High-Level Architectural Design
21 Questions

Gabriel Lopez de Leon - lopezdg - 1310514
Curtis Milo - milocj - 1305877
Max Moore - moorem8 - 1320009
Alex Rahman - rahmaa25 - 1305735
Connor Sheehan - sheehacg - 1330964

March 7th, 2016

Contents

1 Introduction

This document is called th 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).

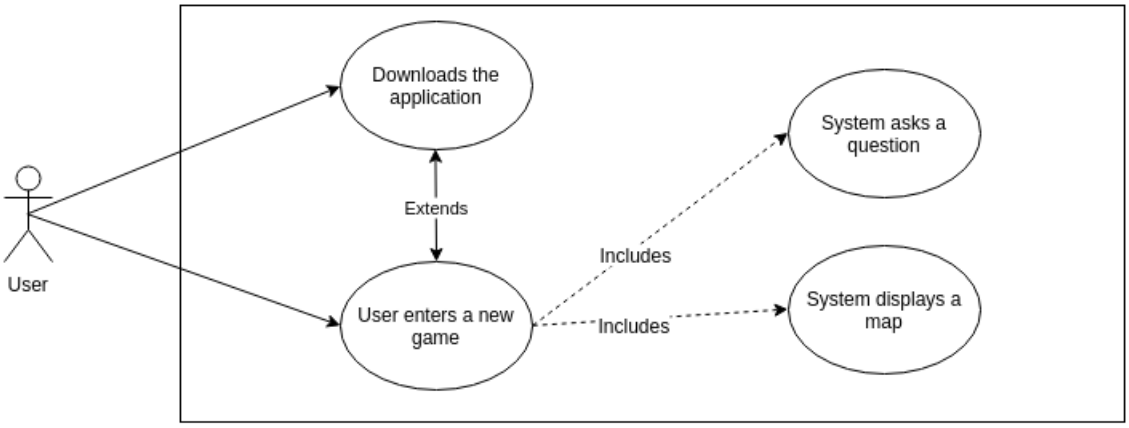

Figure 1: BE1

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

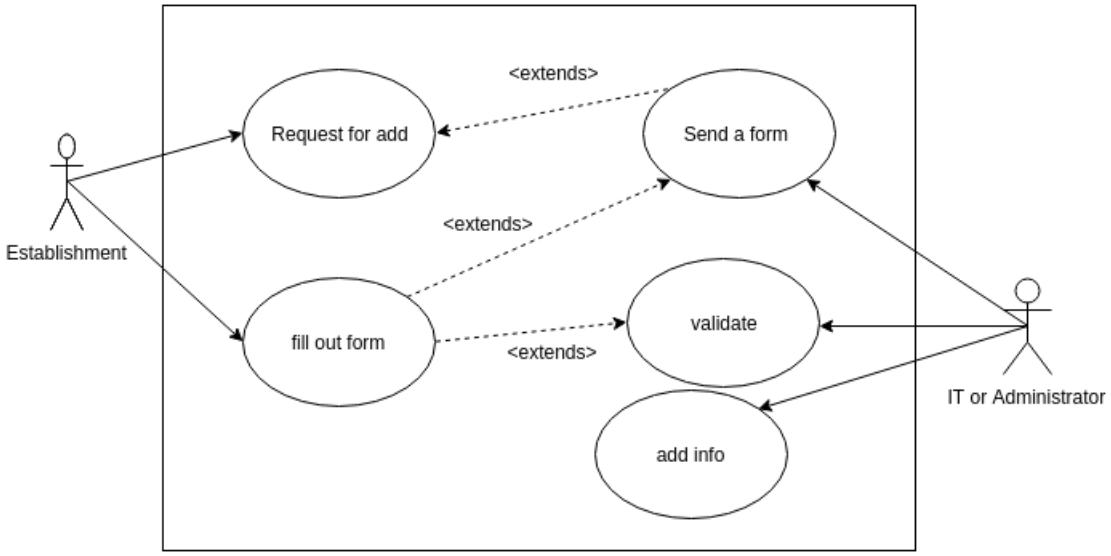

Figure 2: BE2

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

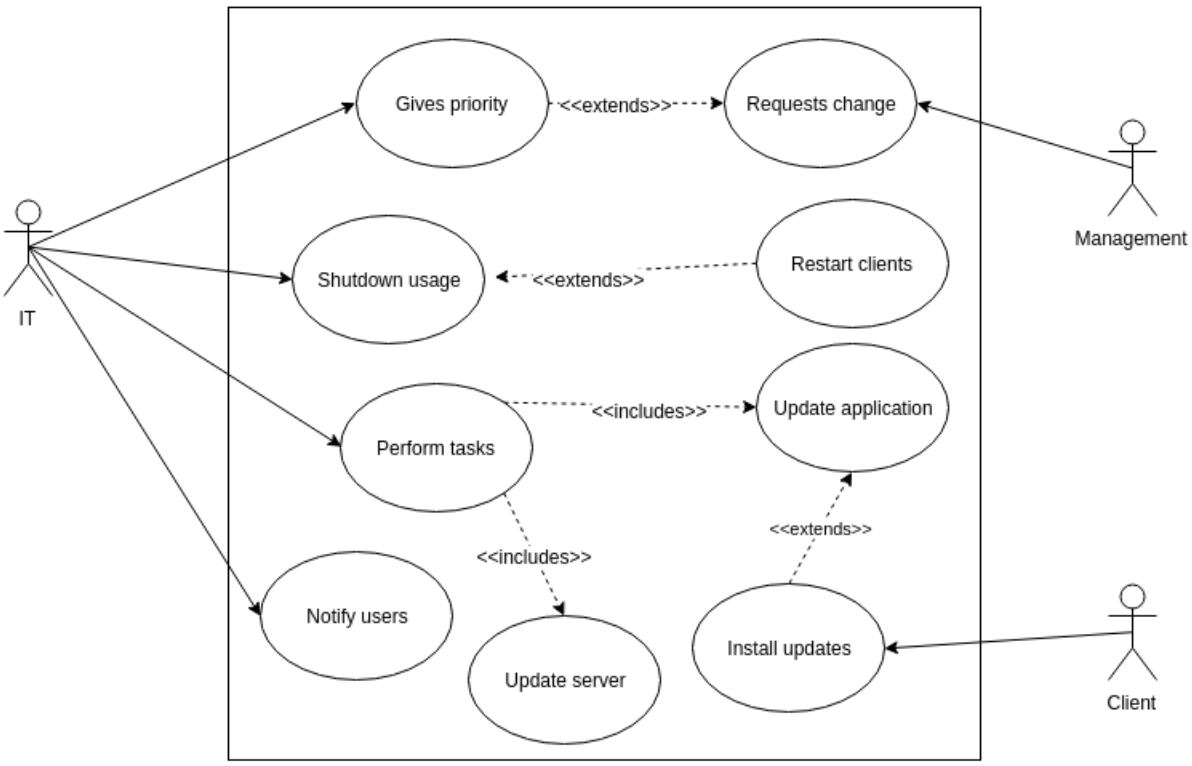

Figure 3: BE3

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

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

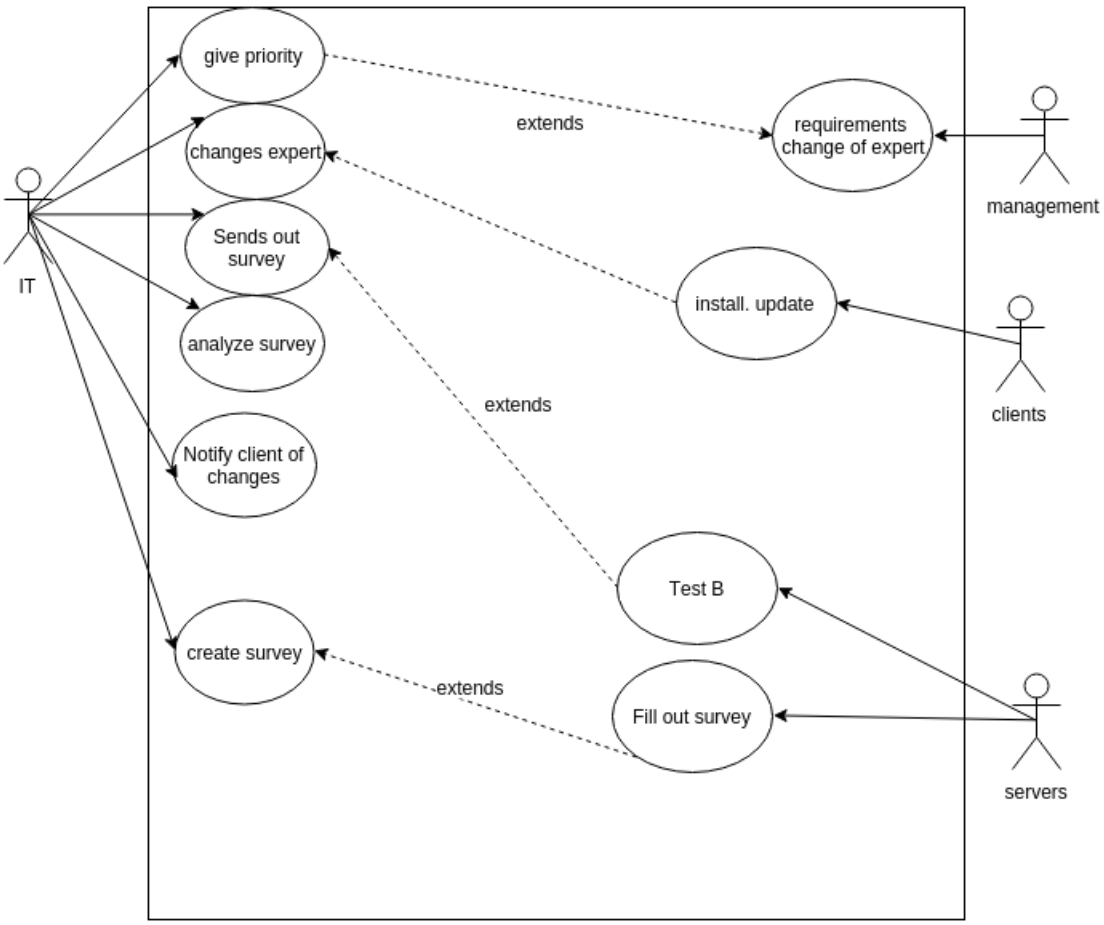

Figure 4: BE4

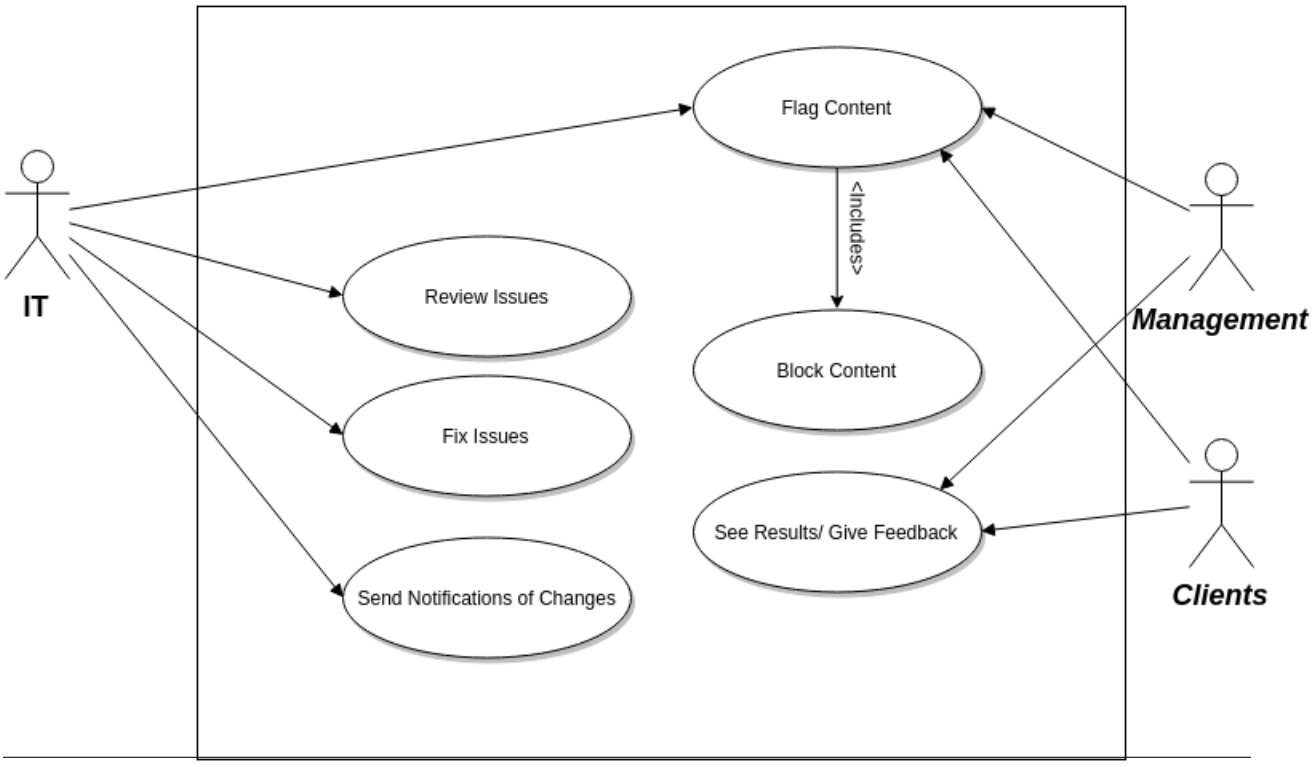

Figure 5: BE5

3 Analysis Class Diagram

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

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.

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.

a) Provide a CRC Card for each identified class

b) Please use the format outlined in tutorial, i.e.,

Class Name:	
Responsibility:	**Collaborators:**

A Division of Labour

Team Member	**Contributions**
Gabriel Lopez de Leon	
Maxwell Moore	
Curtis Milo	
Alexandra Rahman	
Connor Sheehan	Created use case diagram for BE3. Added overview section. Added styling.

Table 1: Division of Labour

Gabriel Lopez de Leon Date

Curtis Milo Date

Max Moore Date

Alex Rahman Date

Connor Sheehan Date

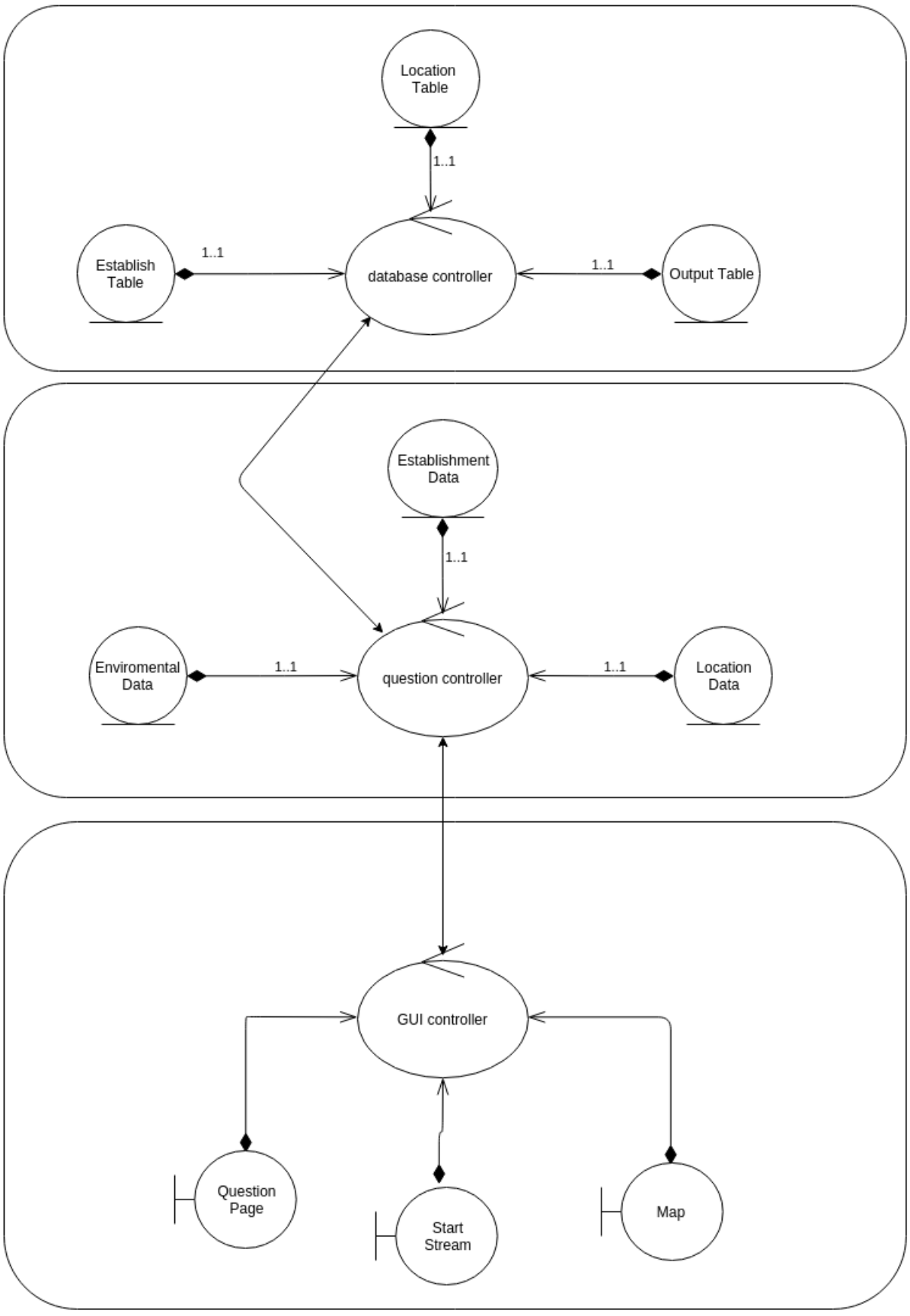

Figure 6: The analysis class diagram