

# CA400 Functional Specification

*Project Title:* Echo Labs

*Group:* Fiona Tuite: 16389251

Munashe Lawrence Chifungu: 16469764

*Document End date:* 03/12/2020

---

## Table of Contents

<b>1. Introduction .....</b>	<b>2</b>
1.1. Overview .....	2
1.2. Business context .....	2
1.3. Glossary .....	2
<b>2. General Description .....</b>	<b>2</b>
2.1 Product / System Functions .....	2
2.2. User Characteristics and Objectives .....	3
2.3. Operational Scenarios .....	3
2.4. Constraints .....	4
<b>3. Functional Requirements .....</b>	<b>4</b>
3.1. User Login/ Registration .....	4
3.2. User Dashboard .....	4
3.3. WebRTC (Video and Voice Calls) .....	5
3.4. Web Chat .....	5
3.5 Add on features .....	6
<b>4. System Architecture .....</b>	<b>6</b>
<b>5. High-Level Design .....</b>	<b>7</b>
5.1. Sequence Diagram: User login .....	7
5.2. Sequence Diagram: Lecturer creates Lab Room .....	8
5.3 Sequence Diagram: Student enters a Lab Room .....	8
5.4 Sequence Diagram: Student requests help .....	9
5.5 Use Case Diagram .....	9
<b>6. Preliminary Schedule .....</b>	<b>10</b>
<b>7. Appendices .....</b>	<b>Error! Bookmark not defined.</b>

# 1. Introduction

## 1.1. Overview

Before the Covid-19 pandemic hit the world, our labs in DCU would be held in a large class of students, with tutors and lecturers giving face to face assistance to those in need. Once Covid-19 came along, A lab in that before setting was not do-able. This is how we came up with the idea of Echo Labs.

The Echo Labs web application is specifically focused on Peer-To-Peer audio, video and chat communication where the goal is to simply connect students, tutors and lecturers together in a lab like scenario. We named the application Echo because its focus is on establishing more effective back and forth communication between students, lecturers and tutors alike. To ensure an opportunity for everyone's voice to be heard. This application will be mainly built using the React web framework accompanied by the WebRTC API in order to develop the audio, video and chat system within this application.

## 1.2. Business context

At this point in time, to adhere to social distancing guidelines most of the communication between students and faculty is being done through SaaS tools, in particular Zoom. While this has been efficient for the theoretical aspects of modules it has not been the best for practical work. Programming, in particular for newer students can be intimidating and have a fairly steep learning curve, of course it's possible to learn at home but the benefit of being able to attend labs and request help from tutors, classmates and lecturers cannot be overlooked. The goal of Echo Labs is to provide a SaaS tool designed with a focus on features that enrich the learning experience from home.

## 1.3. Glossary

- *Firebase*: Application from Google that supports the development of IOS, Android and Web Applications.
- *WebRTC*: Web Real Time Communication allows apps and sites to capture or stream audio/video media between browsers without an intermediary device.

# 2. General Description

## 2.1 Product / System Functions

Below is a list of general functions that we would like to include into our web application that we will go into detail as the document continues:

- User Login
- User profile & Dashboard
- Lab Room
- Ability to request help from a lecturer/tutor
- Audio, video or chat a lecturer/tutor
- Native peer-to-peer code editor
- Insights

## **2.2. User Characteristics and Objectives**

Our system is geared towards students which casts a wide net for our demographic. We will be focusing on DCU Computer Science students, those in their first or second year of studies where lab work is most vital. As such we expect the users to at least be moderately tech savvy however advanced knowledge of computers is not necessary. Our goal is to implement a minimal but effective UX that works exactly as described, creating an environment for students and faculty to communicate efficiently.

## **2.3. Operational Scenarios**

### **Sign Up**

Upon entering the web application, the user will be prompted to login or register using their email address. Since we will be focusing on DCU students for the purpose of this project we'll be implementing an auth feature allowing students to login with their DCU allocated email address through Gmail.

### **Create Lab Session**

Once logged in, faculty members will be able to create a lab session. From the nav bar, faculty can select a "create lab" option. Here they can select students to add to the lab session they are creating. Selected students will then have this lab session appear on their dashboard when they log into their page and will thereafter be able to join this session.

### **Entering a lab room**

Once logged, students will have a view of their labs and whether the lab sessions are active or not. If a session is active students will be able to click on the active session which will take them to the lab where they will have a view of the chat and the call that is being held

### **Requesting help**

When a lab is in progress, students will be able to request assistance from tutors or the present lecturer. From the UI showing who is active a student can select a lecturer who is showing as available and send a request for assistance. From here it is up to the tutor/lecturer to accept the request.

## Accepting a help request

When a help request is sent from a student the tutor/lecturer will receive a notification. If the lecturer/tutor accepts the request, then they will have the option to begin a call with the student or start a chat with them in a private session. From here the lecturer/tutor will have the ability to view or edit code with the student or share their screen.

## 2.4. Constraints

**Time:** We expect time to be the biggest constraint factor during this project. In theory this project is feature rich and each feature has various components of various complexity this means executing these features well in a timely manner will be our most formidable challenge.

**Working from Home:** We'll be developing this project entirely from home. Without the benefit of being able to focus on working in the college and the multitudes of distractions that can occur at home there may be limitations to how productive we can be in a day. As well as this our testing process could be affected by the fact that we cannot interact with others as a result of the current situation.

**Firebase:** We are using a free firebase account so there is a limit on our query count, storage and connections to the app

**Learning Curve:** As this is our first time using some of the services there is the added difficulty of having to learn as we go.

## 3. Functional Requirements

### 3.1. User Login/ Registration

#### Description

Students and faculty should be able to log into the Echo Labs application as this is the main prerequisite to making use of the services. Using Auth we can allow users to register using their DCU provided Gmail accounts. This will verify them as a student or faculty member of DCU and grant them access to the service.

#### Criticality

Without an account there is no way to make use of Echo Labs services, so this is one of the most crucial features for the application.

#### Technical Issues

We will be making use of the Firebase Authentication SDK to create our preferred login/registration UX. This is something we will be learning to do.

#### Dependencies

User Authentication is dependent on the Firebase Authentication SDK

### 3.2. User Dashboard

#### Description

Once logged in the user will find themselves on their dashboard. This is where they will find access to the tools and services that Echo Labs. For students this is where they will find their lab sessions and native code editor. As well as this they will have a view of their chat and WebRTC services. In addition to what the students can see on the dashboard, faculty will also have access to insights and the ability to create lab sessions and add students to them.

#### Criticality

This is the primary control centre for Echo Labs services from the user perspective. The UX here should be succinct, understandable and efficient and therefore will be one of the areas we focus on the most.

#### Technical Issues

Combining the services Echo Labs into a single UI that provides a satisfactory UX will be a challenge. In particular the integration of our WebRTC and chat implementations.

#### Dependencies

The dashboard is a major front-end feature, and its effectiveness is dependent on how well constructed the backend features like WebRTC are.

### **3.3. WebRTC (Video and Voice Calls)**

#### Description

Lab sessions will be hosted primarily using WebRTC. With WebRTC we can create a real time communication system between students and faculty. This will consist of both VoIP and video communication.

#### Criticality

As this is the primary method of communication especially within lab sessions for faculty and students this is a critical feature

#### Technical Issues

Creating a Lab session where everyone can communicate with each other at once and have this formatted correctly via the front end may lead to some technical issues.

#### Dependencies

WebRTC communication will be managed using firebase hosting and firebase cloud firestore. The video call interface will also be integrated with the front end and lab sessions feature.

### **3.4. Web Chat**

#### Description

Other than creating lab sessions using WebRTC, a chat system will be vital for communication without having to make a call. Using chat students will be able to ask questions to each other and the hosting faculty member.

### Criticality

This is important as allowing communication and interaction is a key element of this application.

### Technical Issues

Implementing a client server system that hosts the web chat and WebRTC service for any number of attendees could be technically challenging. In addition, we must consider the constraints involved in using a free firebase account

### Dependencies

Webchat is dependent on the client server system as well as the WebRTC services.

## 3.5 Add on features

### Insights

Insights will be a view of data collected throughout the application. This data includes the duration of lab sessions, user activity and participation and metrics concerning how engaging users found the application to be

### Code Editor

A native code editor, useful for coding labs where students can write and test their code. From here tutors and lecturers can also view how a student is doing with their given assignment and help them where necessary.

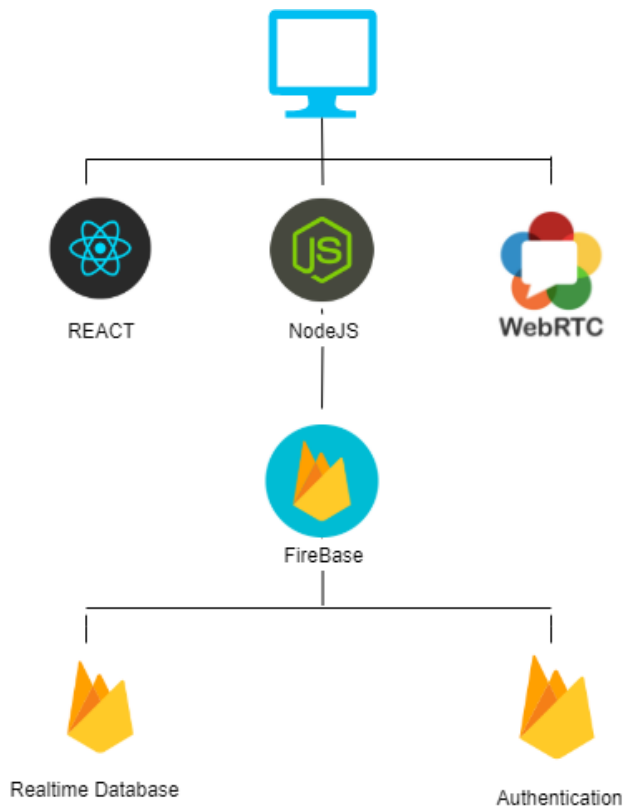
### Screen Share

The ability for faculty and students to stream their desktops/workspaces to each other for the purpose of presentations or editing code together.

## 4. System Architecture

Below, we have an overview of the system architecture that displays the different layers and its relationships.

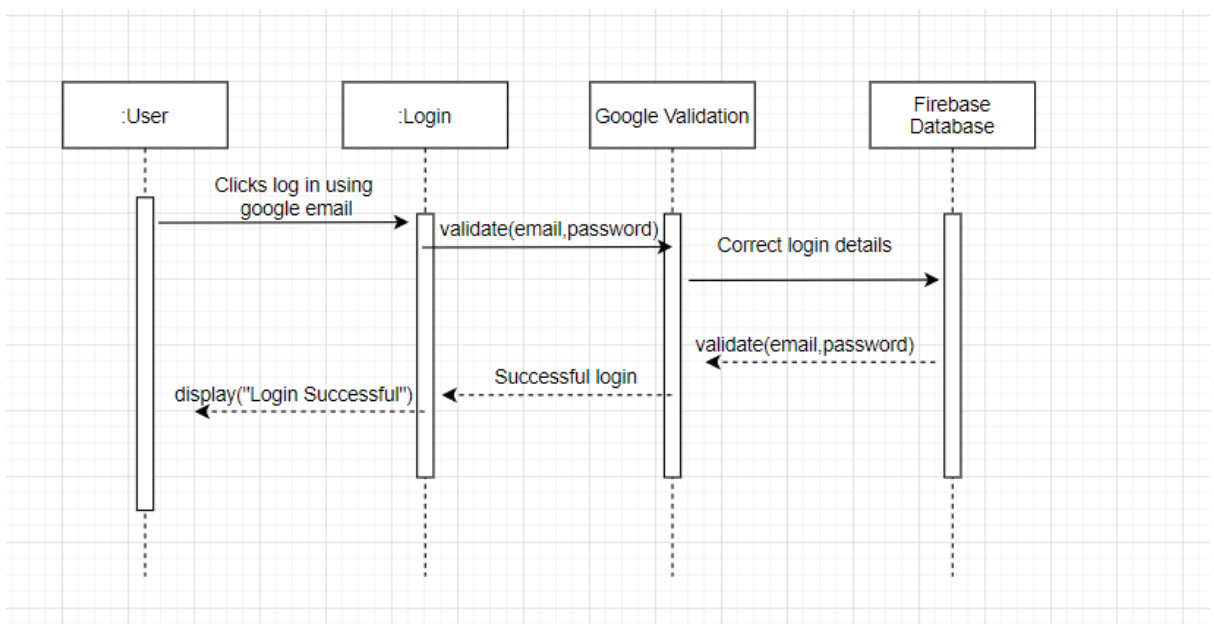
The top tier of the architecture represents the front end of the application which is known as the user interface. This will display useful content to the end user. The second tier represents the application programs in which the diagram below shows the applications: ReactJS, NodeJS and WebRTC. The Last tier represents the data components. Firebase will be used for hosting Echo Labs as well as this. We will use Firebase features such as its authentication to allow user access, and its real time database in order to store user information.



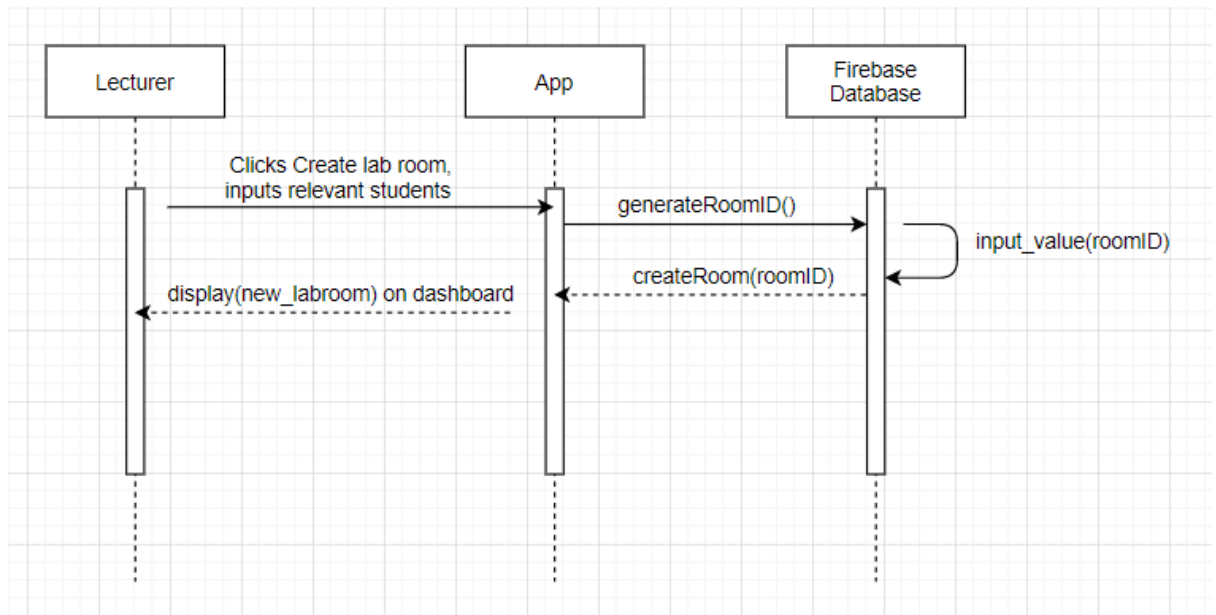
## 5. High-Level Design

Below are different diagrams that portray the function of our application at a high-level.

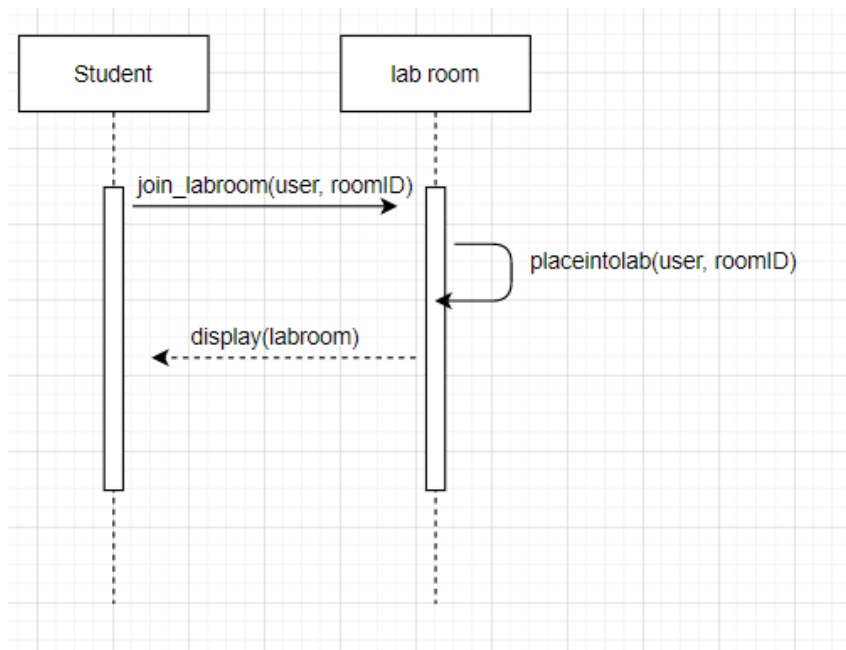
### 5.1. Sequence Diagram: User login



## 5.2. Sequence Diagram: Lecturer creates Lab Room

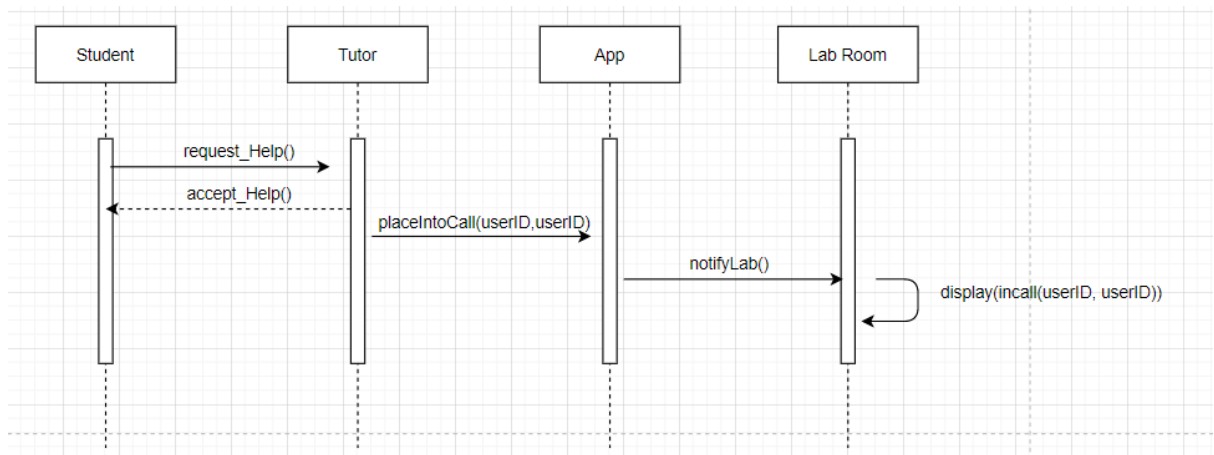


## 5.3 Sequence Diagram: Student enters a Lab Room

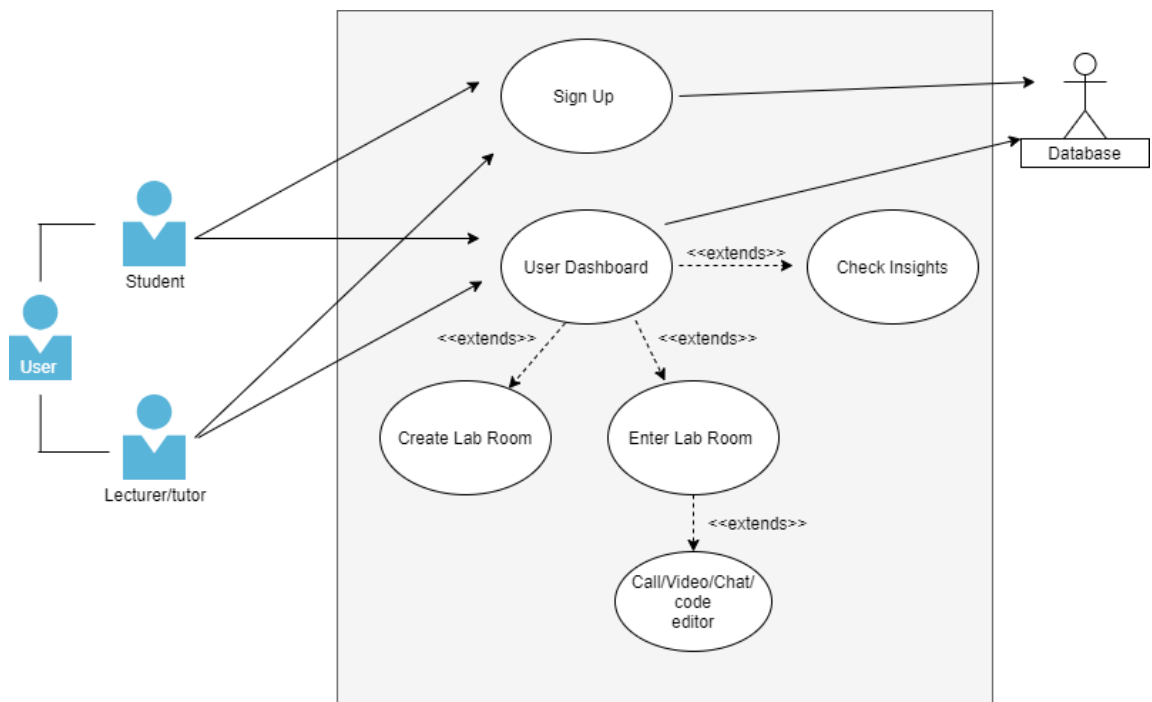




## 5.4 Sequence Diagram: Student requests help



## 5.5 Use Case Diagram



## 6. Preliminary Schedule

Below, is a Gantt chart that portrays our final year projects preliminary schedule. Our goal is to keep in regular contact with each other as well as our supervisor in order to stay on track. This will be done by working together as a team and having regular discord sessions.

## GANTT Chart

