Initial Report

by Project Prime

January 2019

1 Project Description

Project Prime consists of six members: Yusaf, Sandipan, Manny, Cameron, Shefali and Saloni. The project aim is to build a multi-host file synchroniser, which comprises of three components: a server application (the "hub"), a mobile and a desktop client (the "spokes"). Team communication will be conducted via WhatsApp and Skype, and in booked rooms within Kings' premises (e.g. group study rooms). All source code will be stored in GitHub to track the development progress of each component and to share the code base. Google Drive will contain documentation for file-editing/sharing purposes.

1.1 Project Aims

Aim: Develop a multi-host file synchroniser consisting of a server (the "hub"), and two clients: mobile and desktop (the "spokes"). The file synchroniser should be able to manage conflicts and contain files accessible to both the clients.

1. Develop a server application.
   (a) Define server requirements based-on research of existing file-synchroniser servers.
   (b) Build the backend using Node.js.
   (c) Must be able to connect with a MySQL database.
   (d) Must be able to communicate with the mobile and desktop clients simultaneously.
   (e) Must be able to store files.
   (f) Must be able to check for updates.
   (g) Must be able to authenticate requests.
2. Create the MySQL database and connect it to the Node.js server.
3. Develop two clients...
   - Develop the mobile client.

(a) Define the requirements of the mobile client based-on research of mobile apps.
(b) Design the UI of the mobile client via sketching.
(c) Implement the Mobile App using Android Studio.
  - Create the UI of the mobile client.
  - Build a nested directory for files.
  - Write code for uploading files.
  - Write code for creating a shared folder.
  - Write code for checking updates regularly (rsync algorithm).
  - Write code that handle conflicts.
  - Form a connection and set protocol to communicate with the database via the server.

(d) Black box testing, unit testing and integration testing.

- Develop the desktop client.
  (a) Define the requirements of the desktop client based-on research of web file-synchronisers.
  (b) Design the UI of the desktop client via sketching.
  (c) Implement a file management system using Electron.
    - Create the UI of the desktop client.
    - Build a nested directory for files.
    - Write a method for uploading files.
    - Write a method to create a shared folder.
    - Write a method for checking updates regularly (rsync algorithm).
    - Write a method that handle conflicts.
    - Form a connection and set protocol to communicate with the database via the server.

  (d) Black box testing, unit testing and integration testing.

1.2 Rough Timetable

Level 1:

1. Create and operate a server application using Node.js (by the 6th February)
2. Create the MySQL database and form connection between database and server (by the 6th February)
3. Desktop client...
   (a) Make sketches for the UI of the desktop client (by the 3rd February)
   (b) Implement the backend of the desktop client using Electron (by the end of February)

4. Mobile client...
    (a) Make sketches for the UI of the mobile client (by the 3rd February)
    (b) Implement the backend of the mobile client using Android Studio (by the end of February)

Level 2:

1. Implement an algorithm to handle conflicts (by the 1st week of March)
2. Implement the rsync algorithm to handle updates (by the 1st week of March)

Level 3:

1. Software testing of both clients and server application (by the 20th March)

1.3 Current Progress

- Desktop client
    1. Defined the requirements of the desktop client.
    2. Designed the UI of the desktop client via sketching.
- Mobile client
    1. Created different designs of the mobile UI on Android Studio.
    2. Started implementing the mobile client using Java and XML on Android Studio.
- Server application
    1. Researched about server features, including file transfer and encryption.

2 Project Organisation

2.1 Team Roles

The team is divided into three subgroups of two members and each subgroup focuses on one component.

- Yusaf and Sandipan = This subgroup will design, implement and test the desktop client. First, the client will be designed based-on research of similar file-synchronising web applications (e.g. Google Drive). Next, it will be created using Electron and Node.js. Afterwards, the desktop client will undergo black-box, unit and integration testing.

- Manny and Cameron = This subgroup will design, implement and test the mobile client . First, the client will be designed based-on research of file-synchronising mobile apps (e.g. Dropbox). Next, the client will be implemented using using Android Studio (which includes Java and XML) and Node.js. Afterwards, the mobile client will undergo black-box, unit and integration testing.

- Shefali and Saloni = This subgroup will design, implement and test the server application using Node.js. First, the server will be designed based-on research of existing file-synchronising servers. Next, it will be implemented using Node.js and finally, it will undergo interface testing to determine if the clients and servers have correct interaction between each other.

2.2 Peer Assessment

In the peer assessment, Project Prime will possess 100 points in total and will allocate a certain amount of points to each member, where the allocation (including integers and decimals) is based-on the member's individual contribution. Furthermore, the peer assessment will be conducted in a booked room within the university premises. On its process, all non-assessed members will debate how many points the assessed member should be assigned with, and upon agreement, the assessed member will be assigned with those points. However, if a disagreement happens, the debate continues until a consensus is met. The final member will receive the remaining amount of points to equal the total amount of points (i.e. 100).

2.3 Handling Team Conflicts

There are two types of team conflict that can occur during the project, which are group disagreements and arguments between two individuals; for both types, we will use the FUSION method (Robin, 2013) which involves the following key points: making an effort to understand their perspectives, issues and expectations; being specific and concrete about what we want; being intentional about the questions we ask to ensure there is a quality dialogue, showing that we are open to change and options about accomplishing the work, don't let "hot button" language like "you always" and "you never" enter the conversation.

3 References

- Robin (2013) *FUSION: A Six Step Solution to Handling Conflict Across Generations* [online] available at: https://strategichrinc.com/article/fusion-a-six-step-solution-to-handling-conflict-across-generations/ [Accessed on 29 January 2019]