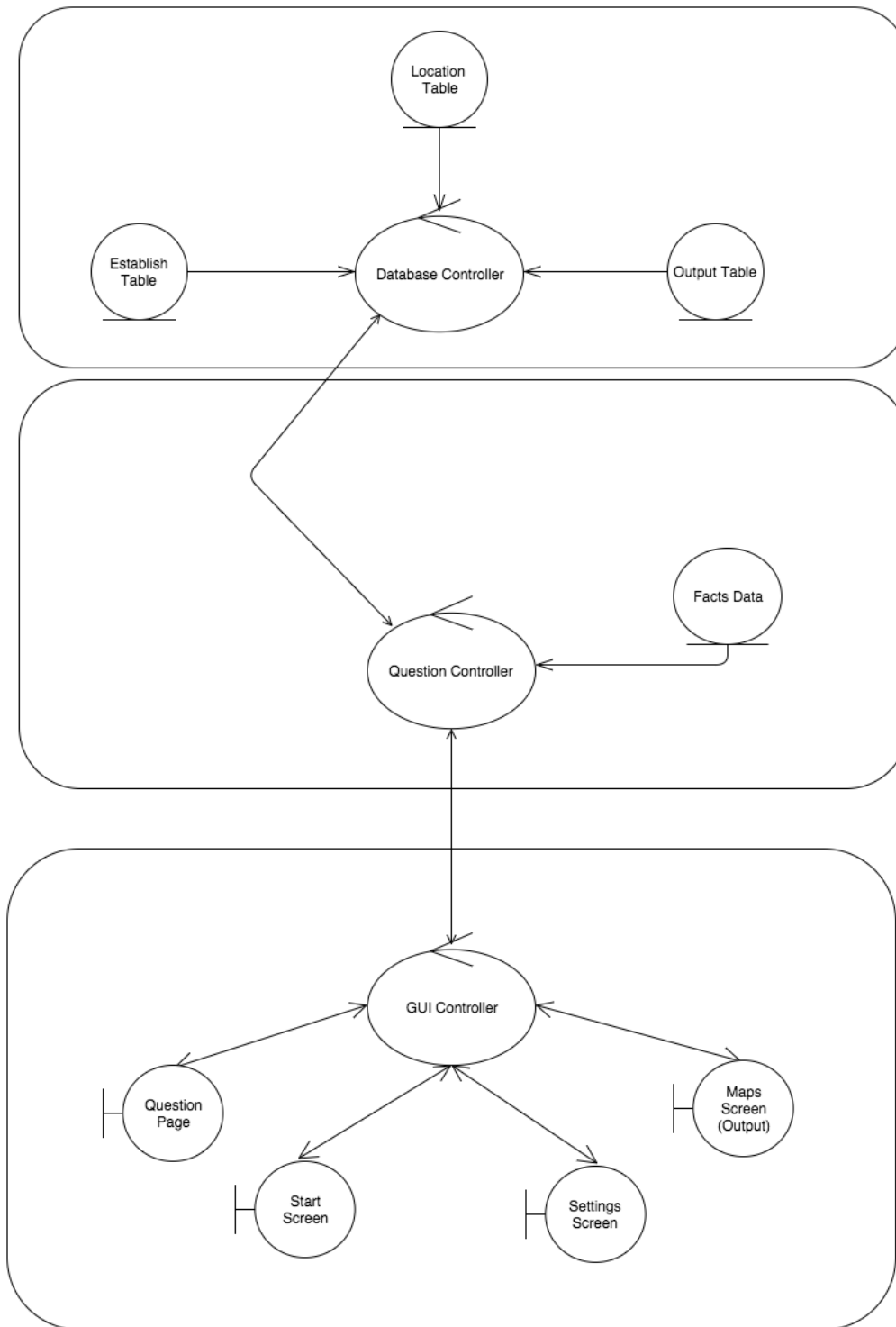


Figure 6: Analysis Class Diagram

4 Architectural Design

We used the blackboard architecture because of the way that the experts we have designed are supposed to be interact. The three will be giving very similar information so it makes sense to have them all compile it in one module and have that module given the information make an informed decision as to the query that we can. This also lends itself well to the principle of being opened to addition but closed to modification, because it will be easy to add more experts but difficult to change the nature of the experts them self. Meaning to change the nature or precision of the query we could simply add more specialized expert classes.

4.1 System Architecture

Figure 6 contains the analysis class diagram. The diagram is designed to have minimal communications between the the 3 separated units. The system design is a black board model, with the experts all being of the same type, and communicating to our "blackboard" the question controller class. The system will have two controllers, one for handling the pages of the system and the other for controlling which expert can ask questions. The controller will give an expert a start signal, the expert will ask his questions, and then write his best guess as to the answer on the "blackboard" or more formally will pass his guess as a parameter back to the question controller. The Black board style of architecture gives a very useful way of ensuring that Any expert can be removed, or more experts can be added given very little effort by the design team as far as overhead is concerned.

4.2 Subsystems

The database controller will be our lowest level of design, the user will never see this module. It will be used to access the location, establishment, and output files, which will all essentially be their own databases containing questions that pertain to the subject, and their own method for attaining a guess.

Next their is the question controlling module, which will be used for communication with the database and the user, information will be passed from the GUI to be used by the question controller. The only database contained within the question module is called the Facts Database, and contains information about how many question an expert has asked and if applicable what solution he proposes and how certain they are of the solution.

Finally we have our GUI controller, which takes information from the user and contains 4 boundary classes which will be passing information to it about clicks, choices and the like. These include the start screen, which is the initial menu a user will be viewing, which can begin a set of questions or change to the settings screen. The question screen is what is being shown during the process of running the app, which will display questions and receive user feedback, finally it will provide an option to terminate early and return to the start screen. If the setting screen is accessed it will provide a sub-menu which would allow a user to change things like accessibility, font colors and sizes, preferences and the like. finally the map screen GUI will be where we provide them with google maps, given our specific query as to where they are searching .

5 Class Responsibility Collaboration (CRC) Cards

This section should contain all of your CRC cards.

Class Name: Expert Controller	
Responsibility:	Collaborators:
<ul style="list-style-type: none"> • Deals with messages passed from the Question Controller • Passes desired questions to the Question Controller • Creates and deletes tables that relate to experts • Accesses the information by performing a query on the tables • Adds information to tables • Modifies information in the tables 	<ul style="list-style-type: none"> • Establishment Table • Environment Table • Location Table

Table 1: CRC for Expert Controller

Class Name: Question Controller	
Responsibility:	Collaborators:
<ul style="list-style-type: none"> • Requests questions from the Expert Controller based on the expert's needs • Accesses questions that have already been asked as well as the answers to said questions • Keeps track of the overall state of the game. This includes the number of questions total that have been asked • Provides a solution based on the experts' best guesses • Process answers from the GUI Controller • Provides the next question for the GUI Controller to display 	<ul style="list-style-type: none"> • Expert Controller • Facts Data • GUI Controller

Table 2: CRC for Question Controller

Class Name: Graphics User Interface Controller	
Responsibility:	Collaborators:
<ul style="list-style-type: none"> Controls which boundary class the user can interact with Processes and verifies information to send to the Question Controller Receives information from the Question Controller to display information or direct to answer Tracks the current and past state 	<ul style="list-style-type: none"> Question Controller Map Screen Start Screen Question Screen Setting Screen

Table 3: CRC for Graphics User Interface Controller

Class Name: Start Screen	
Responsibility:	Collaborators:
<ul style="list-style-type: none"> Provide an interface that the user can interact with Send information to the GUI Controller 	<ul style="list-style-type: none"> GUI Controller

Table 4: CRC for Start Screen

Class Name: Settings Screen	
Responsibility:	Collaborators:
<ul style="list-style-type: none"> Display a variety of settings to the user Allow the user to verify the changes to the current settings Allow the user to go back to previous screens 	<ul style="list-style-type: none"> GUI Controller

Table 5: CRC for Settings Screen

Class Name: Question Screen	
Responsibility:	Collaborators:
<ul style="list-style-type: none"> Receives information from the GUI Controller Takes user input from the users response for the question Sends the answer of the question to the GUI Controller Allows user to quit the current game 	<ul style="list-style-type: none"> GUI Controller

Table 6: CRC for Question Screen

Class Name: Map Screen	
Responsibility:	Collaborators:
<ul style="list-style-type: none"> • Informs the user of the guesses on a map • Informs the user of the guesses as an address • Receives and passes information from the GUI Controller • Requests the user feedback on the correctness of the guess 	<ul style="list-style-type: none"> • GUI Controller

Table 7: CRC for Map Screen

Class Name: Location Data	
Responsibility:	Collaborators:
<ul style="list-style-type: none"> • Holds the question that have been asked and their associated answers • Holds the number of questions the expert himself has asked • Holds most viable current guesses • Provides a probability in which the guess is correct 	<ul style="list-style-type: none"> • Question Controller

Table 8: CRC for Location Data

Class Name: Environment Data	
Responsibility:	Collaborators:
<ul style="list-style-type: none"> • Holds the question that have been asked and their associated answers • Holds the number of questions the expert himself has asked • Holds most viable current guesses • Provides a probability in which the guess is correct 	<ul style="list-style-type: none"> • Question Controller

Table 9: CRC for Environment Data

Class Name: Establishment Data	
Responsibility:	Collaborators:
<ul style="list-style-type: none"> • Holds the question that have been asked and their associated answers • Holds the number of questions the expert himself has asked • Holds most viable current guesses • Provides a probability in which the guess is correct 	<ul style="list-style-type: none"> • Question Controller

Table 10: CRC for Establishment Data

Class Name: Location Database	
Responsibility:	Collaborators:
<ul style="list-style-type: none"> • Allows for the addition of an location • Allows for the removal of an location • Holds a set of questions that correspond to certain location • Search for location based on question results 	<ul style="list-style-type: none"> • Expert Controller

Table 11: CRC for Location Database

Class Name: Environment Database	
Responsibility:	Collaborators:
<ul style="list-style-type: none"> • Allows for the addition of an environment • Allows for the removal of an environment • Holds a set of questions that correspond to certain environment • Search for environment based on question results 	<ul style="list-style-type: none"> • Expert Controller

Table 12: CRC for Environment Database

Class Name: Establishment Database	
Responsibility:	Collaborators:
<ul style="list-style-type: none">• Allows for the addition of an establishment• Allows for the removal of an establishment• Holds a set of questions that correspond to certain establishments• Search for establishment based on question results	<ul style="list-style-type: none">• Expert Controller

Table 13: CRC for Establishment Database

A Division of Labour

Team Member	Contributions
Gabriel Lopez de Leon	
Maxwell Moore	Wrote introduction(1.0), helped design BE's entered 1,3 into draw.io for submission.
Curtis Milo	
Alexandra Rahman	Wrote the system description and helped come up with the scenarios and use cases. Created the use case diagram for BE4. Collaborated on the CRC cards and the analysis class diagram. Refined the CRC cards and the analysis class diagram. Helped input the class responsibility collaboration cards into the document. Finally, helped edit and format the document.
Connor Sheehan	Created use case diagram for BE3. Added overview section. Added styling.

Table 14: Division of Labour

 Gabriel Lopez de Leon

 Date

 Curtis Milo

 Date

 Maxwell Moore

 Date

 Alexandra Rahman

 Date

 Connor Sheehan

 Date