

Specification Document

Project Name: QR Quest

Team Number: 2



|  |  |  |
| --- | --- | --- |
|  |  |  |

Contents

[Document Information 3](#_Toc476926423)

[Version History 3](#_Toc476926424)

[Purpose 4](#_Toc476926425)

[Background & Analysis 5](#_Toc476926426)

[Analysis Process 5](#_Toc476926427)

[Analysis of Similar Systems 5](#_Toc476926428)

[Analysis of Environment 6](#_Toc476926429)

[Roles and Deliverables 7](#_Toc476926430)

[Team Roles 7](#_Toc476926431)

[Deliverables 7](#_Toc476926432)

[Project Plan 8](#_Toc476926433)

[Hardware and software platforms to be used for developing and running your solution 8](#_Toc476926434)

[Definition of terms 9](#_Toc476926435)

[Solution requirements 10](#_Toc476926436)

[Functional Requirements 10](#_Toc476926437)

[Non-Functional Requirements 11](#_Toc476926438)

[Other considerations 11](#_Toc476926439)

[Assumptions 11](#_Toc476926440)

[Constraints and Dependencies 11](#_Toc476926441)

[Software Design 12](#_Toc476926442)

[Overview of what the software will do and not do. 12](#_Toc476926443)

[Modifications in light of comments made or changes deemed necessary 12](#_Toc476926444)

[Logo Creation 12](#_Toc476926445)

[Initial Interface Designs 13](#_Toc476926446)

[Use Case Diagram 14](#_Toc476926447)

[Component Diagram 14](#_Toc476926448)

[Activity Diagrams 15](#_Toc476926449)

[Class Diagram 17](#_Toc476926450)

[Information within application are meant to be saved in objects. The following class diagram describe the mainstay classes of the system. 18](#_Toc476926451)

[Behaviour Diagram 19](#_Toc476926452)

[Entity Relationship Diagram 20](#_Toc476926453)

[Test Strategy 20](#_Toc476926454)

[References 21](#_Toc476926455)

[Contribution Matrix 21](#_Toc476926456)

# Document Information

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
| Project Name: | QR Quest | | |
| Prepared By: | Team 2 | Preparation Date: | 13/12/2016 |
| Email / Phone: | H.Hornby1@ncl.ac.uk |  |  |
| Document Version No: | 1.8 | Document Version Date: 10/03/2017 |  |
|  |  |  |  |

# Version History

| Ver. No. | Ver. Date | Revised By | Description |
| --- | --- | --- | --- |
| 0.1 | 13/12/16 | H Hornby | Initial draft. |
| 0.2 | 14/12/16 | B Naji | Updated solution requirements. |
| 0.3 | 14/12/16 | A Chow | Updated platforms, references, other considerations and definition of terms. |
| 0.4 | 21/02/17 | J Charsley | Updated analysis. |
| 0.5 | 23/02/17 | H Hornby | Updated platforms, references and other considerations. |
| 0.6 | 24/02/17 | H Hornby | Updated roles and deliverables and project plan. |
| 0.7 | 27/02/17 | H Hornby | Revision of spec design and layout. |
| 0.8 | 28/02/17 | H Hornby | Updated analysis. |
| 0.9 | 02/03/17 | H Hornby | Proof reading and revision of all sections. |
| 1.0 | 04/03/17 | B Naji | Updated solution requirements |
| 1.1 | 06/03/17 | H Hornby | Updated solution requirements and other considerations |
| 1.2 | 07/03/17 | H Hornby | Added UML Diagram, ER Diagram, updated software design and added contribution matrix. |
| 1.3 | 08/03/17 | H Hornby | Updated other considerations, platforms and solution requirements |
| 1.4 | 08/03/17 | B Naji | Added activity diagrams |
| 1.5 | 08/03/17 | V Cody | Added original design diagrams |
| 1.6 | 09/03/17 | L Iley | Added logo creation section |
| 1.7 | 09/03/17 | B Naji | Update activity and class diagrams, add use case diagram |
| 1.8 | 10/03/17 | H Hornby | Added test strategy, updated contribution matrix, final formatting. |
| 1.9 | 10/03/17 | B Naji | Added class diagram, behaviour diagram, and supporting text. |

# Purpose

This document is intended to describe formally - in thorough detail - the structure, design and layout of the QR Quest mobile application. It analyses the software’s whole design from the ground up through provision of detailed diagrams and explanations as well as analysis of other systems. The document covers the software’s goals and requirements, its constraints and dependencies, its interactions between software and hardware and also its limitations. The document also provides information about the development team, the project’s scope and timeframe and also includes a list of key terms and references.

A full list of the specifications sub sections and their purpose can be found below:

**Background & Analysis:**

A description of the environment in which the software is being deployed, the reason for the software’s creation and an analysis of already existing solutions.

**Roles and Deliverables:**

The roles and responsibilities of each team member and a description of what will be delivered to the customer.

**Project Plan:**

A detailed plan of the whole project including soft and hard deadlines, dependencies and a full list of tasks.

**Hardware and Software Platforms:**

The hardware and software requirements for developing and running the solution.

**Solution Requirements:**

An outline, description and weighting of the software’s required functionality.

**Other Considerations:**

Analysis of constraints and assumptions for the software.

**Software Design:**

A full breakdown of the software’s design achieved through the provision of detailed diagrams and analysis.

**Test Strategy:**

A description of how we will test the software, including the types of tests we will perform.

**References:**

A list of references used to create the software and any documentation.

# Background & Analysis

## Analysis Process

Our domain analysis consisted of two different sections. Firstly, we researched similar systems available on the marketplace, assessing the merits and drawbacks of the respective applications. We also looked to see if each application had a feature that they chose to focus on, and if there was a common theme between them. The second section we considered was the environment that the application would be required to run in and how this would affect the application as well as any problems we may have e.g. features that we would have to consider removing.

## Analysis of Similar Systems

**Seek and spot**

This application seemed to present itself for groups who would use it occasionally, but not frequently enough to create accounts and partake in public hunts. It also seemed to have an emphasis on question related clues where the user would usually be give a multiple-choice question, however, it did seem to contain alternative clue types.

Useful features of the app were a guest login for those who would need the app for an activity but would not wish to make an account if they knew they were not going to use the app afterwards. This also simplifies the initial setup as participants would not need to create an account before they could take part. Another helpful feature was allowing creators to set a password for a hunt if it was only for a private group, a feature we had not considered.

Drawbacks to Seek and Spot included a cluttered interface with multiple buttons that you could only tell what they would do once you pressed them. Additionally, it can be difficult to find a hunt that you can go on as most of the hunts shown are either private or in other countries. This could be solved by ordering the hunts by the distance away from you and by having a checkbox to not display private hunts.

**SCAVIFY**

This application seemed to be much more minimalistic with simpler buttons and less clutter on the screen. They may however have gone too far as it is difficult to do much on the app without knowing of a particular hunt. It also seemed to be much more varied than Seek and Spot, containing multiple clue types and a score based system.

Features we liked included having a much more variety in its clues, allowing for GPS, QR codes, multiple-choice questions, images and others. It was also much nicer to look at and their score based system brings more motivation to complete hunts.

Issues with the system were that hunts would only appear if you search specifically for them and therefore if you don’t know that a particular hunt is happening, then it is difficult to find a hunt that is nearby, public and still going on. Additionally, there is no ability to create hunts within the app and very few settings.

**Geocache**

This application’s approach is slightly different from the previous apps we researched; rather than having trails of multiple clues that are completed by using GPS, a question or QR codes, Geocache focuses more on individual caches that result in finding a hidden box containing a list of people who have found it and sometimes several items that can be traded.

Helpful features in geocache include selecting the cache you want to partake in from a map. This makes finding a route that you can go on much easier. One issue with the app was the reliance on single caches and finding items rather than a more integrated approach. Although these are not necessarily bad ideas, they would not be suitable for the type of application that we would be creating.

## Analysis of Environment

Each year, the Newcastle University School of Computing Science takes first year students on a treasure hunt around the campus in order for them to learn their way around, learn more about the buildings around campus and also get used to working with their teams. The current system provides each team with a list of clues on a piece of paper, they must then work out their destination from the clues. Once a team arrives they meet with a member of staff who gives the team some information about the location and sends them off to their next destination.

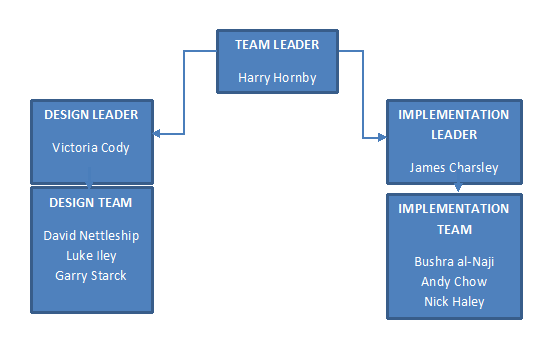
This system uses a lot of resources, including large amounts of paper and the help of a lot of staff. This is expensive and time consuming, and so by taking the treasure hunt and transforming it into a mobile application much of the expense is saved. The stakeholders of this system would include the quest organisers, who would create the quests, the students partaking in the treasure hunt, and the university administrators who would be responsible for university property.

The application must be easy to set up as all teams would be required to download it and use it. Additionally, there should be an ability to make a session private as some of the areas may only be accessible by university students. Therefore, a user based system should be implemented. As many of the clues are likely to lead to locations within buildings, there may some issue with GPS, especially when dealing with multiple floors.

Using QR codes could also potentially cause issue with university administrators as the organisers may not have access to university property such as the walls in order to place to place the QR codes. We have therefore chosen to provide multiple clues types, such as GPS, QR codes or image based clues so that there are a range of options for the user and creator of the hunt.

# Roles and Deliverables

## Team Roles



**Team Leader**

The team leader’s role is to oversee the whole project, organising a project plan, handling deadlines and leading team meetings as well as distributing work between the entire team to ensure a fair workload. The team leader is not a member of a specific sub-team and instead steps in to help wherever work needs to be done.

**Implementation Team**

They work together and individually to develop the application, dividing the features between themselves. As well as programming, the team performed research for the application and provide UML diagrams for their work. The team has a lead programmer who organises meetings and distributes tasks within the team.

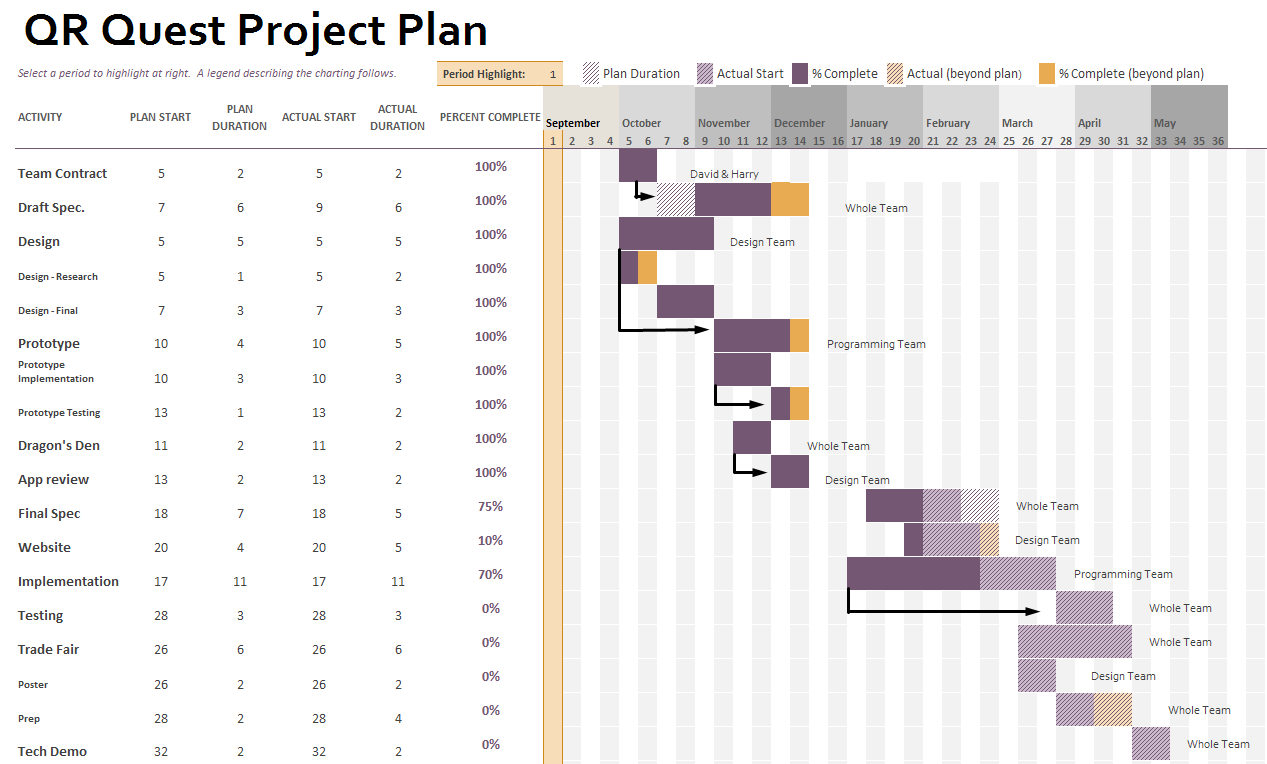
**Design Team**

They designed the application from the ground up including detailed diagrams, feature lists and UI designs. In addition to this the design team work on most of the documentation with some help from the programming team. The team has a lead designer who organises meetings and oversees the work done by the team.

## Deliverables

We will deliver to the customer an Android mobile application (.APK file), available for download from our website, with the inclusion of a user guide.

# Project Plan



# Hardware and software platforms to be used for developing and running your solution

Software:

* Android App
* Android Studio 2.2.3
* Notepad++ 7.2.2
* Adobe Photoshop CS3, CS5
* Android Virtual Device running Android 4.03 or higher
* OS capable of running Android Studio

Hardware:

* Mobile System
* Mobile device running Android 4.03 or higher
* Mobile device with a constant internet connection
* Mobile device with GPS functionality
* PC which can run required software

# Definition of terms

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **3G/4G** | These are the third and the fourth generation of wireless mobile telecommunications technology |
| **Wi-Fi** | This allows devices to connect to the internet via a router |
| **SDK** | Software Development Kit – a set of tools used to create applications for a software package |
| **App** | Application - this is usually a small piece of executable software for use on a desktop or mobile device |
| **UI /GUI** | User Interface - this is a way for the user to interact with the software with a mouse and keyboard, it can consist of icons or a command terminal to enter commands. |
| **OS** | Operating System - examples include Windows, Linux and Android. This allows users to interact with the hardware in the computer/mobile device and is necessary to run applications. |
| **GPS** | Global Positioning System – a global navigation satellite system that provides geolocation and time information to a GPS receiver in all weather conditions, anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites. |
| **PC** | Personal Computer – In this case refers to any kind of computer including laptops or desktops. |
| **QR Quest** | The application being designed and implemented. |
| **UML** | Unified Modelling Language used to represent programs via diagrams of various types. |
| **API** | An application programming interface |
| **QR Code** | A machine-readable code consisting of an array of black and white squares, typically used for storing URLs or other information for reading by the camera on a smartphone. |
| **Quest** | An activity on the app which involves users solving clues in order to find the next location to go to, essentially a treasure hunt. |
| **Entity Relationship (ER) diagram** | A diagram which shows the relationships between tables in the software’s database tables. |

# Solution requirements

## Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | Priority | Supplier  Compliance | Supplier  Comment |
| 1. The login activity should transfer users to either a registration activity or a menu activity | H | Full |  |
| 1. The registration activity should enable new users to create a user account | H | Full |  |
| 1. The app should enable users to construct new treasure hunts (Quests) | H | Full |  |
| 1. The app should enable users to add clues to a created treasure hunt | H | Full |  |
| 1. The app should match each created clue to a location and QR code | H | Full |  |
| 1. The app should accept correct scanning results only within a certain distance from solution location | H | Full |  |
| 1. The app should allow users to modify/edit their existed Quests | H | Full |  |
| 1. The app should enable users to modify/edit existing clues within their existing Quests | H | Full |  |
| 1. The players map view should point out their location on the map | H | Full |  |
| 1. The app’s QR code scanner should validate whether the scanned code is correct or not | H | Full |  |
| 1. The app should forward players to next clue whenever they solved one clue | H | Full |  |
| 1. The app should allow users to sign out of their account | H | Full |  |
| 1. The app should enable users to send QR Codes images to their email | H | Full |  |
| 1. The app should allow users to update their account details | M | Full |  |
| 1. Scanning QR Codes should be within a reliable distance from solution’s location | M | Full | Placing QR Codes is not always allowed in certain areas |
| 1. The app should enable users to play either one of their own Quests or public Quests | M | Full |  |
| 1. The app should allow users to save unfinished quest game and complete them later on | M | Full |  |
| 1. The users should be able to search for a Quest from quest list by typing its name in search field. | M | Full |  |
| 1. The app should allow users to change colour scheme | L | Full |  |
| 1. The app should allow users to change font size | L | Full |  |
| 1. The app should allow users to stop receiving notifications | L | Full |  |
| 1. Creators of quests should get alerts whenever a player finishes the game | L | Full |  |

## Non-Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | Priority  (H, M, L) | Supplier  Compliance (Full, partial or will not be delivered) | Supplier  Comment |
| 1. Time to transfer users between activities no longer than 4s | H | Full |  |
| 1. Registration emails send with latency of no greater than 5 minutes for new accounts | H | Partial | As suppliers we do not always have full control of email latency |
| 1. Scanning QR codes should have a latency no greater than 10s | H | Full |  |
| 1. Updated and new Quests should appear on the database for users to search within 5 minutes | H | Full |  |
| 1. Emails containing QR codes should send with latency of no greater than 5 minutes. | H | Partial | As suppliers we do not always have full control of email latency |
| 1. Notifications from the app should have latency no greater than 5s | H | Full |  |
| 1. Loading times for Quests should not exceed 20s | H | Full |  |

# Other considerations

## Assumptions

* The given QR Codes are in the intended positions and are left undisturbed.
* The maps provided by the API is up to date.
* The user has an Android phone running version 4.03 or above.
* Locations given as clues are legally and physically accessible to the public.

## Constraints and Dependencies

* The app requires constant internet connection.
* The app requires a phone with GPS functionality.
* The app relies on physical QR Codes and GPS which are not 100% reliable.
  + QR Codes can be physically removed.
  + GPS Signal not always available.

# Software Design

## Overview of what the software will do and not do.

As per the requirements, the software **WILL**:

* Allow the user to create Quests with personalised clues and solutions.
* Allow users to play their own and other’s Quests in teams of up to 10.
* Allow users to search for Quests to play.
* Allow users to complete Quests by scanning QR Codes.
* Allow users to complete Quests based on GPS location.
* Allow users to edit accessibility features such as font size and colour scheme.
* Allow Quests to be edited and updated once created.
* Allow users to edit account details.
* Allow guest accounts to be created for first time players.
* Allow new users to register accounts on the app.

The software **WILL NOT:**

* Provide any form of useful navigation for the user, the map is purely for tracking your own location.
* Store any personal information on the user other than the fields provided (email / username).
* Give location information to 3rd parties.
* Email the user with unwanted promotions or adverts.

## Modifications in light of comments made or changes deemed necessary

* Instead of a separation between students and staff the app is now separated into two distinct sections, one for creating the treasure hunts and one for partaking. This changed in order to expand the audience of the app and provide an easier interface for creating the hunts as it now does not depend on a staff member being present.

## Logo Creation

When designing the logo we wanted something simple but effective. We began by doing some market research into what makes an effective logo.

We decided that we wanted our logo to be simple and memorable, it had to be suitable on all backgrounds and work in any size (for posters, flyers etc.).

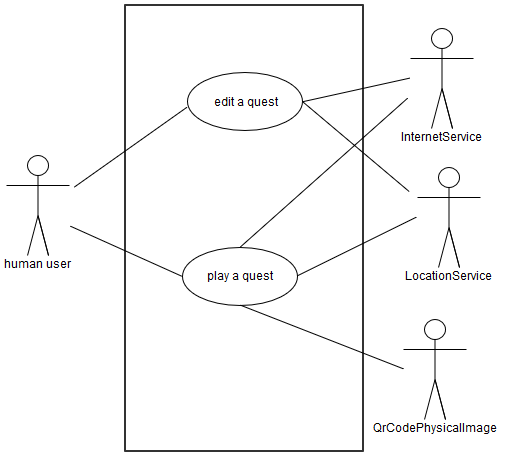
Because we’d decided on the name QR Quest we wanted to try creating the logo using nothing more than the words themselves. We came up with a few early designs, but we thought they were missing something.

Continuing work on the logo, we tried adding simple images and changing the colour of the text. In the end, we settled with a design similar to what we’d been looking at but with a little bit of colour.

The final logo sticks to our original plan by being simple but memorable, with a design that we believe can be transferred to any media. This minimalist design perfectly represents our app.

## Initial Interface Designs

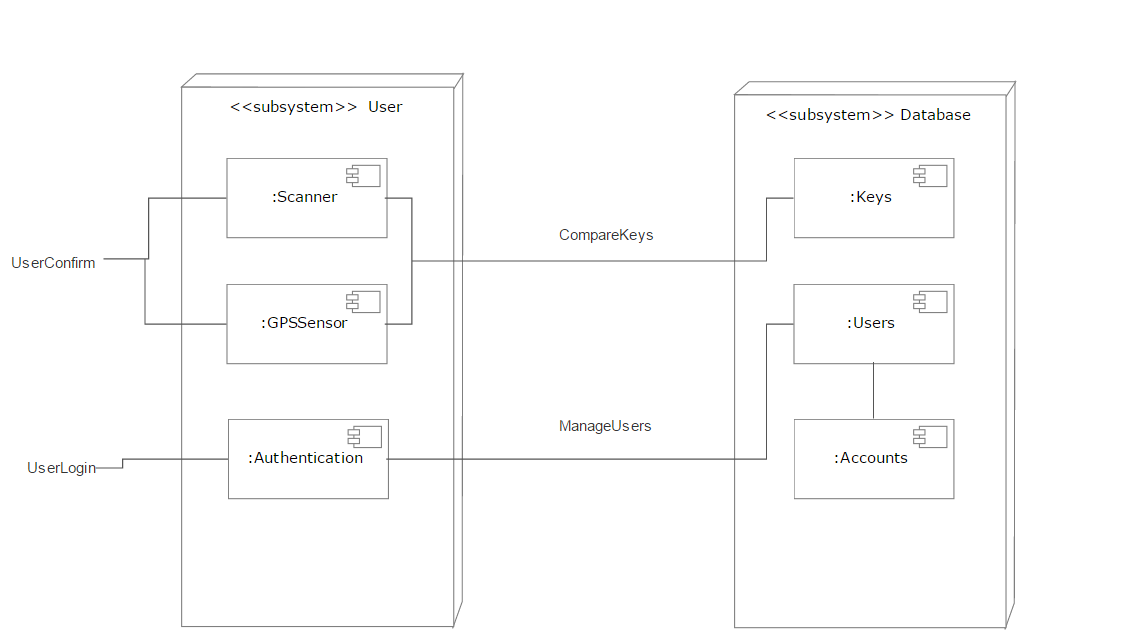




## Use Case Diagram

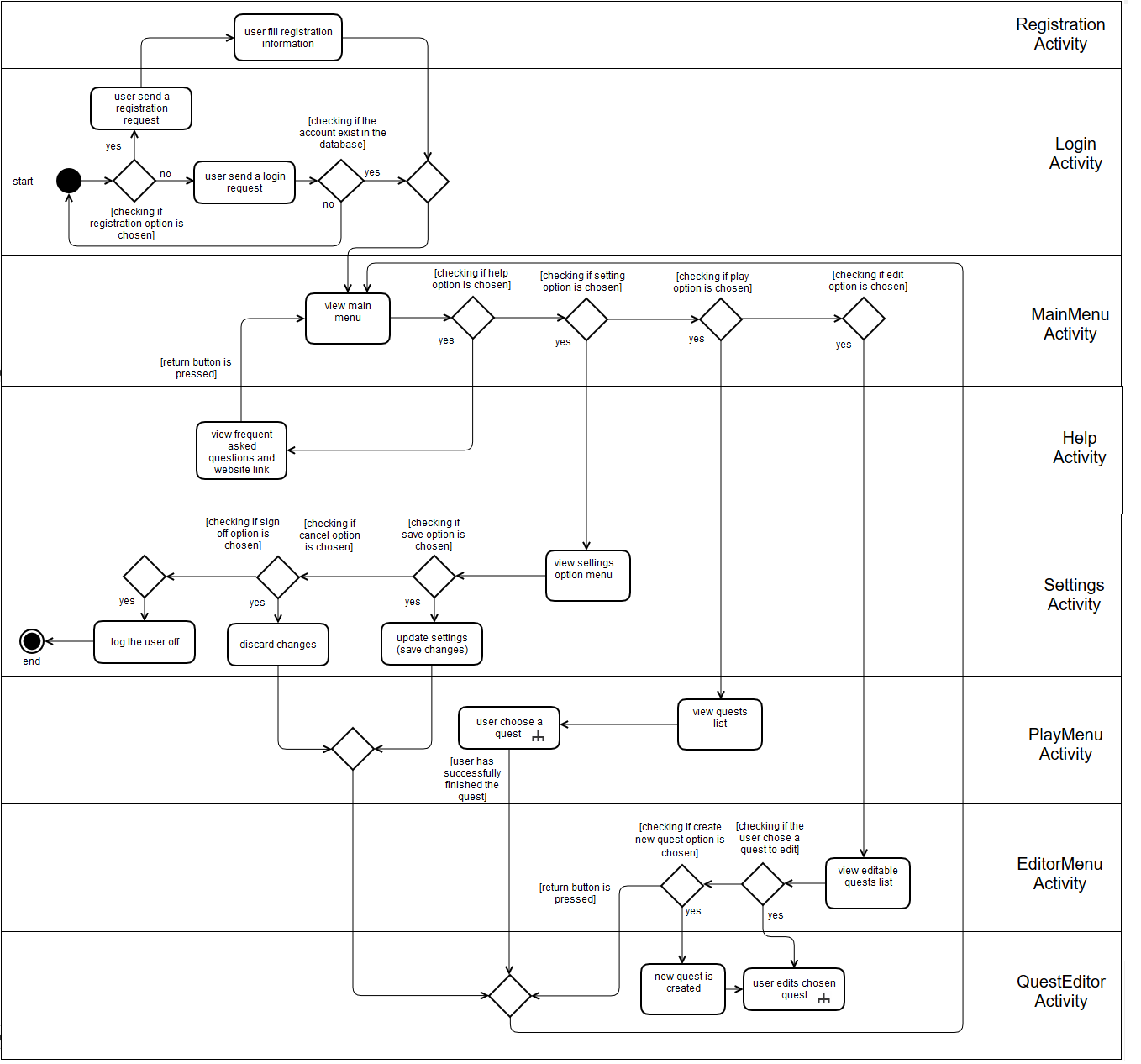
The diagram on the right side demonstrates use cases (actions) and their relations with various subjects (actors) as well as external requirements. ‘edit a quest’ use case requires human, internet service, and location service participations and ‘play a quest’ use case require QR code printed images along with the other three mentions participants.

## Component Diagram

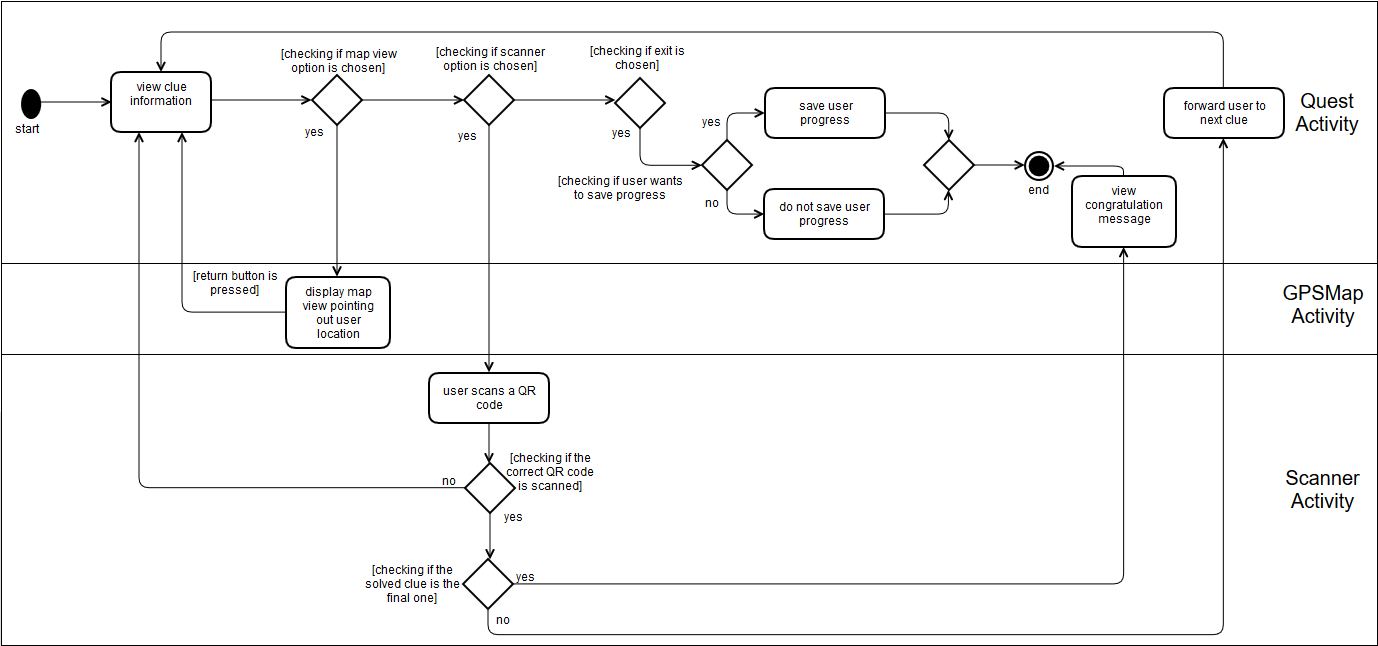
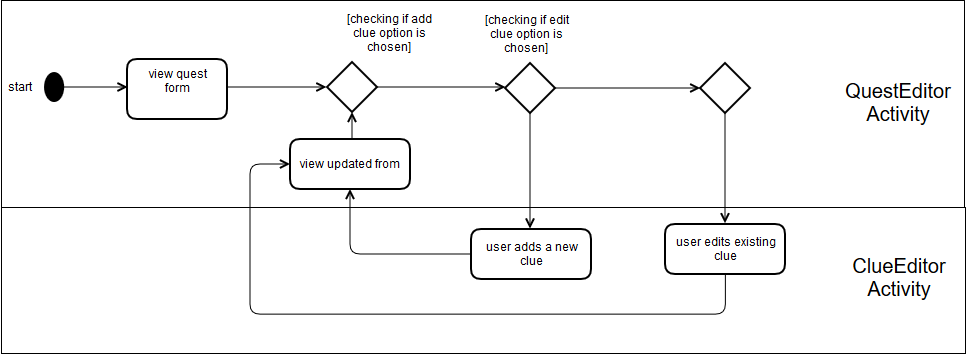


## Activity Diagrams

The following diagram describes the application’s dynamic behaviours starting from logging in to signing off. It also illustrates the potential workflows of activities taking in consideration decisions which users could take. Within this diagram, there are two call behaviours ‘user chooses a quest’ and ‘user edits chosen quest’ which are explained in more detailed by the followed ‘Quest play navigation’ and ‘Quest edit navigation’ diagrams respectively.



1. Overall system navigation

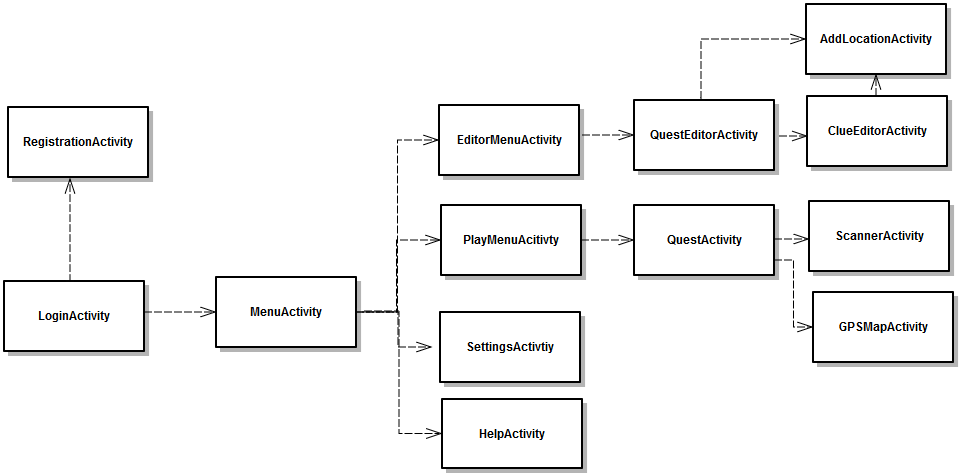


3. Quest edit navigation

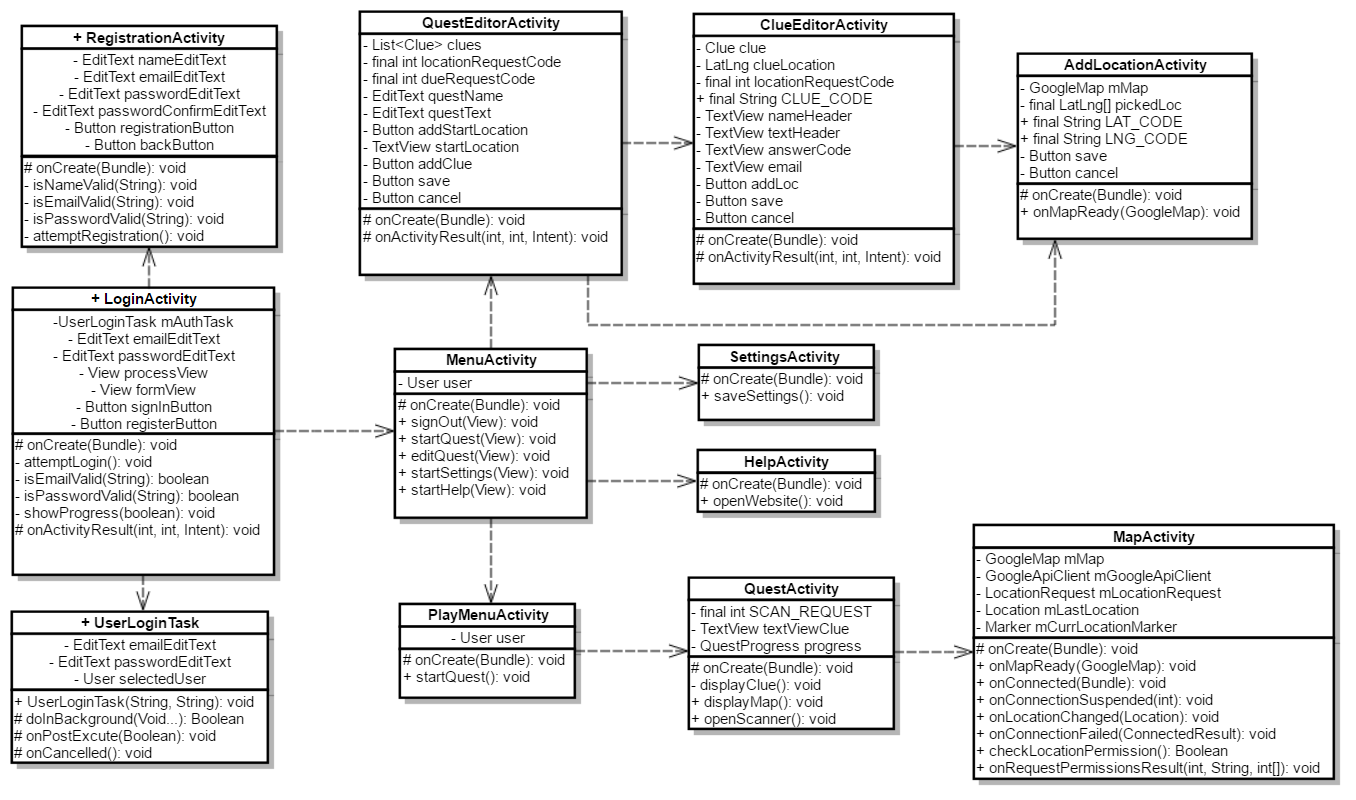
2. Quest play navigation

## Class Diagram

The following diagram illustrates the general static structure of the application. Classes are in relation with each other in term of dependency.

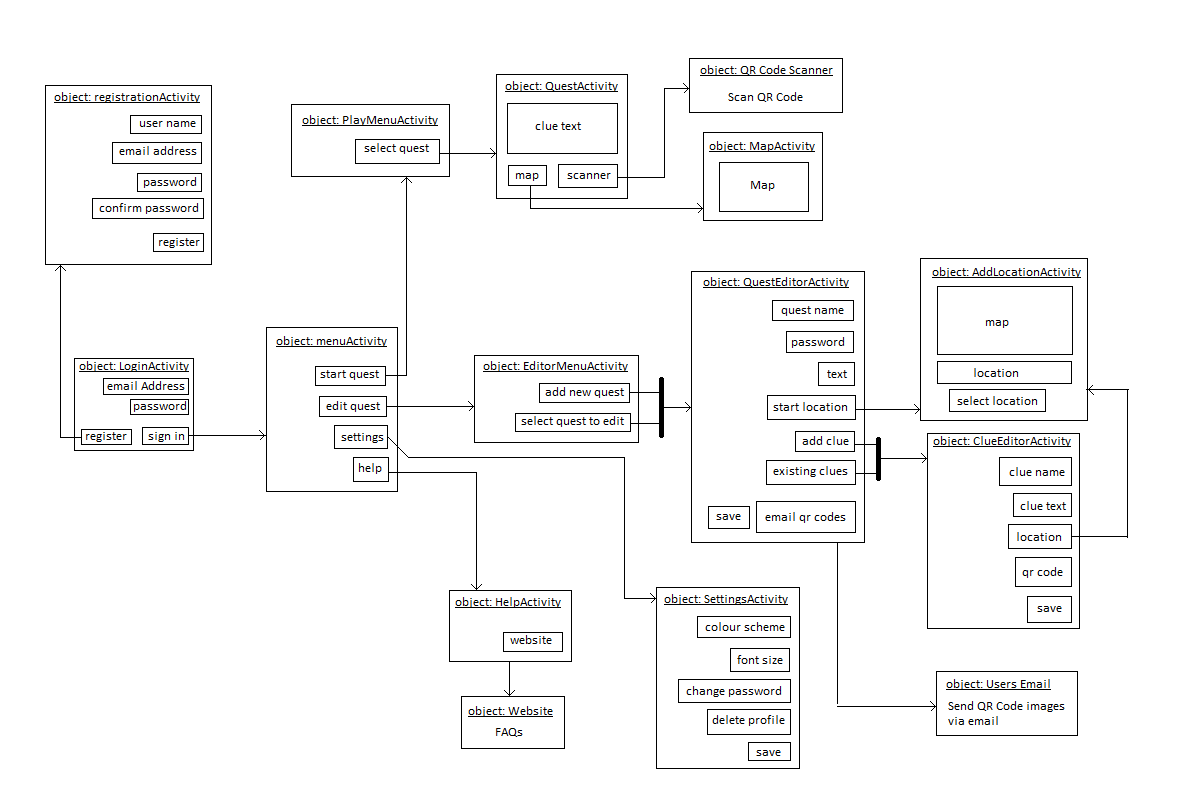


The following diagram is the detailed version of the above diagram. It includes classes details which are divided into two sections operations and attributes.



## Information within application are meant to be saved in objects. The following class diagram describe the mainstay classes of the system.

## Behaviour Diagram



## Entity Relationship Diagram

# Test Strategy

|  |  |
| --- | --- |
| Test Series: | Purpose of test series: |
| 1 – Interface Testing | Test that every button leads to where it should, and that all menu options produce the correct result. Ensure that all messages are not blank and produce the correct data. |
| 2 – Validation Testing | Test to see if validations of username and password are correct, and that features such as QR Code validation are working, and that they do not upset normal data flow (producing errors where non exist etc.). |
| 3 – Integration / System Testing | Check to see if SQL commands return the correct data, and that all table relationships are working. Test that all data is written to the correct tables within the database and all validation on such data is not interfering with the reading and writing of data. |
| 4 – Black Box Testing | Test against the requirements to see if they have all been met, are fully functional and that no incorrect error messages are produced when running through these processes. |

# References

Android Studio: https://developer.android.com/studio/index.html

Android Studio training: <https://developer.android.com/training/index.html>

Android Studio tutorials point: <https://www.tutorialspoint.com/android/android_studio.htm>

UML Activity Diagrams References: <https://msdn.microsoft.com/en-us/library/dd409360.aspx>

UML Use Case Diagram: <http://www.uml-diagrams.org/use-case-diagrams.html>

# Contribution Matrix

