Lab 2: Inclusive Classroom Product Specification

Randy D Layne

CS 411W

Professor T. Kennedy

October 29, 2021

Version 1

Table of Contents

Table of Figures

Table of Tables

1 Introduction

Young underprivileged students are adversely affected by having to learn online because of a lack of stable high-speed Internet and because they lack the knowledge and support at home needed to adapt when the Internet fails. The onset of the COVID pandemic in April of 2020 forced school systems to adjust to new teaching styles. While colleges have been using resources for online learning for decades, grade schools are not so fortunate. The shift in teaching style is affecting students' ability to learn. Younger students are struggling the most with the change to virtual or hybrid learning. If the student encounters a technical problem, there is not much they can do about it. The student will likely ask their parents, who might lack the troubleshooting knowledge, to fix the problem. The next course of action is to contact the teacher, who, like the parent, likely lacks the skill necessary to deal with a technical issue. The other half of the problem is the lack of stable high-speed internet access at home for lower-income families. Roughly 40% of students from low-income homes must complete their homework on public internet connections. These connections tend to be slow when many people are on at one time, and frequently there is a time limit, too. A young student relies on their parents or a guardian to take them to public places, which might not always be possible.

Students that do not have reliable internet could face disciplinary actions because of not being able to submit assignments on time. These actions could be in the form of points deducted or zeros for their assignment grade. Figure 2 shows what the current process is like for a student trying to submit an assignment and how dependent the process is on having stable high-speed internet. The process flow and difficulties a student faces trying to attend a live video conference are similarly illustrated in Figure 1

[This space intentionally blank.]

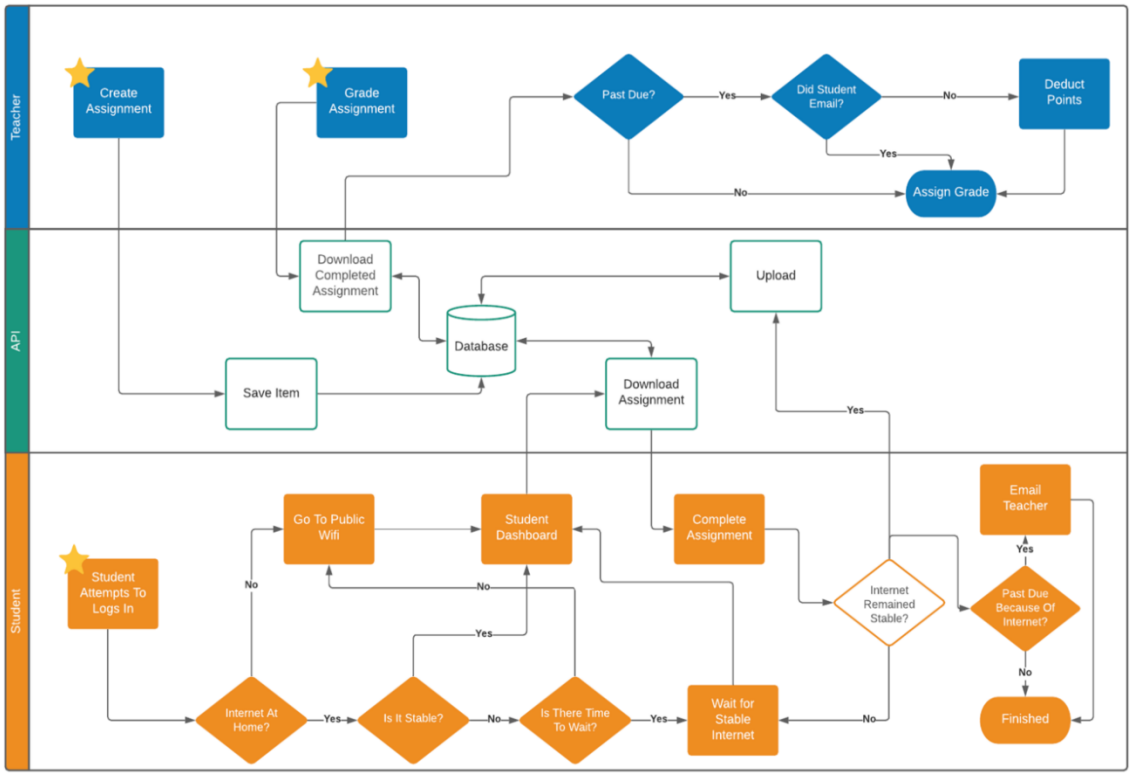

Figure 2: Current Assignment Flow

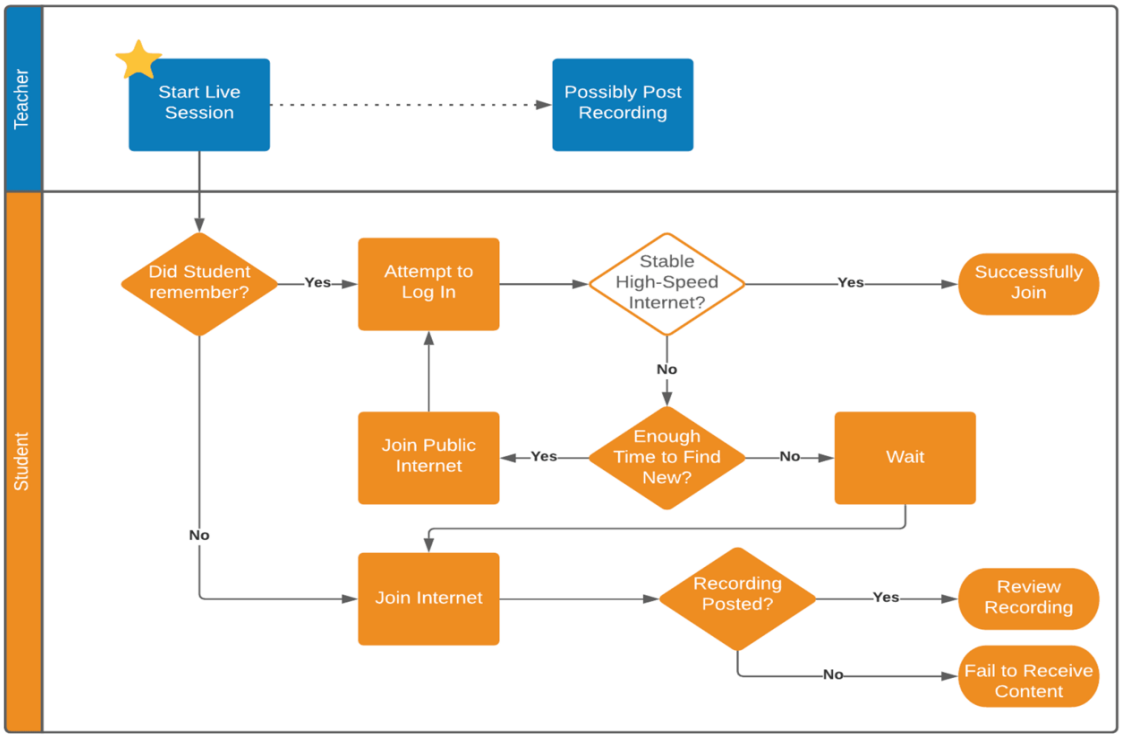

Figure 1: Current Video Flow

1.1 Purpose

Inclusive Classroom will be a native application for all user interactions combined with and API for long-term storage and retrieval of data. The student-facing sections of the software will have a simple UI that is easy to use for children. It will be a native application so that it can run without needing internet access. The student software will have a high level of automation, such as uploading and downloading files and zipping and unzipping files to make things as simple as possible. The teacher-facing view will not need to be constrained to a particular device since a teacher will likely have good internet access at work. Teachers will be able to log in on any device via login and password. The teacher interface will be designed with convention over configuration in mind to keep things simple for the student. For a detailed breakdown of how the solution helps simplify the process of downloading and completing assignments, see Figure 3.

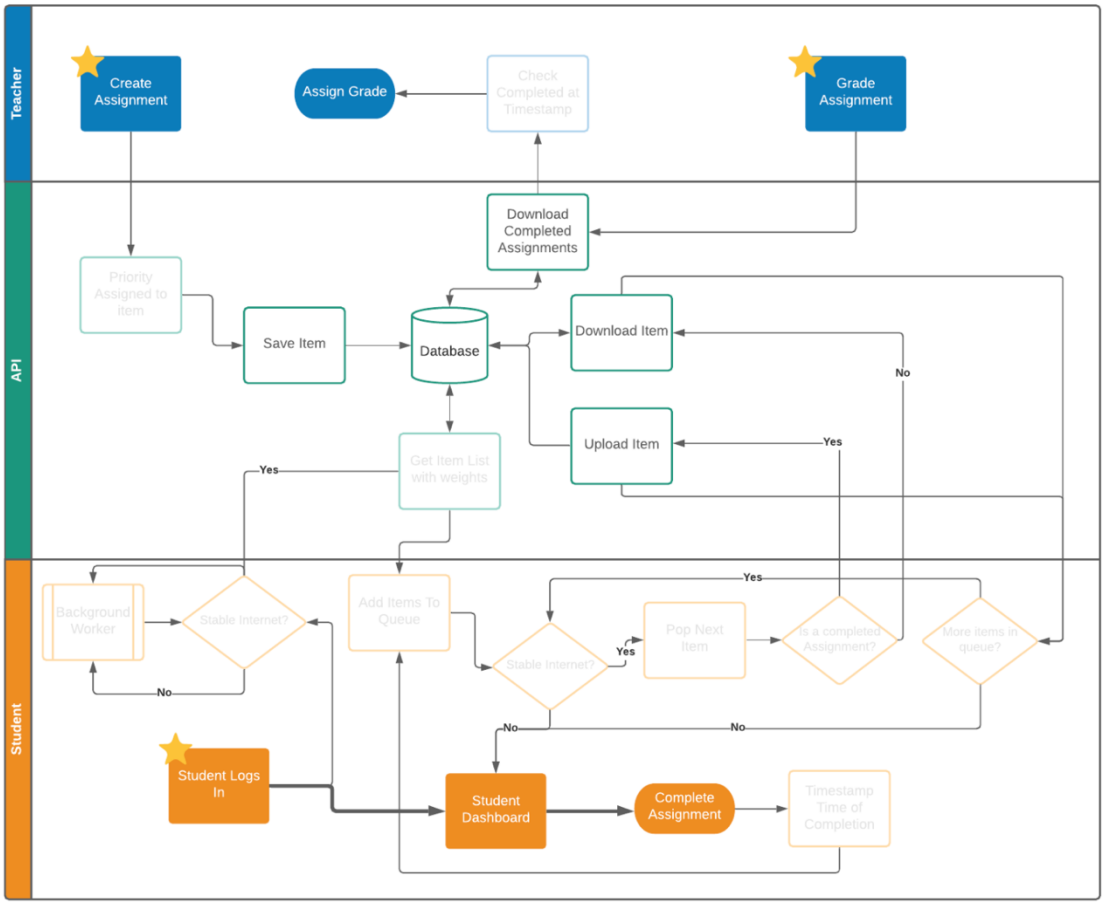

Figure 3: Solution Flow - Assignments

Inclusive Classroom's primary goal is to increase the accessibility of online learning to low-income students. The solution is a two-pronged approach with the primary sections being the student-side flow and the teacher-side flow. The student side will have the ability to passively download and upload, both assignments and lectures. We will also be attaching a timestamp to completed assignments to enable teachers to determine whether an assignment has been completed in time. The major goal of the teacher-side flow will be to allow the teacher to interact with these students with as little headache as possible. This will be done by automating processes for uploading lectures and sending notifications of live stream status to students. The teacher will also be able to easily review the timestamp associated with the assignment.

1.2 Scope

The primary focus of Inclusive Classroom is helping students complete schoolwork in a virtual classroom environment. The design places particular emphasis on students based in the U.S. who don't have stable high-speed internet and that lack a support group with the necessary skills to help when there are computer and internet issues.

Due to the time constraint of having to present a working prototype by December 2021, the prototype for Inclusive Classroom will focus on the key elements that make IC unique amongst the competitions, such as the automatic uploading and downloading of materials when an internet connection is detected. Of the four major roles of the real-world IC, the focus for the prototype will only be on two of those: the student and the teacher roles, with some limited functionality of the admin role added as well. The key to demonstrating the automation portions of the application is to have enough working functionality in the prototype to have teacher be able to create and grade assignments and student be able to complete those assignments, with the software being responsible for downloading and submitting them.

[This space intentionally blank.]

1.3 Definitions, Acronyms, and Abbreviations

ESL - English as a Second Language

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

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

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

HTTP – Hypertext Transfer Protocol

IC - Inclusive Classroom

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

ORM – Object-relational mapping; programming technique for converting data between incompatible type systems

RFC – Request for Comments; a formal document from the Internet Engineering Task Force

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

UI - User Interface

1.4 References

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.

“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.

1.5 Overview

The purpose of this product specification is to describe the general architecture of the Inclusive Classroom prototype. This includes the major software components, UI interfaces, API interfaces, layouts, and features of the prototype. The requirements section will also specify in detail the expectations of the finished prototype.

2 General Description

Inclusive Classroom will be designed to mitigate the current issues with virtual learning affecting students and teachers. As a native application, Inclusive Classroom will prevent under-privileged students from having to regularly access the internet to receive a quality education. Students will be able to view and submit assignments, view lectures, and follow the pace of a class all without internet connection. Teachers will be able to upload lesson plans, determine the class timeline for students to follow, record and upload lectures, and create and grade assignments. A school administrator will be able to view all students and teachers, evaluate a teacher's performance, and view individual students' academic progress to identify students that may be struggling in school. Through the implementation of such features, students, teachers, and administrators will be able to focus on education rather than resolving technical issues.

2.1 Prototype Architecture Description

The client application will serve as an interface between the user and the API. Both the student and the teacher will utilize the client app. Each user type will have to authenticate. Student will authenticate via a token that is created on their school assigned device. The teacher

will use a traditional username and password combination to log into the system. Future iterations will allow single sign on through the institutions SSO system.

The client application will be developed using React Native to create native applications for Android and iOS devices. The React portion of the application will store data locally until it needs to upload it to the API, or when new assignments are downloaded. This will be accomplished through a local SQLite 3 database instance stored on the device. In addition to the storing of assignments, the database will also store video streams, class data, and assignment grades. The component breakdown can be seen visually in the box labeled “Client Side App” in Figure 4.

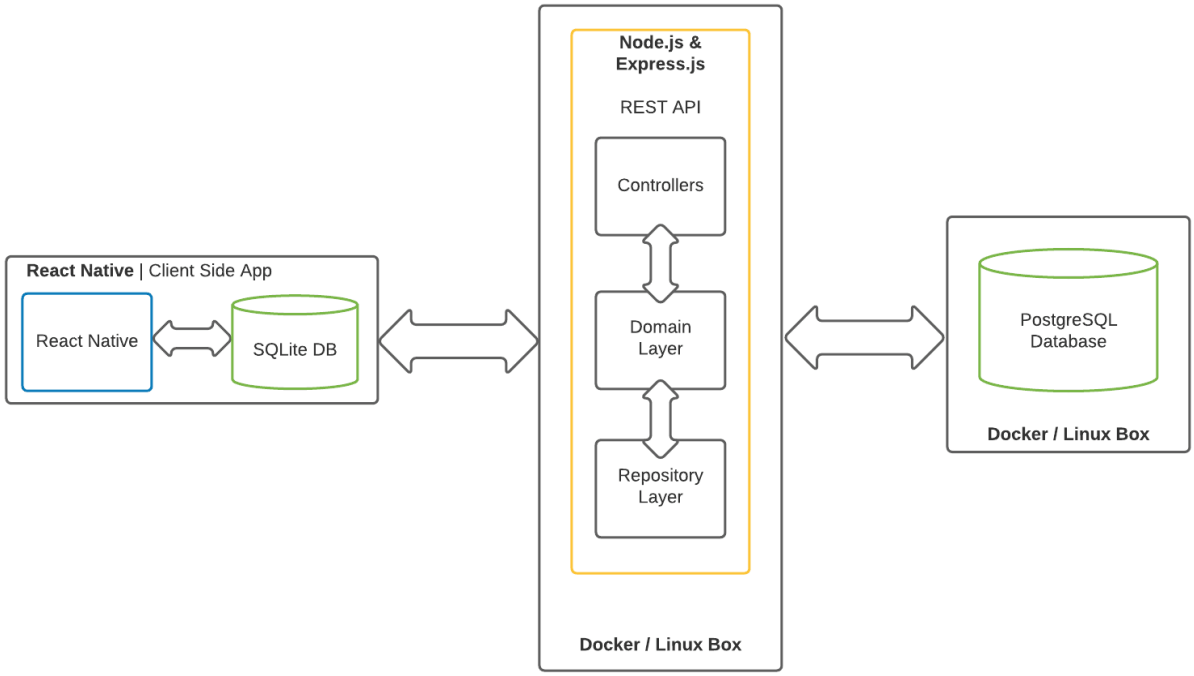

Figure 4: Prototype Component Diagram

The API portion of Inclusive Classroom serves the role of being a central repository for all stored data in which the client app accesses. It will be written in Node JS using the Express framework. It will use Redis as an in-memory database to cache frequently used requests to increase response time. The Express application will communicate with the database through the Sequelize ORM. The application will serve as a gatekeeper to a user looking to download

assignments and videos from the database. It will also store information about managing a class structure. The app will use a priority queue to determine which information, such as assignments and videos, are most important for a user to download and upload at that time. The API will then send the information back to the user in bursts according to the priority assigned to it.

The database for the API will be a Postgres SQL instance stored in Amazon's RDS system. It will store grades and credentials directly. For items it can't store directly, such as assignments and videos, it will store a URL to the corresponding S3 location. The API components are found in **Error! Reference source not found.** under the boxes labeled "Docker/Linux Box."

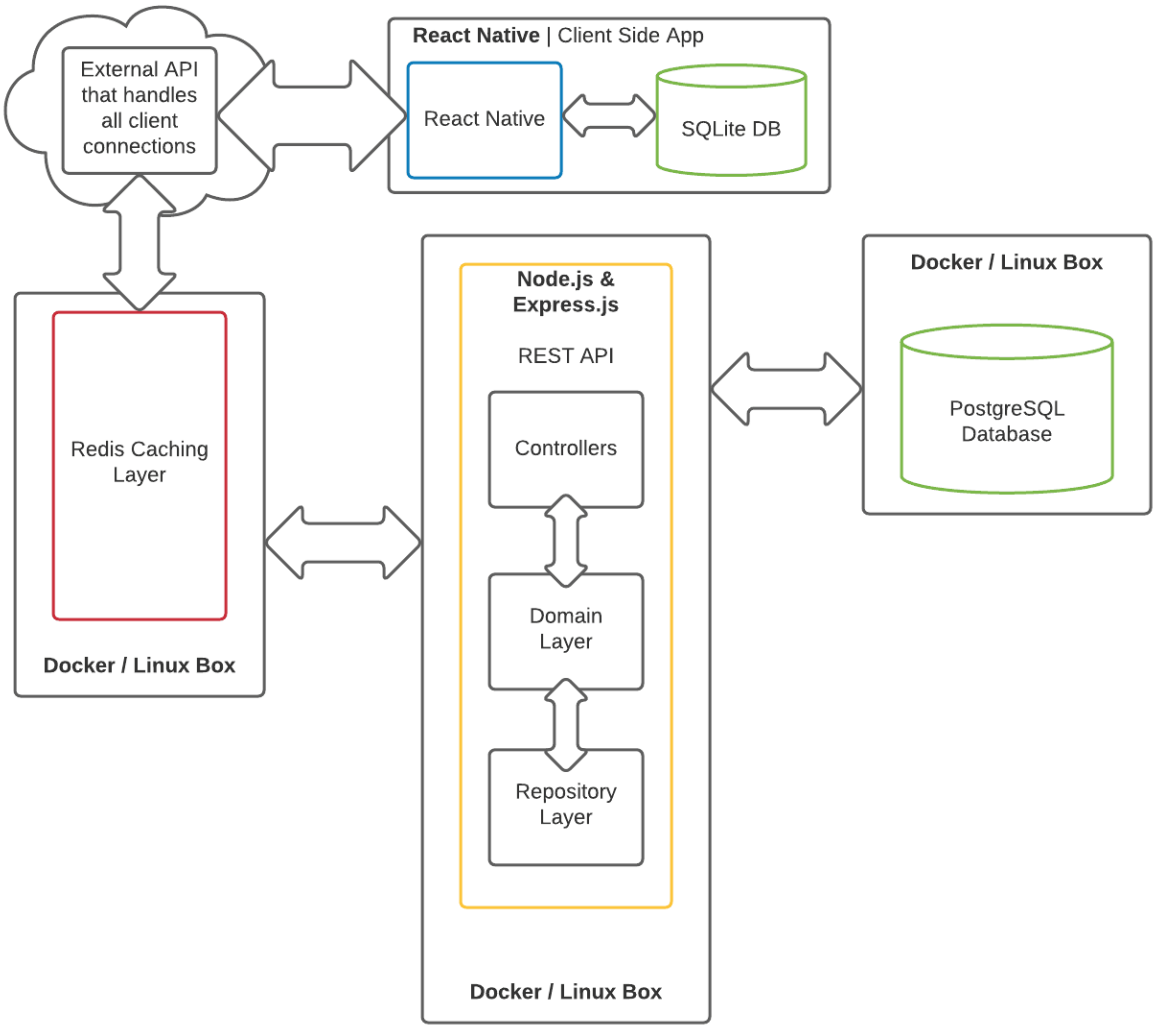

Figure 5: Major Functional Components Diagram

2.2 Prototype Functional Description

The Inclusive Classroom prototype must first be accessible to users on a Windows 10, Mac, or Chromebook device. An internet connection will be required for users accessing Inclusive Classroom for the first time. After ensuring the application is accessible, a user will be prompted to create an account or log into an account. Inclusive Classroom will use authentication to ensure the application is secure and user information is protected. After successfully logging into their account, the user's dashboard view will depend on their user type. For students, the dashboard will consist of a simple UI displaying current assignments and links to video lectures. For teachers, the dashboard will consist of links to create assignments, grading, and starting video lectures. For administrators, the dashboard will consist of links to view all students and teachers, evaluate a teacher's performance, view information about a specific class, and view information about an individual student's academic progress. Inclusive Classroom will include a web conference page for all users to meet in a private setting to discuss personal matters. Inclusive Classroom will utilize automation to mitigate the lack of technical skills amongst users. Inclusive Classroom will feature a weighted priority queue for downloading assignments, automatic detection of the Internet, and automatic submission of assignments when internet access is detected. These features of Inclusive Classroom will ensure that users regardless of their access to internet or technical skill level can participate in a meaningful learning experience.

The Inclusive Classroom prototype will exhibit many of the critical features of the real-world product. The major differences between the prototype and the real-world product will be the absence of an Information Technology specialist role and limited functionality of certain features, such as creating any type of assignment, postdating created content, auto-recording lectures, and reporting. The differences are largely attributed to the lack of time, skills, and personnel required to implement these features. For a list of the features included in the Inclusive

Classroom real-world product and prototype and a better understanding of their similarities and differences, refer to Table 1.

2.3 External Interfaces

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

Table 1: Prototype Features

The prototype for Inclusive Classroom will need to communicate effectively with hardware responses to user input as well as being able to communicate between major components of the prototype itself. These interfaces are split between three major categories: hardware interfaces, software interfaces, and user interfaces.

2.3.1 Hardware Interfaces

The hardware required to implement Inclusive Classroom will differ for the client-facing app and the API. The client-facing app will require a device to be run on, e.g., a Windows 10 PC, Mac, or Chromebook.

The API will be set up and run from the cloud using Amazon Web Services (AWS). Specifically, it will make use of an EC2 instance to host the API itself, an RDS Postgres SQL instance to store the database, Elastic IP to act as a load balancer and gateway between the public and private networks of the API, and finally an S3 bucket to store unstructured data such as scanned documents and submitted assignments. Inclusive Classroom will use Docker to provide containerization and to ensure that local development environments mimic production and Kubernetes to provide a consistent way to deploy changes to the code base into production.

2.3.2 Software Interfaces

Inclusive Classroom will consist of a client application and a back-end API each with their own software requirements. IC's client application will be a React Native web, Android, and iOS application to ensure optimal accessibility for users. The client application will use a SQLite Database for local storage. The API will use Node.js to execute JavaScript on the server-side of the application. Inclusive Classroom's API will use Express.js built on top of Node.js to increase functionality and handle HTTP requests. The API will communicate with a server-side PostgreSQL database.

2.3.3 User Interfaces

Inclusive Classroom's client-side application will serve as an interface between the user and the API (Team Gold, 2021). To run the Inclusive Classroom client-side application, a user will need to have a compatible device such as those listed in section 2.3.1 and some combination of a pointer device such as a mouse, trackpad, or touchscreen as well as a typing device such as a keyboard or touchscreen. Inclusive Classroom will require all user types to successfully authenticate to access their account and the content that Inclusive Classroom will display will be based on the user's authorized role.

2.3.4 Communications Protocols and Interfaces

The Inclusive Classroom prototype will communicate with the API using a REST format and following the standards set forth in RFC 2068 concerning communication via HTTP and HTTPS.