



Faculty of Natural & Mathematical Sciences

7CCSMGPR – Group Project

Assignment 1 of 1 – Initial Report (Default Team)

Authors

Alain DOEBELI
Christakis VASILIOU
Ignacio DOMINGUEZ GARCIA-GUIJAS
James KAYONGO
Konstantinos OIKONOMOU
Nikolaos RIZOS

Module Tutor

Dr. Laurence TRATT

February 5, 2019

1 Project Description

1.1 Objective Description & Levels

Our first goal was to understand and capture the requirements of the project that we were about to undertake. We concurred that in order to satisfy all the requirements of the project, the presence of a server, a desktop and a mobile application were needed, so we decided to delegate the objectives to each of the aforementioned subcategories. Furthermore, in order to aid the correct scheduling of the work, we decided to color-code the objectives by order of priority, as seen at Table 1. Green objectives represent the bare minimum functionality of the corresponding component and should be completed first. Yellow objectives represent important functions of the subsystems and should be done immediately after. On the contrary, red objectives represent features that are not essential for the system to be operational, but could potentially be added to expand the services it provides. Finally, blue objectives are quality of life improvements that will be implemented only if the quality of the system has been thoroughly tested and time allows for their development.

| Objectives | |
|--|--|
| Server | Desktop/Mobile |
| Set up Node.js app | Set up Electron and Android apps |
| Receive & save a file through HTTP | Create the GUI |
| Check for conflicts of a file being uploaded | Show files in a local dir |
| Send file information through HTTP | Request file information from server |
| Send latest file version through HTTP | Push file changes to server |
| Deal with possible conflicts | Pull outdated files from server |
| Implement different users | GUI and functionality for multiple users |
| Implement permissions amongst users | Allow users to give permissions |

Table 1: Color-Coded Objectives

1.2 Proposed Approach

Our approach is to create a server with its database, a desktop application and a mobile application and connect them as shown in Figure 1. To set up the server we plan to utilize the framework Express.js and therefore Node.js in general. One of Node.js main advantages is that by default it queues the inputs and that can help us deal with potential file conflicts from multiple users requests in the same time period. Furthermore, our client application

will use Node.js as well, so another obvious advantage is the easier connectivity between server and client. Finally, another major advantage is that our database will be implemented with MongoDB. MongoDB is a NoSQL database and Node.js nullifies the need to modulate syntax differences during the server-database communication. We plan to build our desktop application utilizing the framework Electron because it can produce cross-platform applications and moreover uses Chromium. This results in easier creation of the application and GUI using HTML, CSS and JavaScript. The proposed approach is concluded with the mobile application, which will be developed as an Android-native application in Java using Android Studio. Both the desktop and the mobile app will have the same functionality regarding their communication with the server.

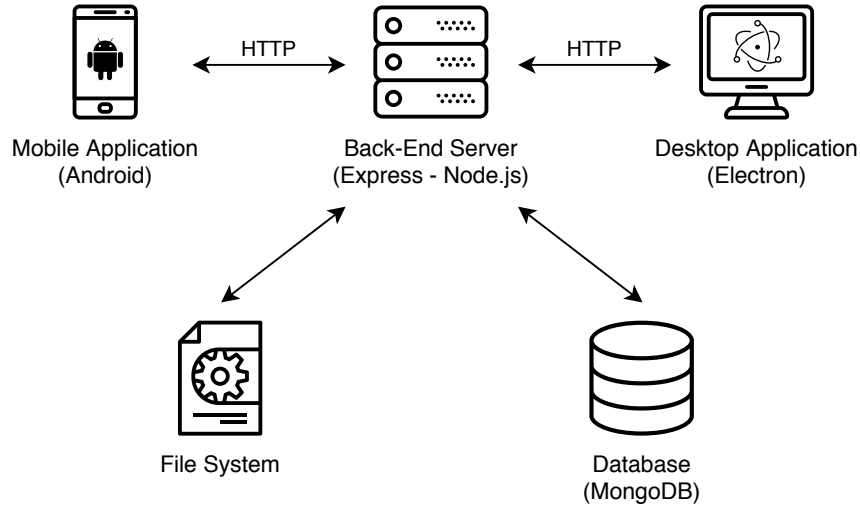


Figure 1: Project Diagram

1.3 Timetable & Initial Progress

After the first week meeting, which was dedicated to form a plan and develop our approach, we decided on scheduling the work in 2-week Sprints, using the Scrum methodology, explained in more detail in Chapter 2.1. The full schedule can be seen in Table 2 and includes all the objectives, time for testing and preparing the final report and presentation, as well as some contingency allowance in case our plans are inaccurate to some extent. As of right now, we have completed all the objectives of Sprint1 and the majority

of the Yellow objectives, so we are ahead of schedule.

| Timetable | |
|------------------|--|
| Date | Description |
| 17/1 - 23/1 | Planning and Initial Task Delegation |
| 24/1 - 6/2 | Sprint1: Green + Yellow(As many as possible) |
| 7/2 - 21/2 | Sprint2: Remaining Yellow + Red(optional) |
| 22/2 - 6/3 | Sprint3: Remaining Yellow + Red(optional) |
| 7/3 - 27/3 | Testing, Final Report, Presentation, Contingency allowance |

Table 2: Project Timetable

2 Project Organisation

2.1 Project Methodology

We decided to follow an agile approach for our project, as it is a natural fit for software development and it provides working iterations of the final product in every cycle. More specifically, we opted for the Scrum methodology but with some necessary adjustments, in order to fit our project and timetable. For example, we opted for 2-week Sprints rather the more traditional 1-month Sprints, due to time constraints and decided not to have a Scrum master. Finally, after the end of each Sprint we will have a Sprint review meeting to decide goals for the next Sprint and reflect on our performance. For the initial Sprint we split in groups of 2 and each group will work on the server, the desktop app and the mobile app respectively. The goal is that every team member has worked on every part of the project by its end.

2.2 Peer Assessment & Conflict Handling

Our plan regarding peer assessment is to split the available points evenly amongst all group members. If a team member fails to deliver on time or refuses to communicate with the rest of the team a team meeting will decide if points will be deducted from him, but our approach is to give everyone at first the benefit of doubt. If any conflicts arise, a team meeting will be called and the reasons for the conflict will be discussed. Any decisions will be taken by vote if necessary but early signs suggest that constructive discussion is often enough to resolve any issues.

References

- [1] *Express.js*. <https://expressjs.com/>. [Online; accessed 18-January-2019].
- [2] Node.js Foundation. *Node.js*. <https://nodejs.org/en/>. [Online; accessed 18-January-2019].
- [3] MongoDB Inc. *MongoDB*. <https://www.mongodb.com/>. [Online; accessed 20-January-2019].
- [4] GitHub. *ElectronJs*. <https://electronjs.org/>. [Online; accessed 18-January-2019].
- [5] Ganesh Iyer. “Node.js: Event-driven Concurrency for Web Applications”. In: (Jan. 2013). DOI: 10.13140/RG.2.1.2591.9849.
- [6] L. Rising and N. S. Janoff. “The Scrum software development process for small teams”. In: *IEEE Software* 17.4 (2000), pp. 26–32. ISSN: 0740-7459. DOI: 10.1109/52.854065.