



**SEG Major Project  
Project Number 5**

**Team Tenacity 8-Hobbytes = 1 Hobbyte members:**

Miriam Tamara G. Aarag - 1601261,

Julius Babenskaskas - 1612647,

Funke Sowole - 1617272,

Jeongho Suh - 1504262,

George Thornton - 1617550,

Kaitlin Villa-Fernandez - 1501227,

Yafan Yin - 1661173.

**Project Brief:** Being-A-Parent app

**Client:** Dr Joshua Harwood

**Organisation:** NHS South London and Maudsley Parents intervention group

**Platform:** Androids, iOS | **Backend:** Firebase

**Android Name:** TeamTenacityMajorProject

**iOS Name:** TeamTenacityMajorProject

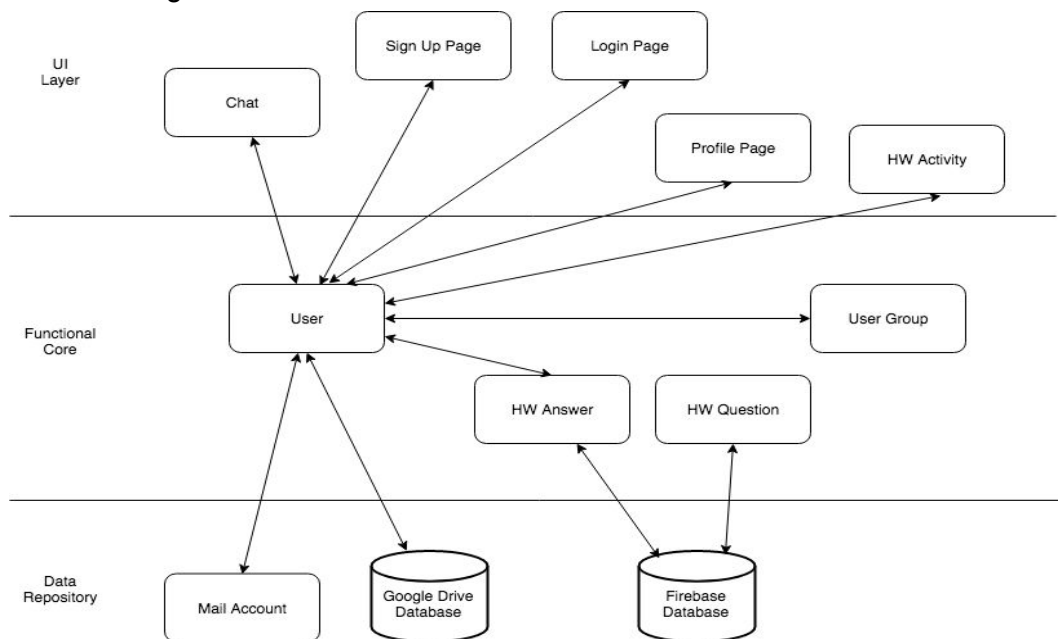
I.

## II. Introduction

We were asked by Dr Joshua Harwood, an NHS Clinical Psychologist working for the Centre for Parent and Child Support (CPCS) to create an app for parents with very young and newborn children. Empowering Parents, Empowering Communities (EPEC) is a program run by the CPCS. Its goal is to provide support for parents who have little experience raising children. As stated in the CPCS report, very few parents attend the drop-in, mostly due to lack of accessibility, so the purpose of the app is to allow parents who are nervous to attend the drop-in sessions to view the course material remotely. Our client hopes that this would encourage the parents to attend the drop-in sessions.

## III. Outcome

Architectural Diagram:



| <u>Android</u>  | <u>iOS</u>   |
|---|--|
| <ul style="list-style-type: none"><li>● <b>Firestore</b><ul style="list-style-type: none"><li>○ Auth</li><li>○ Database</li><li>○ Storage</li></ul></li><li>● <b>CSV Library</b></li><li>● <b>Picasso</b></li><li>● <b>Messages Library</b></li></ul> | <ul style="list-style-type: none"><li>● <b>Firestore</b><ul style="list-style-type: none"><li>○ Auth</li><li>○ Database</li><li>○ Storage</li></ul></li><li>● <b>Replicode</b><ul style="list-style-type: none"><li>○ Re-used Login/Sign Up, Chat Code (Robert Canton)</li></ul></li><li>● <b>Messages Library</b></li><li>● <b>Pod File</b></li></ul> |

Team Tenacity 8-Hobbytes = 1 Hobbytes

Miriam Tamara G. Aarag - [1601261](#)

Julius Babenskaskas - [1612647](#)

Funke Sowole - [1617272](#)

Jeongho Suh - [1504262](#)

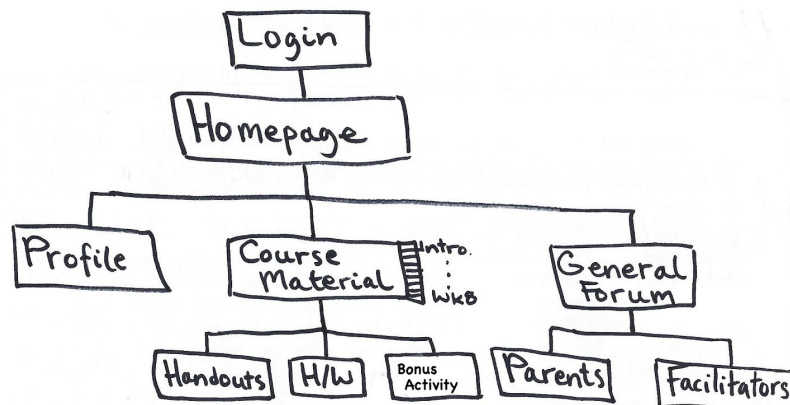
George Thornton - [1617550](#)

Kaitlin Villa-Fernandez - [1501227](#)

Yafan Yin - [1661173](#)

## IV.Design

The following diagram is the diagram we drew up for the client based on his initial requirements. The diagram contains an overview of the main components of the app and how they are connected to each other.



The next diagram contains an overview of the app's current and final design. The diagram shows how the application moves between different activities. Each UML class in the diagram represents a different activity. An arrow from one activity to another means that the activity being pointed to is started by the activity pointing to it. Note that the HomepageActivity, CourseWorkActivity, ChatRoomActivity, and ProfileActivity can be started by the HomepageActivity and any of its subsequent activities. However, for the sake of keeping the diagram simple, these arrows have been omitted. When the application is first started it will launch the MainActivity, as denoted by the empty circle next to this activity. The diagram classes do not contain all the attributes and methods owned by the activity, but rather a smaller selection of these to help illustrate the activity's primary functionality.

For the app's design, it has been divided into several separate features, such as the user having to register to use the app, the user being able to answer homework questions etc. This design makes it possible to update a specific feature easily. To incorporate this design pattern the app has been designed in such a way that each activity handles exactly one main feature. For example the LoginActivity allows the user to login, and the UserRegistrationActivity allows the user to register for the app. Each activity is also associated with it's own unique layout to support allocating tasks in such a way.

The app follows a factory pattern. The factory pattern is a creational pattern which allows objects to be created without having to specify the exact class of the object. This is achieved by calling a factory method rather than a constructor. The factory method is usually implemented in an interface or abstract class, and makes sure that the new object is created. The factory pattern promotes loose coupling as the different classes interact only with the interface/abstract class and not directly with each other. In the developed application the factory pattern has been incorporated through the use of the Android class Intent. Intent is an abstract description of an operation to be performed. In this specific

Team Tenacity 8-Hobbytes = 1 Hobbytes

Miriam Tamara G. Aarag - [1601261](#)

Julius Babenskias - [1612647](#)

Funke Sowole - [1617272](#)

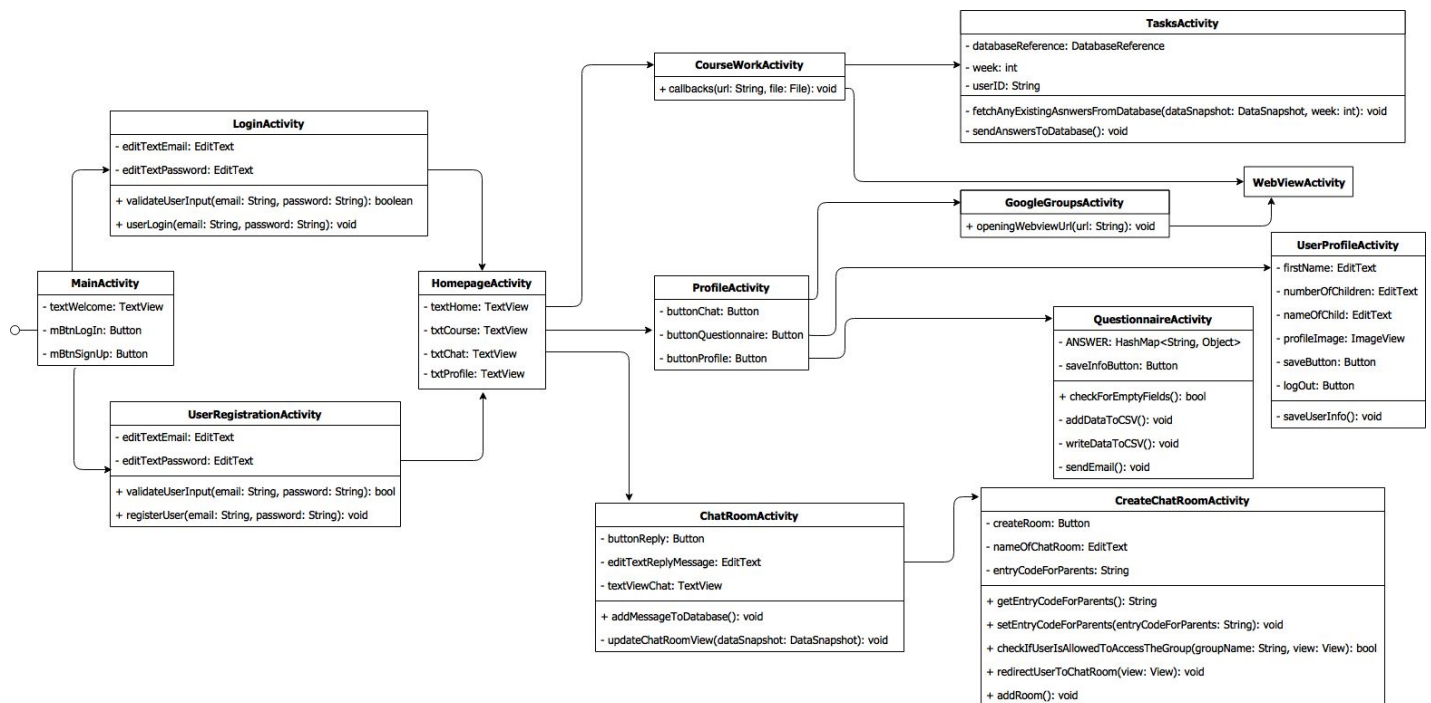
Jeongho Suh - [1504262](#)

George Thornton - [1617550](#)

Kaitlin Villa-Fernandez - [1501227](#)

Yafan Yin - [1661173](#)

case Intent has been used to launch activities. This structure ensures that none of the activities in the app ever interact directly.



The design patterns used for iOS include a reusable table view cell. Also de-queuing a table view cell when it is no longer in use. This can be seen in the Homework Responses for instance, where the app reads the List of questions from the Firebase database, and presents them to the parents to allow them to answer their homework. iOS tries to improve efficiency in many ways, so they keep they keep in memory only the cells which are visible. It then dequeues the other cells from memory when they are no longer in use. Therefore, TableViewCell also uses the Iterator pattern, as it iterates through the list of cells the app is to display.

**Team Tenacity 8-Hobbytes = 1 Hobbytes**

Miriam Tamara G. Aarag - [1601261](#)

Julius Babenskaskas - [1612647](#)

Funke Sowole - [1617272](#)

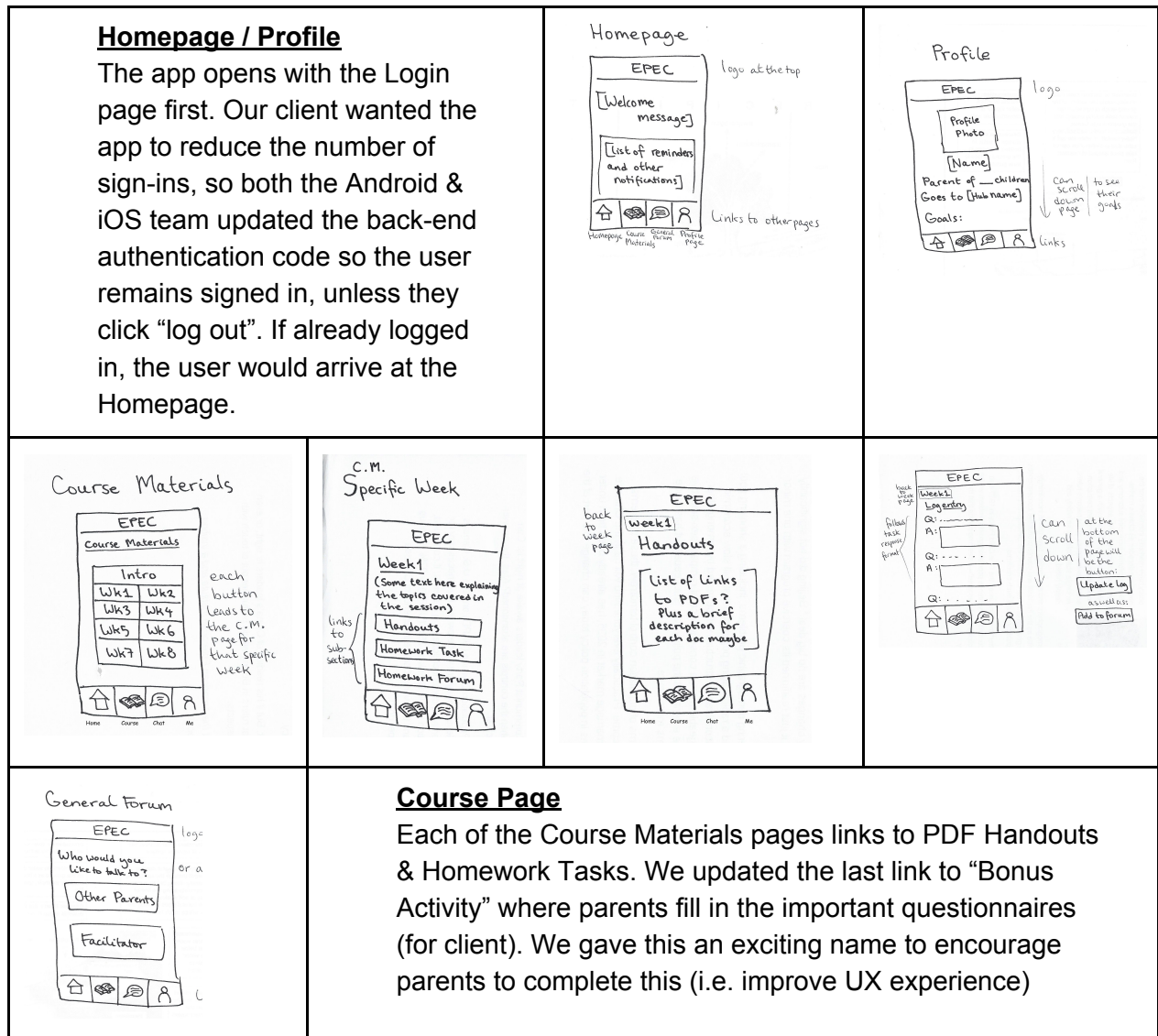
Jeongho Suh - [1504262](#)

George Thornton - [1617550](#)

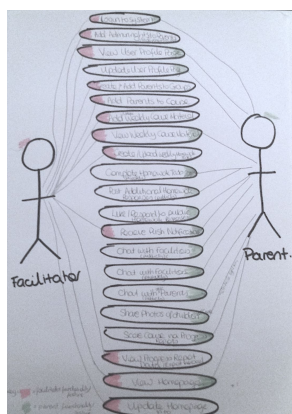
Kaitlin Villa-Fernandez - [1501227](#)

Yafan Yin - [1661173](#)

## Layout Design



## Use Case Diagram



Link to user case and user story:

<https://github.kcl.ac.uk/K1630494/TenacityMajorProject/wiki/Design:-Use-Case-Story>

Team Tenacity 8-Hobbytes = 1 Hobbytes

Miriam Tamara G. Aarag - [1601261](#)

Julius Babenskaskas - [1612647](#)

Funke Sowole - [1617272](#)

Jeongho Suh - [1504262](#)

George Thornton - [1617550](#)

Kaitlin Villa-Fernandez - [1501227](#)

Yafan Yin - [1661173](#)

## V. Software Quality

We conducted unit tests and ensured code quality for both Android and iOS.

### Both Teams

Both teams ensured that the program was free from bugs and maintained clean code. For instance both Android and iOS team uses author / java docs, commented the code meaningfully, gave relevant class names, kept methods short and meaningful, we used features branches, and also gave relevant commit messages. As stated in our kanban board, everyone is largely responsible for testing their own working increment.

### For iOS

The majority of tests done for iOS would be unit tests for the login view controller. We used the integrated XCode test editor (XCTest). Below details how code coverage tests are carried out for iOS which are pre-installed with XCode.



After running a few XCTest tests on the code, Xcode would automatically check the code coverage statistics and present them to you.

### For Android

The Login and Registration pages were tested that are both part of the Feature to allow parents & facilitators to log-in and register to the system. The tests ensure that only a password which is greater than 6 characters, redirects users to the homepage activity appropriately, ensures email is in a correct format and that email and password fields are not empty. These login tests are also present for iOS.

Firebase has automated tests for android, which do not require you to input any code. Once your app is connected to the Firebase database, Firebase would automatically run tests to ensure that your app displays correctly for a range of different devices such as Nexus to LG, and android versions dating back to Android Studio version 2.0 (We currently use 4.3). The process is as easy as connecting the Firebase test file into your Android tests, and Firebase automatically would send UI test report, and fix any errors.

**Team Tenacity 8-Hobbytes = 1 Hobbytes**

Miriam Tamara G. Aarag - [1601261](#)

Julius Babenskias - [1612647](#)

Funke Sowole - [1617272](#)

Jeongho Suh - [1504262](#)

George Thornton - [1617550](#)

Kaitlin Villa-Fernandez - [1501227](#)

Yafan Yin - [1661173](#)

## VI. Team Organisation

### a) Methodology Used

We used the Agile Methodology during our project. We:

- i) We used a Kanban board to maintain our workflow and updated it regularly.
- ii) We worked in "Sprints".

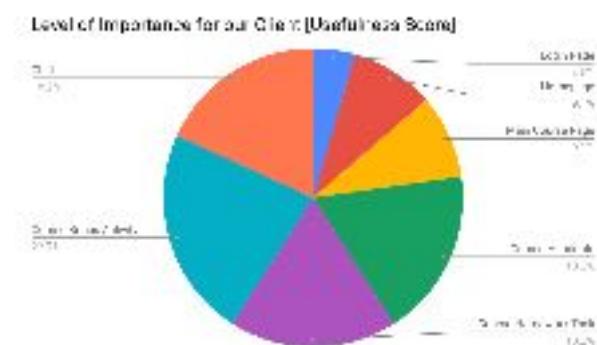
### b) Team Roles

To deal with developing for both IOS and Android, the group was divided into teams. Over time there were some changes in the number on people on each team, but at the end the group was organised as follows.

- Android: Julius, Tamara, Yafan
- IOS: Funke, George, Jengho
- Design Documents: Kaitlin

Julius was the leader of the Android team, and Funke was the leader of the IOS team. Their roles were to merge to merge features branches to master and to keep an overview of their respective teams and what they were working on. Tamara was project manager and client liaison. Her job was to have an overall overview and communicate with our client.

- |   |   |
|---|---|
| <p>1. <b>Login Page</b></p> <ol style="list-style-type: none"> <li>i. Android: Julius</li> <li>ii. IOS: Funke</li> </ol> <p>2. <b>Home Page</b></p> <ol style="list-style-type: none"> <li>i. Android: Yafan</li> <li>ii. IOS: George</li> </ol> <p>3. <b>Main Course Page</b></p> <p><u>Course PDF Handout</u></p> <p>Android: Jeongho</p> <p>IOS: Jeongho</p> <p><u>Course Homework Tasks</u></p> <p>Android: Tamara</p> <p>IOS: Funke</p> <p><u>Course Measures</u></p> <p>Web-based: George</p> | <p>4. <b>Chat</b></p> <ol style="list-style-type: none"> <li>i. Android: Julius</li> <li>ii. IOS(web-based): Funke</li> </ol> <p>5. <b>Profile</b></p> <ol style="list-style-type: none"> <li>i. Android: Julius</li> <li>ii. IOS: Funke</li> </ol> <p>6. <b>Architecture and Page Design</b></p> <ol style="list-style-type: none"> <li>i. Kaitlin</li> </ol> <p>7. <b>Documentation</b></p> <ol style="list-style-type: none"> <li>i. Everyone</li> </ol> |
|---|---|



#### Team Tenacity 8-Hobbytes = 1 Hobbytes

Miriam Tamara G. Aarag - [1601261](#)  
 Julius Babenskaskas - [1612647](#)  
 Funke Sowole - [1617272](#)  
 Jeongho Suh - [1504262](#)  
 George Thornton - [1617550](#)  
 Kaitlin Villa-Fernandez - [1501227](#)  
 Yafan Yin - [1661173](#)

The above chart details the key features for the app, and the relative importance of each feature (as decided by the client, from the use case). We only implemented core features - however without certain features the client may deem the App not useful. Below details the relative man hours / level of effort it took the team member to implement the feature.

### **Team Organisation**

- Week 4
  - **SPRINT 1**
  - Everybody implement their feature from the product backlog.
  - Arrange a second meeting with the client for the coming weeks. Everybody research unit & integration testing for their own feature.
- Week 5
  - **SPRINT 2**
  - Continue to work on implementing their feature from the last week, and from the product backlog. Look into unit testing.
- Week 6
  - **SPRINT 3**
  - Check that the essential features are almost finished for the App to prepare to show to the client at the end of the sprint for the initial evaluation. Start basic testing.
- Week 7
  - **SPRINT 4**
  - End of Sprint. Sprint retrospective. Merge feature branch to master branch and prepare to show the client for the pre-arranged meeting.
- Week 11
  - **Submit. Complete Documentation**

#### **Team Tenacity 8-Hobbytes = 1 Hobbytes**

Miriam Tamara G. Aarag - [1601261](#)

Julius Babenskaskas - [1612647](#)

Funke Sowole - [1617272](#)

Jeongho Suh - [1504262](#)

George Thornton - [1617550](#)

Kaitlin Villa-Fernandez - [1501227](#)

Yafan Yin - [1661173](#)



## VII. Other Information

| Risk Assessment |  |  |  |
|-----------------|--|--|--|
| No              | The Risk   | Risk Mitigation Strategy   | Resolved by end of project?  |
| 1               | A team member could 'accidentally' delete the GitHub repository .  | Ensure all team members regularly pull / clone the repository.   | <ul style="list-style-type: none"> <li>Yes, nobody deleted the repo.</li> </ul>  |
| 2               | The Client may change his requirements significantly, half way through the project                       | Ensure the client agrees this is what he wants before implementing difficult features.                                     | <ul style="list-style-type: none"> <li>Yes, we ensured that we understood the brief by identifying any ambiguous requirements before starting implementation.</li> </ul>   |
| 3               | Our client may be undecided on the Apps platform (i.e. Web or Mobile)                                    | Look into various technologies. Compare the advantages of Native vs Hybrid (e.g. Android Studio vs PhoneGap Cordova )      | <ul style="list-style-type: none"> <li>Yes, our skillset did not make programming in PhoneGap (which uses Javascript) a viable option as most team members do not have React or Javascript experience.</li> <li>Our team members have experience with Java &amp; iOS so creating 2 native apps is the most viable option.</li> </ul> |
| 4               | Our client may be busy with other work which may delay communication.                                    | Send a Weekly e-mail each Thursday to update the client on project progress.   | <ul style="list-style-type: none"> <li>This problem was resolved as we decided to spent the time using Server &amp; Mobile tutorials on Lynda.</li> <li>We also confirmed that we understand what he asked for when he responds.</li> </ul>  |
| 5               | We may not be able to communicate with the client as frequently as we would like to.                     | Ensure we prepare before each meeting and in each e-mail all questions that we have.                                       | <ul style="list-style-type: none"> <li>Yes, the client was able to attend every Skype Meeting and in person meeting apart from one because we ensured that we planned meetings with him in advance.</li> </ul>   |
| 6               | The client may have limited technical knowledge, so may propose ideas that are not technically feasible. | Do not agree to work that we are know that we are unable to complete.  | <ul style="list-style-type: none"> <li>Yes, we agreed with the client to simplify the requirements, but in a way that still made the App useful to the client.</li> <li>We also suggested features that would meet the client's requirements, but would be less difficult to implement.</li> </ul>                                   |
| 7               | The team may have other coursework to complete.  | Split the learning materials amongst team members so we can all share what we have learnt without unnecessary overlapping. | <ul style="list-style-type: none"> <li>Yes we managed our time properly as a group and prioritised important work.</li> <li>We also split work into particular features, so everyone mainly learnt work related to their feature.</li> </ul>   |
| 8               | One or more team members may have frequent absences from team meetings.                                  | Ensure that the team members communicate to each-other via the Kanban board.   | <ul style="list-style-type: none"> <li>Yes, we redistributed the work amongst the remainder of the team when this is the case.</li> </ul>  |
| 9               | We may use material we are not legally allowed to use (i.e. NHS logo) on the app.                        | Ask the client to send us un-copyrighted images, or images we have been approved us to use.                                | <ul style="list-style-type: none"> <li>Yes, because the client sent us un-copyrighted images.</li> </ul>   |

### Appendix

Firestore Credentials: [teamtenacity959@gmail.com](mailto:teamtenacity959@gmail.com) | teamtenacity999

Replicode: iOS Re-used code: <https://github.com/RobCanton/SwiftFirestoreTutorials>

#### Team Tenacity 8-Hobbytes = 1 Hobbytes

Miriam Tamara G. Aarag - [1601261](#)

Julius Babenskaskas - [1612647](#)

Funke Sowole - [1617272](#)

Jeongho Suh - [1504262](#)

George Thornton - [1617550](#)

Kaitlin Villa-Fernandez - [1501227](#)

Yafan Yin - [1661173](#)