Lab 2 – Inclusive Classroom Product Specification

Sergiy Yermak

CS 411w

Thomas Kennedy

October 29, 2021

Version 1

Table of Contents

Table of Figures

1. Introduction

Inclusive Classroom (IC) is a computer application conceived in spring of 2021 by an Old Dominion University student named Randy Layne for a Professional Workforce Development class. During the spring semester the idea has been further developed through presentations on its feasibility, design, and prototype. The development has been done by six students: Travis Bennett, Dalton Hanbury, Greg Hubbard, Colton Hurst, Grant Ralls, and Randy Layne. In fall of 2021 the second part of the development began bringing with it two new group members, Aubrie Davie and Sergiy Yermak. This lab 2 report is part of that second semester of development. The predecessor of lab 2, lab 1, discussed the general product description of IC. Whereas this lab will dive into the product specifications. The first section of this paper will discuss the purpose behind building the application, describe the scope of the development, and provide definitions, acronyms, abbreviations, and references. Section 2 will provide a general description for the prototype, which includes architectural and functional descriptions, as well as extended interfaces. Although section 3 is part of lab 2, it will not be included in this document. Section 3 will provide a list of specific requirements that the prototype will meet.

1.1. Purpose

Due to a recent pandemic, schools, specifically those below college level, had to rapidly adapt to a new style of teaching for their students. Digital learning has been growing since 2002 when No Child Left Behind Act was passed and increased the pressure on schools to improve graduation rates (Heinrich). However, online learning has proven to have its own share of challenges. First such challenge is the lack of technical expertise across all parties involved. Blackboard is a leading provider of online learning and serves over 20 million K-12 students nationwide (Blackboard.com). However, on the first day of school in 2020, Blackboard reported a list of problems with its service, which included: failing to load, loading slowly, and unable to register new users (Lozano). This event forced many of the students, parents, and teachers to act as tech support to get the classes started, leaving some falling behind on the first day of school. Second problem with online learning is the lack of stable high-speed internet access at home, especially for lower income families. According to Pew Research Center, from among parents with children whose schools were closed and report to have lower income; 43% said they had to do their schoolwork on a cellphone, 40% reported to have used public Wi-Fi to finish their schoolwork, and 36% reported that they were unable to finish their schoolwork due to lack of internet access at home (Vogels).

The purpose behind the call for a new solution is the continued failure of online learning platforms, which bring the risk of deterioration of children's education. Both figures below [Fig. 1, Fig. 2] show how the current processes fail students when there is a lack of stable high-speed internet available.

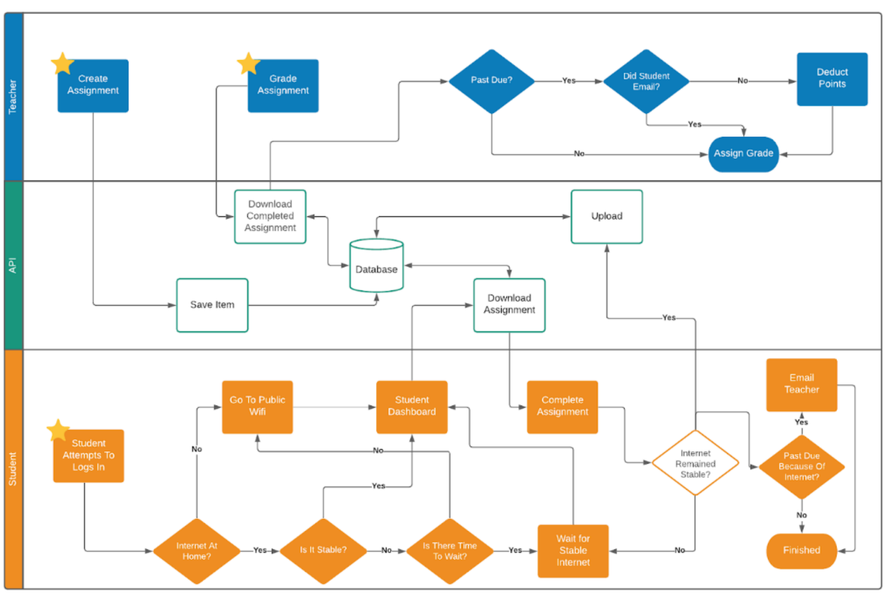

Figure 1 Current Assignment Process Flow

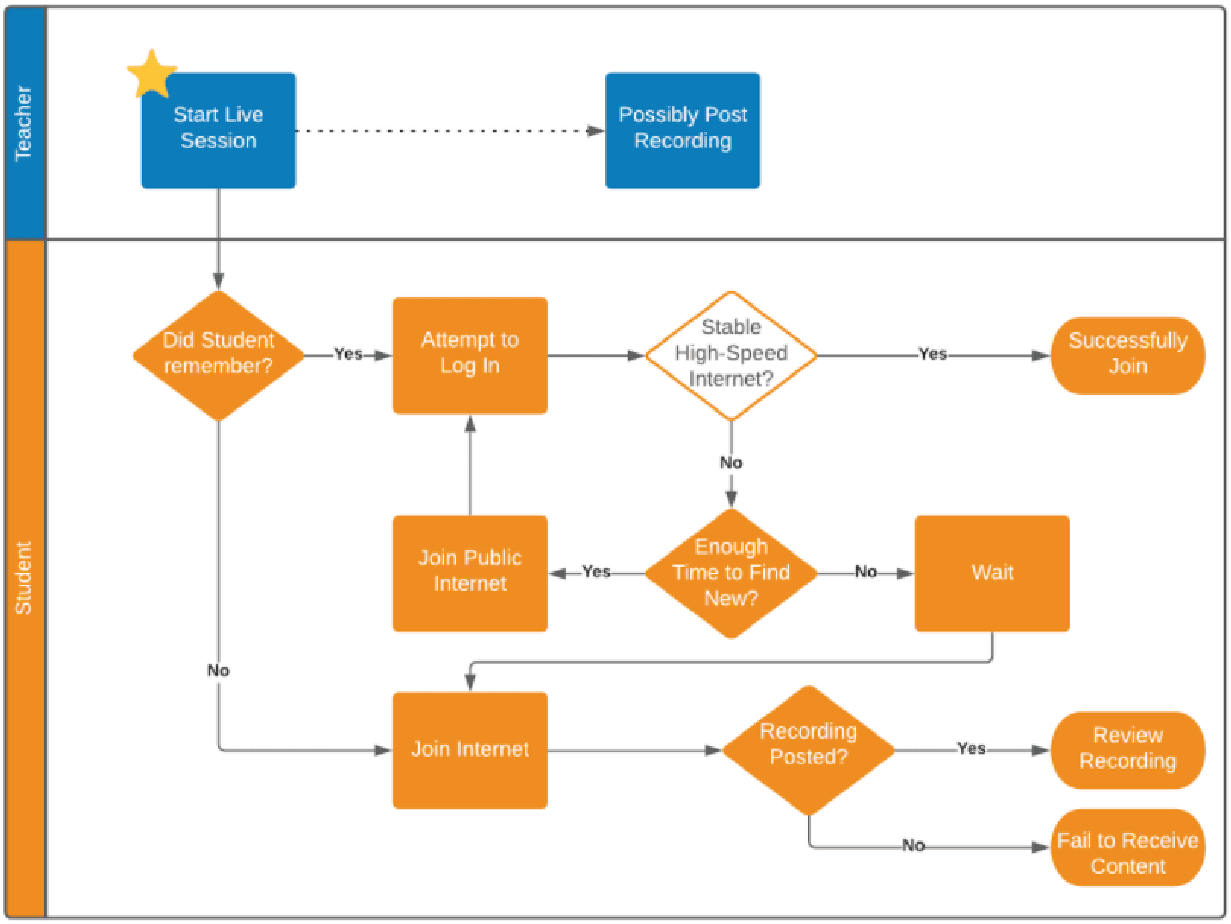

Figure 2 Current Live Video Process Flow

1.2. Scope

It is crucial that the scope, defined as the functions and features of the product, falls correctly within the limits. The primary limits of this prototype are team software development experience and time constraints. Only about a half of the group has real world software development experience, the rest are fresh computer science students who at this point only had a single software development class. Furthermore, the problem is exaggerated by the lack of knowledge using certain tools, such as the Reach JavaScript library used for building user interfaces. The time constraint of this project is short, about 3 months of work in total. Although that is enough time to build a prototype, other factors such as other responsibilities/classes, learning new tools, and writing paperwork for the prototype subtract from the already limited time allotted. With the limits in mind, a few real-world features had to be either not included or limited in the scope of the prototype. In a typical online school environment, there are many roles that fulfil required tasks. However, it would be too time consuming to prototype each role. Teacher and students are the two most essential roles so they will be prototyped. Although an administrator is important, that role will have limited functionality. While an Information Technology specialist will outright not be included. Assignments come in a wide range of types which will be time consuming to implement, therefore only a limited type such as typed out answers will be included in the prototype. Any type of reporting and postdate created content will not be included. To see a full list what will be included, refer to [Fig. 7] in section 2.2.

1.3. Definitions, Acronyms, and Abbreviations

1.2.1 **High-speed Internet** - Internet with consistent download speeds of at least 3.8 Mbps (Zoom)

1.2.2 **English as a Second Language** (ESL) – English as taught to people whose main language is not English.

1.2.3 **Family Educational Rights and Privacy Act** (FERPA) - Federal law that protects the privacy of student education records

1.2.4 **Google Classroom** - “Free web service developed by Google for schools that aims to simplify creating, distributing, and grading assignments” (Google)

1.2.5 **littleLearners** - Former CS 410 group solution that emphasizes simple UI for students in the K-5 age range (Del Razo)

1.2.6 **Stable Internet** - Internet with less than 1% dropped packets (ICTP)

1.2.7 **UI** – The User Interface (UI) is he point at which human users interact with a computer, website, or an application.

1.2.8 **IC** – Inclusive Classroom

1.4 References

Heinrich CJ, Darling-Aduana J, Good A, Cheng H (Emily). "A Look Inside Online Educational Settings in High School: Promise and Pitfalls for Improving Educational Opportunities and Outcomes." American Educational Research Journal. 2019;56(6):2147-2188. doi:10.3102/0002831219838776

Lozano, J., Licon, A. G., & Boone, R. (2020, September 9). *Technical problems disrupted remote classes throughout nation*. Pittsburgh Post-Gazette. Retrieved October 19, 2021, from https://www.post-gazette.com/news/education/2020/09/08/schools-reopening-computer-glitches-technical-problems-online-learning-classes/stories/202009090090.

Anderson, Monica, and Andrew Perrin. "Nearly One-in-Five Teens Can't Always Finish Their Homework Because of the Digital Divide." Pew Research Center, Pew Research Center, 30 May 2020, www.pewresearch.org/fact-tank/2018/10/26/nearly-one-in-five-teens-cant-always-finish-their-homework-because-of-the-digital-divide/.

"Children's Online Privacy Protection Rule ('COPPA')." Federal Trade Commission, 1 Dec. 2020,www.ftc.gov/enforcement/rules/rulemaking-regulatory-reform-proceedings/childrens-online-privacy-protection-rule.

"Chromebook Support." LCS, www.lcsedu.net/departments/information-technology/chromebook-support.

"Classroom FAQ - Classroom Help." Google, Google, support.google.com/edu/classroom/answer/6025224?hl=en&ref_topic=7175444.

Del Razo, Gabriel, et al. "LittleLEARNERS." Team Orange, www.cs.odu.edu/~cpi/old/410/orangf20/.

ICTP Science Dissemination Unit, ICTP Science Dissemination. "ICTP-SDU Home Page." ICTP-SDU: about PingER, web.archive.org/web/20131010010244/sdu.ictp.it/pinger/pinger.html.

Kamenetz, Anya, and Eda Uzunlar. "NPR/Ipsos Poll: Nearly One-Third Of Parents May Stick With Remote Learning." NPR, NPR, 5 Mar. 2021, www.npr.org/2021/03/05/973373489/npr-ipsos-poll-nearly-one-third-of-parents-may-stick-with-remote-learning.

Raphael, JR. "Android Apps for Chromebooks: The Essentials." Computerworld, Computerworld, 19 Feb. 2019, www.computerworld.com/article/3234533/android-apps-for-chromebooks-the-essentials.html.

Section 504 & Students with Disabilities." Washington Office of Superintendent of Public Instruction, 2021, k12.wa.us/policy-funding/equity-and-civil-rights/information-families-civil-rights-washington-schools/section-504-students-disabilities.

"System Requirements for Windows, MacOS, and Linux." Zoom Help Center, support.zoom.us/hc/en-us/articles/201362023-System-Requirements-for-PC-Mac-and-Linux.

Team Gold. (2021, October 19). Lab 1 – Inclusive Classroom Product Description. Retrieved October 27, 2021 from https://inclusive-classroom.github.io/website/labs.html.

"The 504 Plan." The Center for Children with Special Needs, 2018, cshcn.org/childcare-schools-community/the-504-plan.

VBCPS. "VBCPS Adds 19,000 Chromebooks to Achieve 1:1." Virginia Beach City Public Schools, www.vbschools.com/news/archived_news/2019/chromebooks.

VDH. "COVID 19 Cases In Virginia." Virginia Department of Health., www.vdh.virginia.gov/coronavirus/coronavirus/covid-19-in-virginia-cases/. Accessed 20 Feb 2021

Vogels, Emily A. "59% Of U.S. Parents with Lower Incomes Say Their Child May Face Digital Obstacles in Schoolwork." Pew Research Center, Pew Research Center, 10 Sept. 2020, www.pewresearch.org/fact-tank/2020/09/10/59-of-u-s-parents-with-lower-incomes-say-their-child-may-face-digital-obstacles-in-schoolwork/.

"Web Applications with Spring." Spring, spring.io/web-applications.

Blackboard. (n.d.). Retrieved October 26, 2021, from https://cuchd.blackboard.com/.

1.5 Overview

The rest of this paper will discuss the following topics. General description is a simple paragraph describing what Inclusive Classroom will be. The prototype architecture section will go into details about what software and hardware will be used to build the application prototype. Prototype functional description will detail what features will be included in the prototype. Then finally the external interface section will discuss any outside hardware and software that will need to be used alongside of Inclusive Classroom.

2. General Description

As stated in the purpose, the reason a new application needs to be made is that the currently available providers of online learning have flaws that negatively affect students' education. The solution to these flaws is a student, teacher, and administrator facing application called Inclusive Classroom (IC). IC fixes the issue of technical inexperience by designing a simple UI that is easy to use for children. The teacher interface will be designed with convention over configuration in mind to keep things simple for the student. Furthermore, the software will have a high level of automation for such operations as uploading and downloading files, zipping and unzipping files (Gold Team Lab 1). A critical feature of IC is that it will be a native application. This means that the application will be able to run without always needing internet access. The application will be able to upload and download the information students need when the internet is available but working on assignments and viewing lectures will not require the need of a stable connection. The solution process flows [Fig. 3, Fig. 4] solves the current problems and allows the students to work.

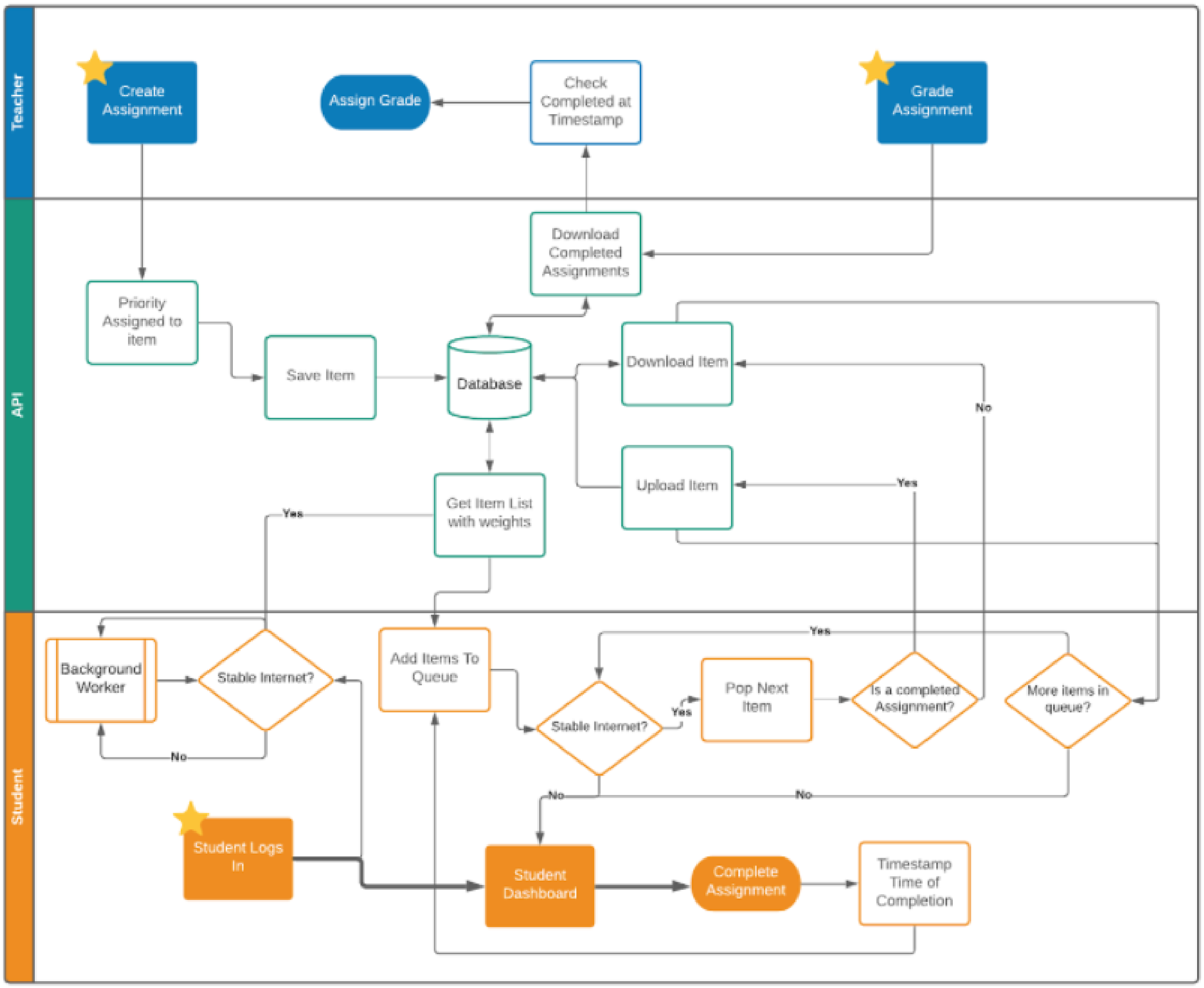

Figure 3 Solution Process Flow for Assignments

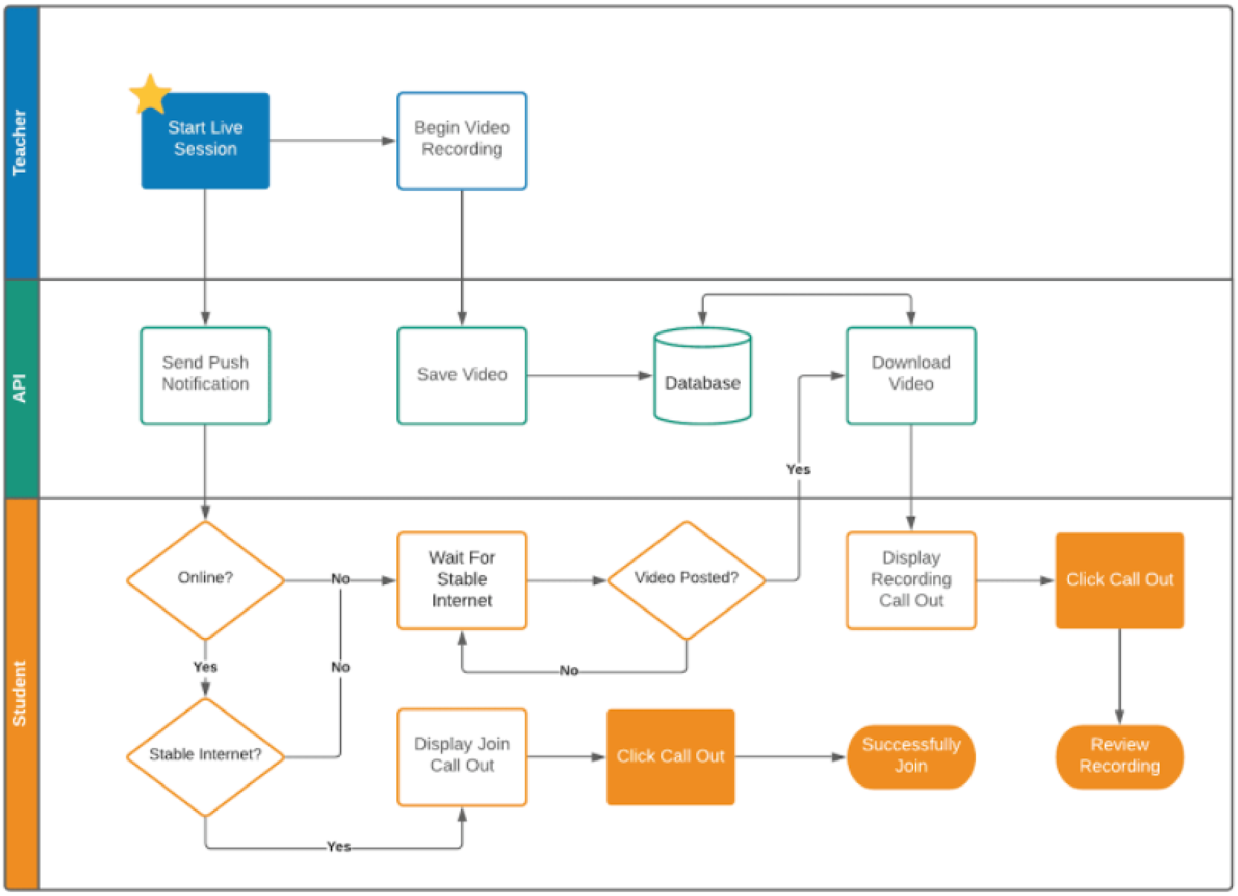

Figure 4 Solution Process Flow for Live Video

2.1 Prototype Architecture Description

The architecture for the prototype of Inclusive Classroom will be similar as the one used in the real-world product. It will use JavaScript as its primary language. The client side used by teachers, students, and the administrator will be built using React Native. React Native is an open-source UI software framework created by Facebook. SQLite, a relational database management system, will be used to store information such as live stream recordings, assignments, classes, teachers, and students. In the real-world product the client will access the data stored by the Redis server. Redis is on open source, in-memory data structure store, used as a database cache, and message broker [Fig. 5]. However, for the prototype, the client-side app will directly connect to the server side. The server side will use an asynchronous event-driven JavaScript runtime engine called Node.js will be used to execute the application. All the major Node.js apps rely on Express.js framework therefore it will be included in the project. Furthermore, Sequilize is a promise based Node.js ORM for SQLite and will be used to map JavaScript objects to the database. The server-side information will be stored in PostgresSQL. For a better understanding refer to the Major Functional Component Diagram below [Fig. 6].

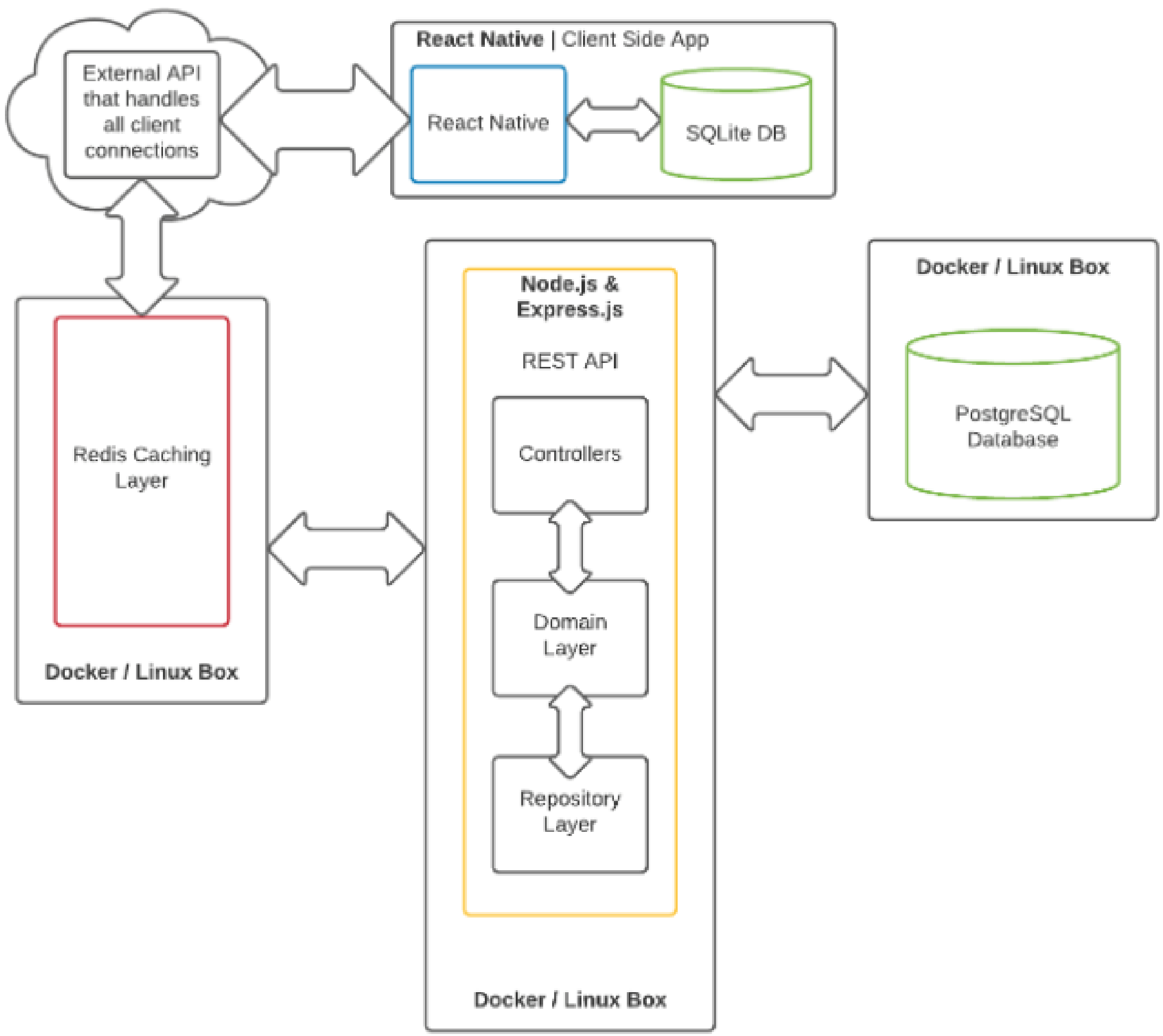

Figure 5 Real World Product Major Functional Component Diagram

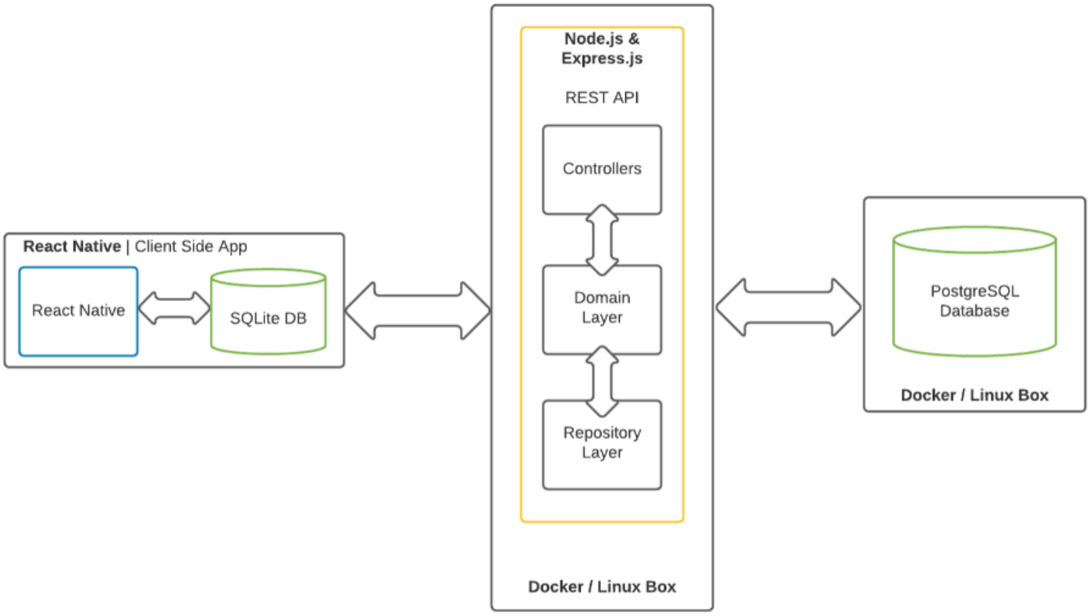

Figure 6 Prototype Major Functional Component Diagram

2.2 Prototype Functional Description

The prototype of Inclusive Classroom will include most of the core features of the real-world product. Account roles span 3 major parties, which includes: students, teachers, and school administrators. All three roles will be able to create an account with Inclusive Classroom and access a role specific dashboard. Students will have access to join live video sessions, view recorded videos, get feedback on graded assignments, view class details, and edit their profiles. Teachers will be able to create, view, and grade assignments, record videos, and view their class details which includes students, assignments, and material. Administrator role will have the ability to create new classes as well as assign a teacher and students to the newly created class. Furthermore, the administrator will have access to both teacher and student information such as: the total number of students assigned to a teacher, all classes assigned to a teacher, student name, classes a student is enrolled in, and over class grades of a student. To ease the lack of technical expertise, the application will emphasize automation. The prototype will have the ability to automatically detect internet access so it may synchronize with the backend server. Provided that, the prototype will have a weighted priority queue for downloading assignments once an internet connection is detected. Likewise, the application will also automatically upload submissions of assignments. To make sure that the students are submitting their work on time, the prototype will timestamp completed assignments. For an accurate list of features that will be included in the real-world product and in the prototype refer to [Fig. 7] below. For an accurate description of all the functions that will be included in the prototype refer to section 3.

Feature	RWP	Prototype
Account Roles	Student, Parent, Teacher, Admin, IT	Student, Teacher, Administrator(limited) only
Automatic Internet Detection	Yes	Yes
Background Workers	Yes	Partial
Complete/Submit Assignments	Yes	Yes
Create Assignments	Yes	Yes, limited in types
Grade Assignments	Yes	Yes
Postdate created content	Yes	No
Recorded Videos	Yes	Yes, may not auto record
Reporting	Yes	No
Timestamp Completed Assignments	Yes	Yes
Video Conferencing	Yes	Yes
Weighted Priority Queue	Yes	Yes

Figure 7 Prototype Feature Comparison

2.3 External Interfaces

The client-facing app will require a hardware interface device to be run on, e.g., a Windows 10 PC, Mac, or a Chromebook. The client's software will use React Native framework and SQLite for local storage. The backend API will be set up to run on an Amazon Web Services (AWS). Express.js will allow Node.js to easily handle HTTP request such as GET, POST, UPDATE, and DELETE. Sequilize will map JavaScript object to the PostgreSQL database.

Appendix

N/A