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

List of Figures

List of Tables

1 Introduction

This document is called the High-Level Architectural Design document and it will be giving the stakeholders a large overview of what the organization of the project "21 questions" will look like. This large overview will include such things as how the primary business events will take place, the design insofar as how the classes will be communicating with each other, though few details about the classes themselves, and diagrams including which classes will be controllers, slaves, or simply data storage classes.

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 10. 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.

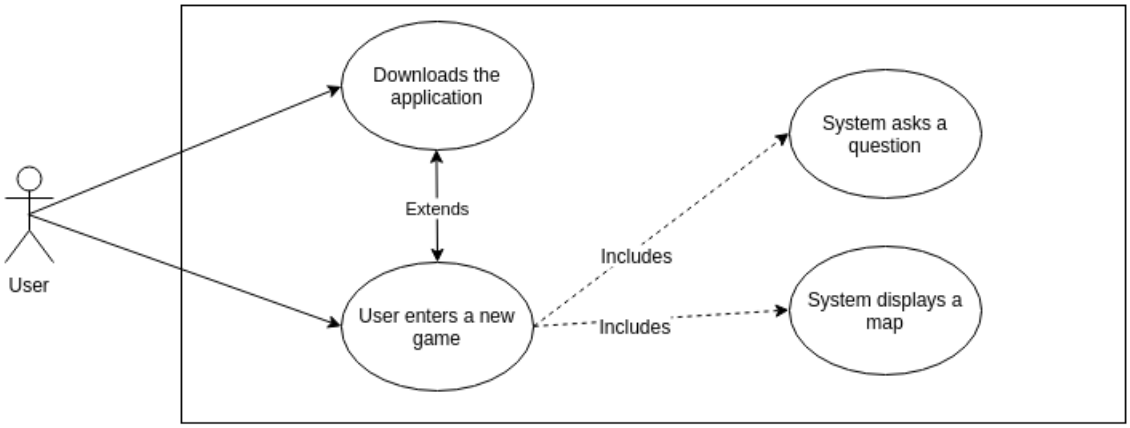

Figure 1: Use Case for BE1

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.

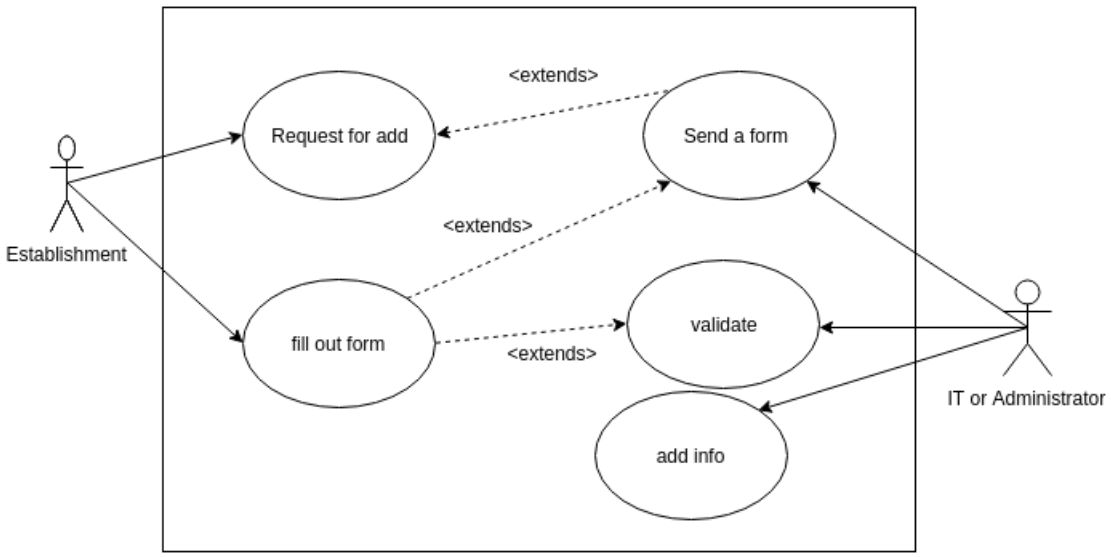

Figure 2: Use Case for BE2

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.

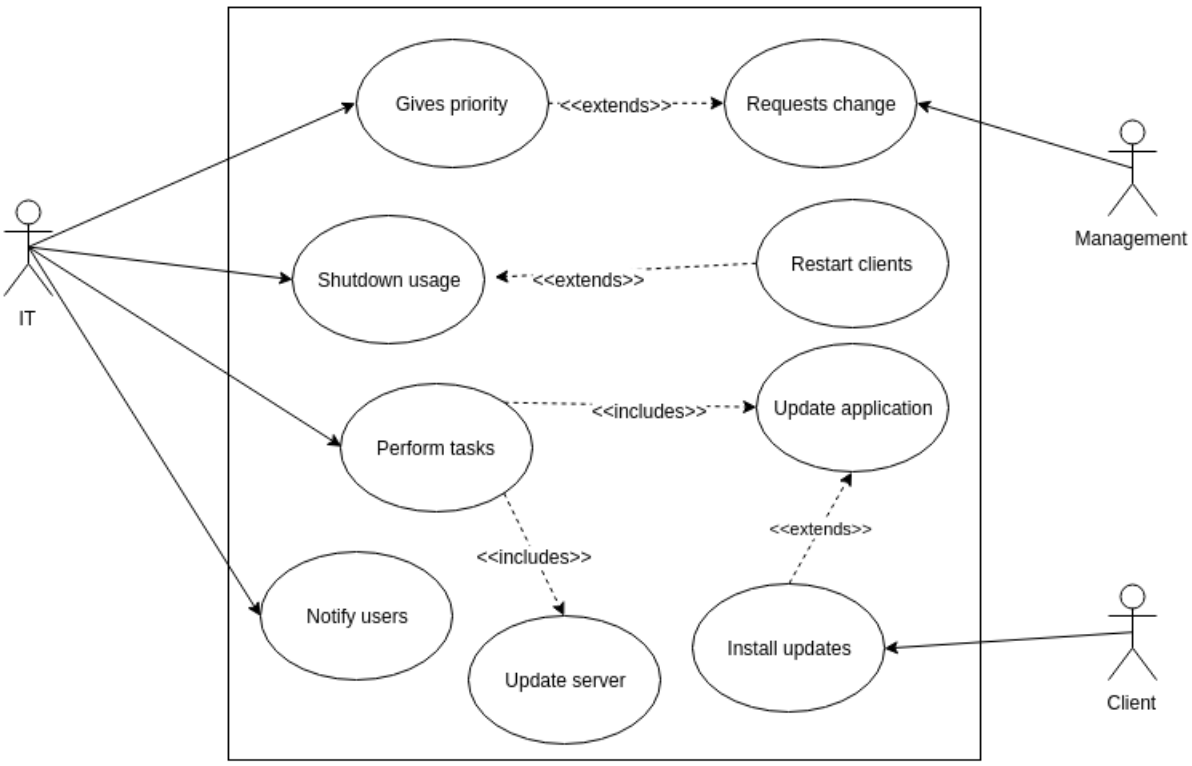

Figure 3: Use Case for BE3

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.

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.

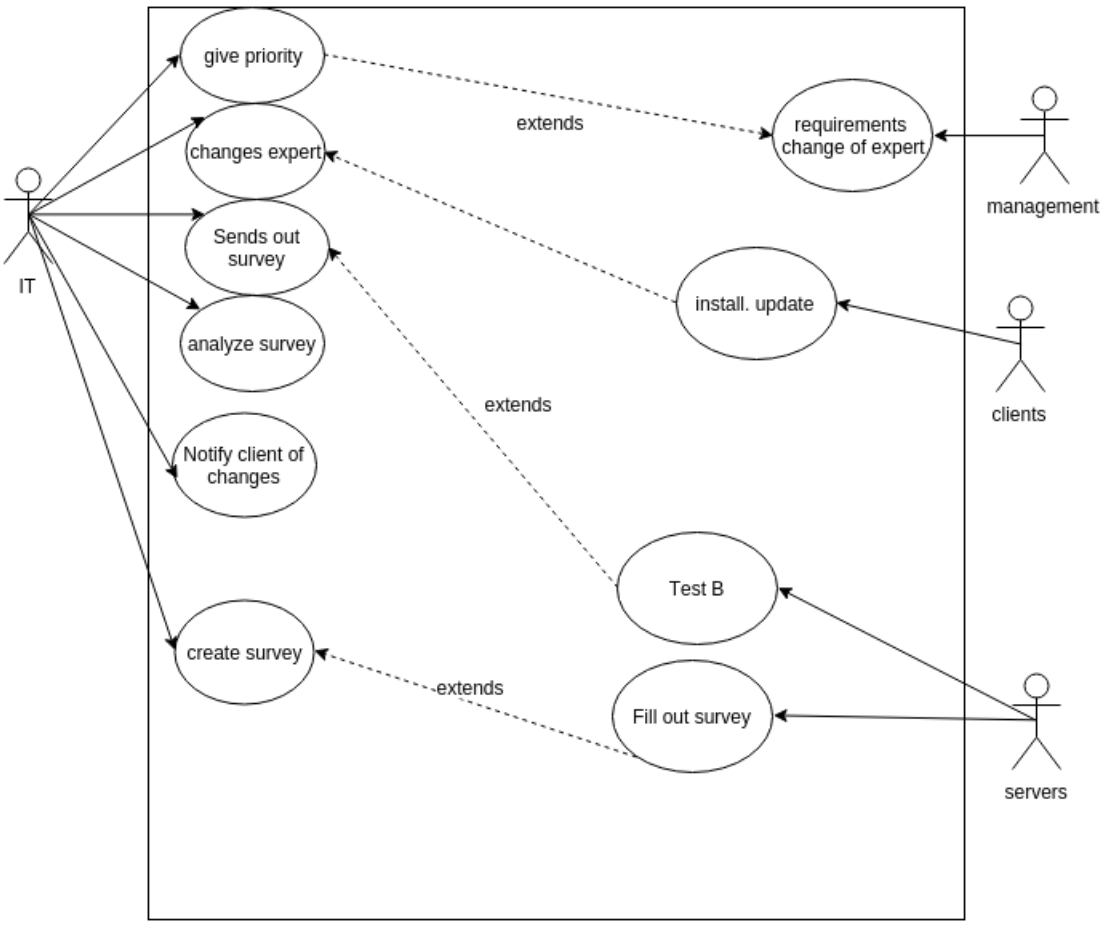

Figure 4: Use Case for BE4

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

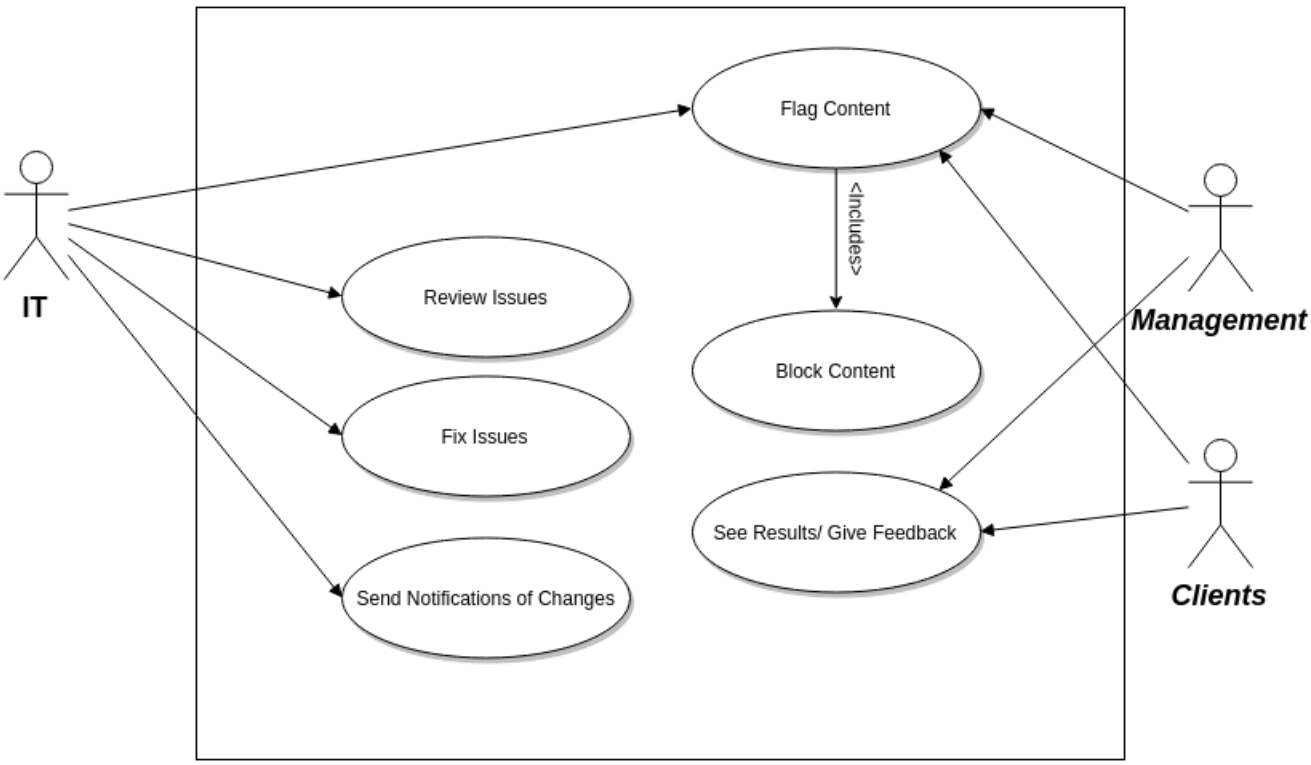

Figure 5: Use Case for BE5

3 Analysis Class Diagram

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

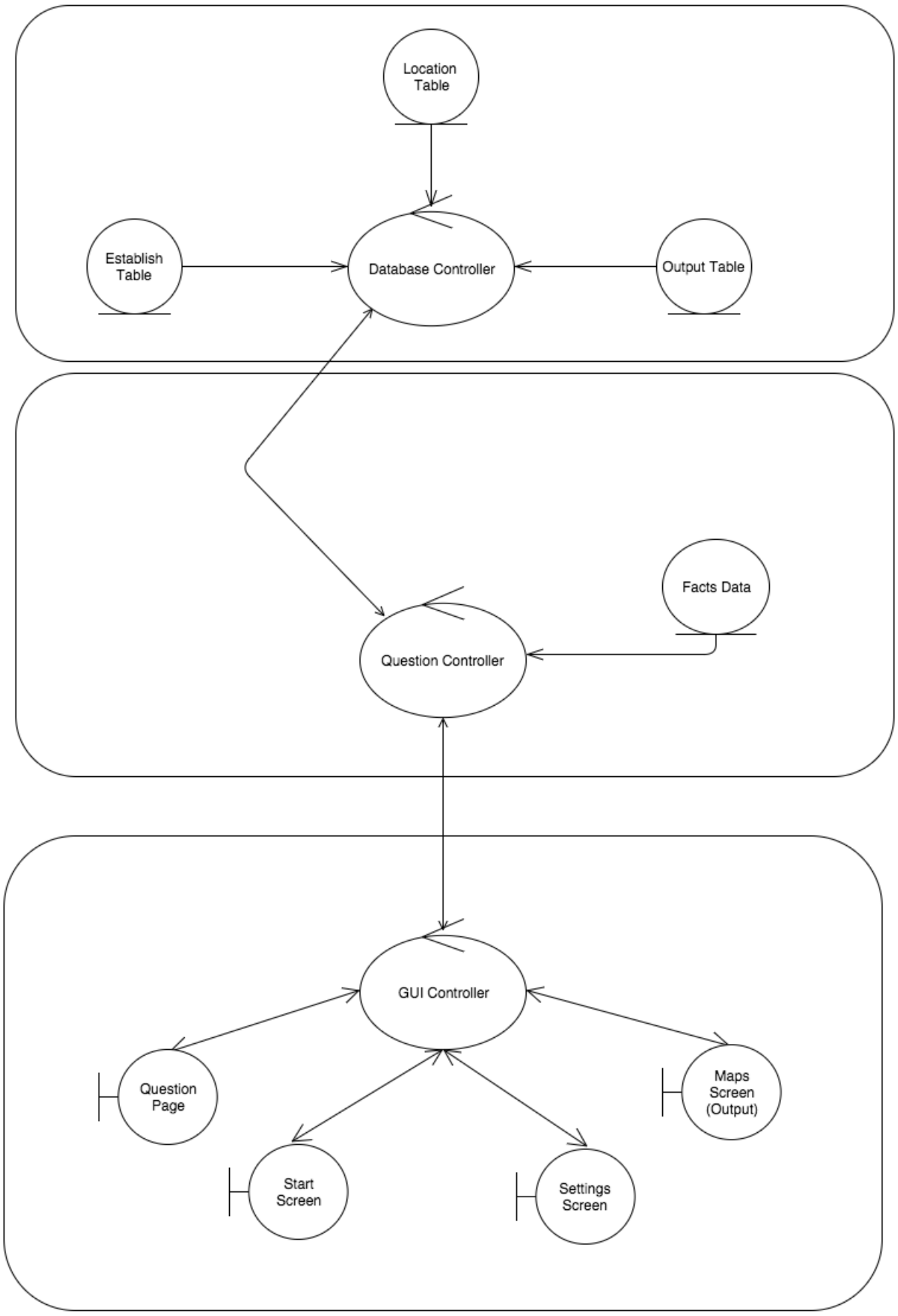

Figure 6: Analysis Class Diagram

4 Architectural Design

This section should provide an overview of the overall architectural design of your application. You overall architecture should show the division of the system into subsystems with high cohesion and low coupling.
We used a Blackboard architecture because.....

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 client server model, with the client side utilizing the model-view-controller architectural pattern. 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 information of what questions to ask is communicated from the server and stored in the clients side in an ADT. The client server style of architecture gives a very useful way of ensuring that all users will have the same information available to them without having a large amount of mobile updates. The model view controller is useful for being able to organize the structure of the clients side.

4.2 Subsystems

a) Provide a brief description of each subsystem. Be sure to document its purpose and relationship to other subsystems.

5 Class Responsibility Collaboration (CRC) Cards

This section should contain all of your CRC cards.

Class Name: Expert Controller	
Responsibility:	**Collaborators:**
• 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	• Establishment Table • Environment Table • Location Table

Table 1: CRC for Expert Controller

Class Name: Question Controller	
Responsibility:	**Collaborators:**

• 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	• Expert Controller • Facts Data • GUI Controller

Table 2: CRC for Question Controller

Class Name: Graphics User Interface Controller	
Responsibility:	**Collaborators:**
• 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	• Question Controller • Map Screen • Start Screen • Question Screen • Setting Screen

Table 3: CRC for Graphics User Interface Controller

Class Name: Question Page	
Responsibility:	**Collaborators:**
• 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	• GUI Controller

Table 4: CRC for Question Page Boundary Class

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 creates the use case diagram for BE4. Collaborated on the CRC cards and the analysis class diagram.
Connor Sheehan	Created use case diagram for BE3. Added overview section. Added styling.

Table 5: Division of Labour

Gabriel Lopez de Leon Date

Curtis Milo Date

Max Moore Date

Alex Rahman Date

Connor Sheehan Date