Lab 1: Inclusive Classroom Description

Randy D Layne

CS 411W

Professor T. Kennedy

September 28, 2021

Version 1

Table of Contents

Table of Figures

Table of Tables

[This space intentionally blank.]

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 1 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 2.

[This space intentionally blank.]

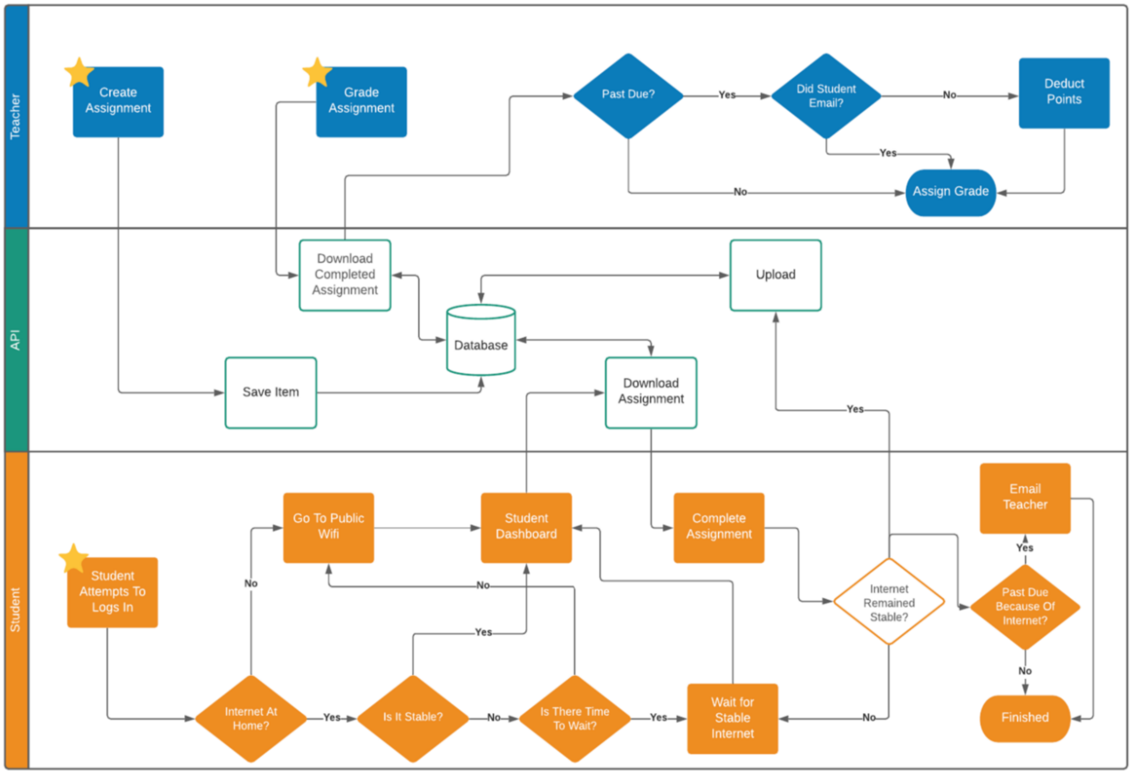


Figure 1: Current Process Flow - Assignments

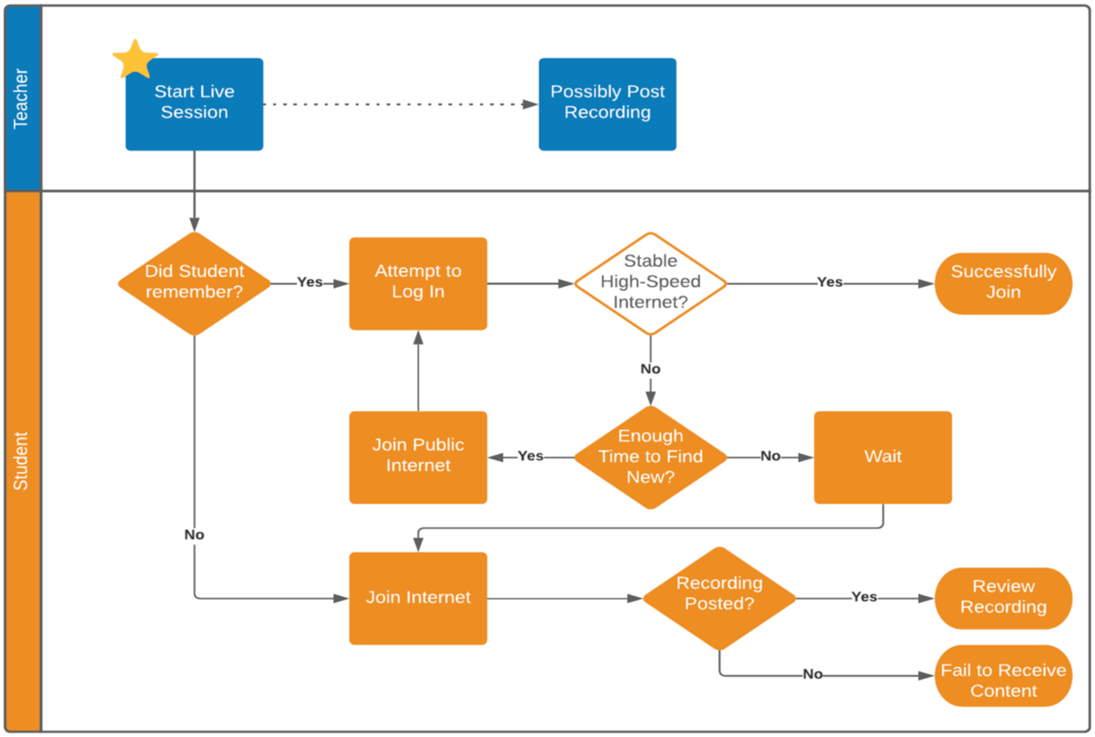


Figure 2: Current Process Flow - Video Conferencing

The solution will be a student-facing application and a teacher/admin application called Inclusive Classroom. The student-facing 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.

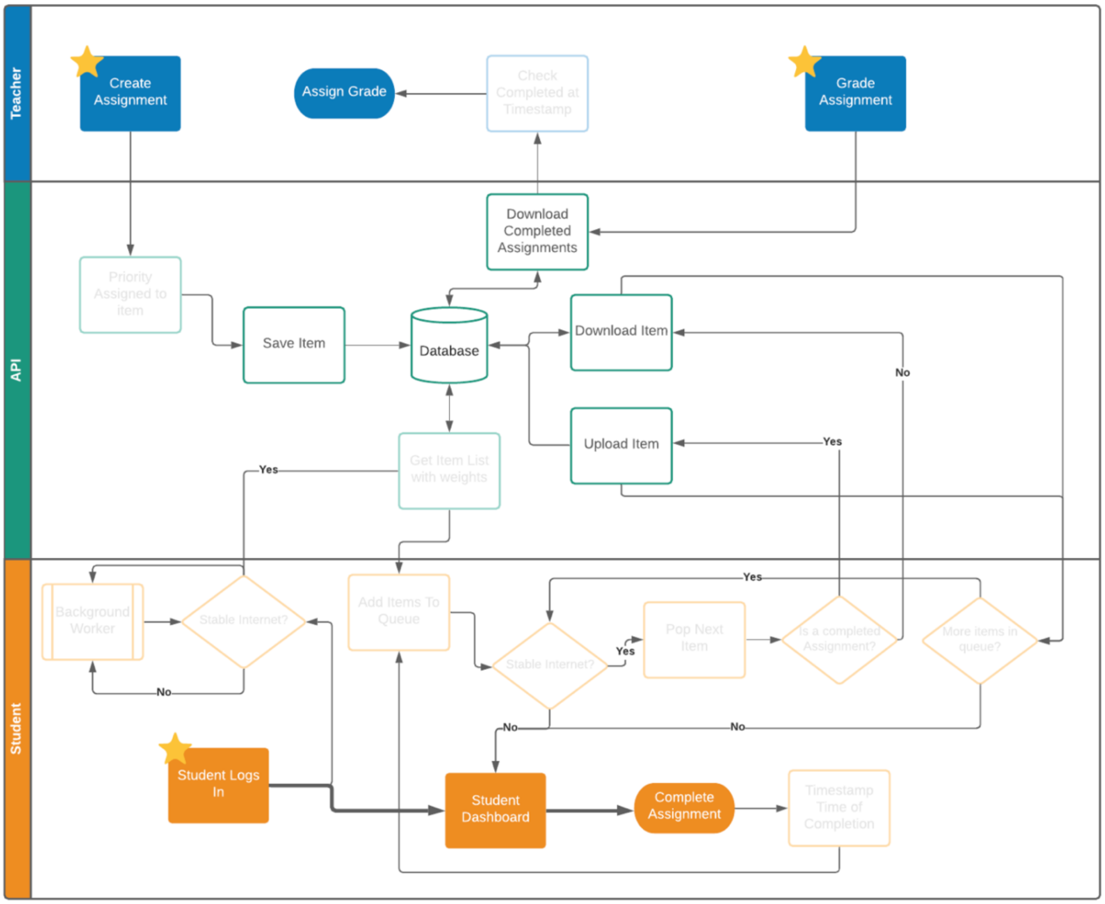


Figure 3: Solution Flow - Assignments

2 Product Description

Inclusive Classroom's primary goal is to increase the accessibility of online learning to low-income students. Our solution is a two-pronged approach. The two primary sections will be 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.

2.1 Key Product Features and Capabilities

The Inclusive Classroom software will have key features and capabilities that distinguish it from the rest of the competition. The biggest concept behind Inclusive Classroom is the idea that many students learn in environments without stable internet connections. The client app of Inclusive Classroom will be able to operate completely without the internet. If the software is set up by the school or parent(s) with internet access, the student can, in theory, go the entire year without internet until the last day. All teaching materials can be prepared beforehand, and the software will manage all submissions, lectures, etc. to keep the student on pace. If the student can access the internet intermittently throughout the year, the software will automatically synchronize with the backend server, submitting the assignments as if the student had submitted them with internet access the entire time.

At the beginning of the school year, teachers will be able to plan out the entire year, including all lectures, assignments, quizzes, exams, class updates, etc. When the student client software is first set up, it will load all the preplanned data and manage it on the client-side. The student will then be able to follow through with the class however the teacher-designed it. They will be able to watch the lectures at their own pace, or a pace deemed by the teacher. They will be able to complete assignments as they open (weekly, for example), submitting them with or without the internet. In this way, as the internet is not required through the year, there will be no degradation in student experience; the software will allow students in areas without internet access, or limited internet access, to learn effectively. Even if the teacher can't plan out ahead of time for the whole year, even the ability to add in assignments on a weekly basis will help students with limited internet access to still be able to get their needed assignments.

The software will also be available on nearly all operating systems and computers. It will support Chromebooks, as well as the main three operating systems, Windows, Mac OS, and Linux. By supporting these, Inclusive Classroom will be an option for nearly all school systems across the United States.

2.2 Major Components

The major components for the application will be split into two major categories: required hardware and software to be developed. The required hardware lists the needed infrastructure to set up Inclusive Classroom's backend on a server as well as the machines that the client-side application can run on. The software to be developed describes the two major programs that together form the Inclusive Classroom application.

2.2.1 Required Hardware

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.

2.2.2 Software to be Developed

There will be two major software components that together form the Inclusive Classroom application: the client-side application and the backend API.

2.2.2.1 Client Application

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.

2.2.2.2 API

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 Figure 4 under the boxes labeled "Docker/Linux Box."

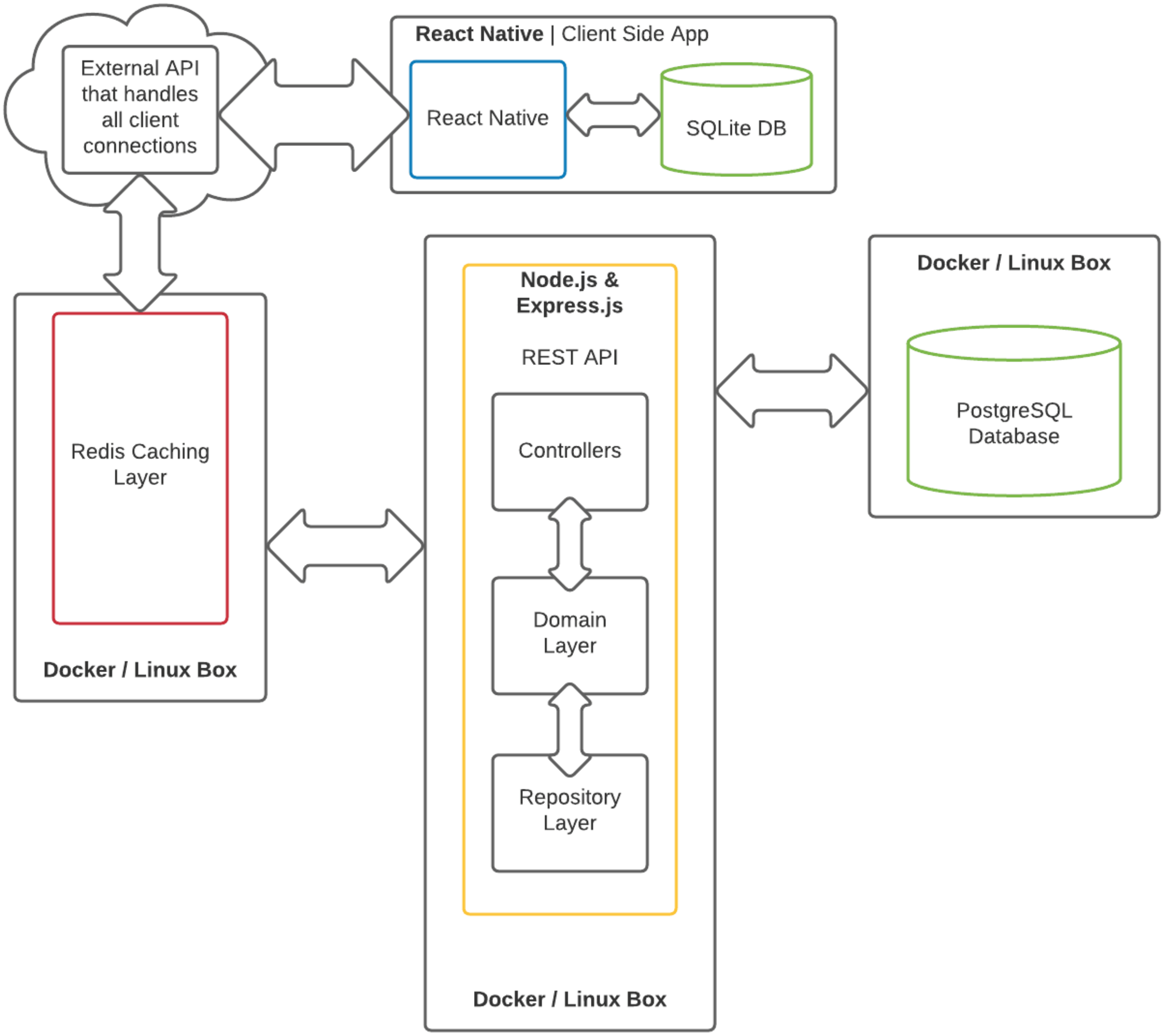


Figure 4: Major Functional Components Diagram

3 Identification of Case Study

Inclusive Classroom's primary target are students that come from low-income families that do not have access to stable high-speed internet. These students will likely not be able to adapt easily when a source of technology fails to function properly. These students will use Inclusive Classroom to view live streaming of classes or view them as a recording if they do not have internet at the time. They will also use Inclusive Classroom to download, complete, and upload assignments for classes. The download and upload portions will happen automatically whenever Inclusive Classroom detects and internet connection without the student needing to click any buttons. The student's main concentration should be on the completion of assignments, with the other tasks being abstracted away.

4 Product Prototype Design

The prototype for the Inclusive Classroom application will include much of the core functionality outlined in Software to be Developed. The prototype will use all the same frameworks and languages that the production counterpart will use.

4.1 Prototype Focus

The most emphasis will be placed on the parts of the application that automate tasks since this is a key differentiator between Inclusive Classroom and its competition. These tasks include:

- Creating a weighted priority queue for downloading assignments.
- The ability to detect when the client device is connected to the Internet.
- Submitting completed assignments automatically when Internet is detected.

4.2 Client-Side UI

The UI of the client-side prototype will include five main views that teachers, students, and administrators will interact with. All views are also in the production design but may not have the same level of functionality. The different actions that can be performed inside the prototype and how they relate to actions in the production version of Inclusive Classroom are illustrated in Table 1.

4.2.1 Login View

The login view will be universal in that each role will be required to log into their respective dashboards through this view.

4.2.2 Student Dashboard

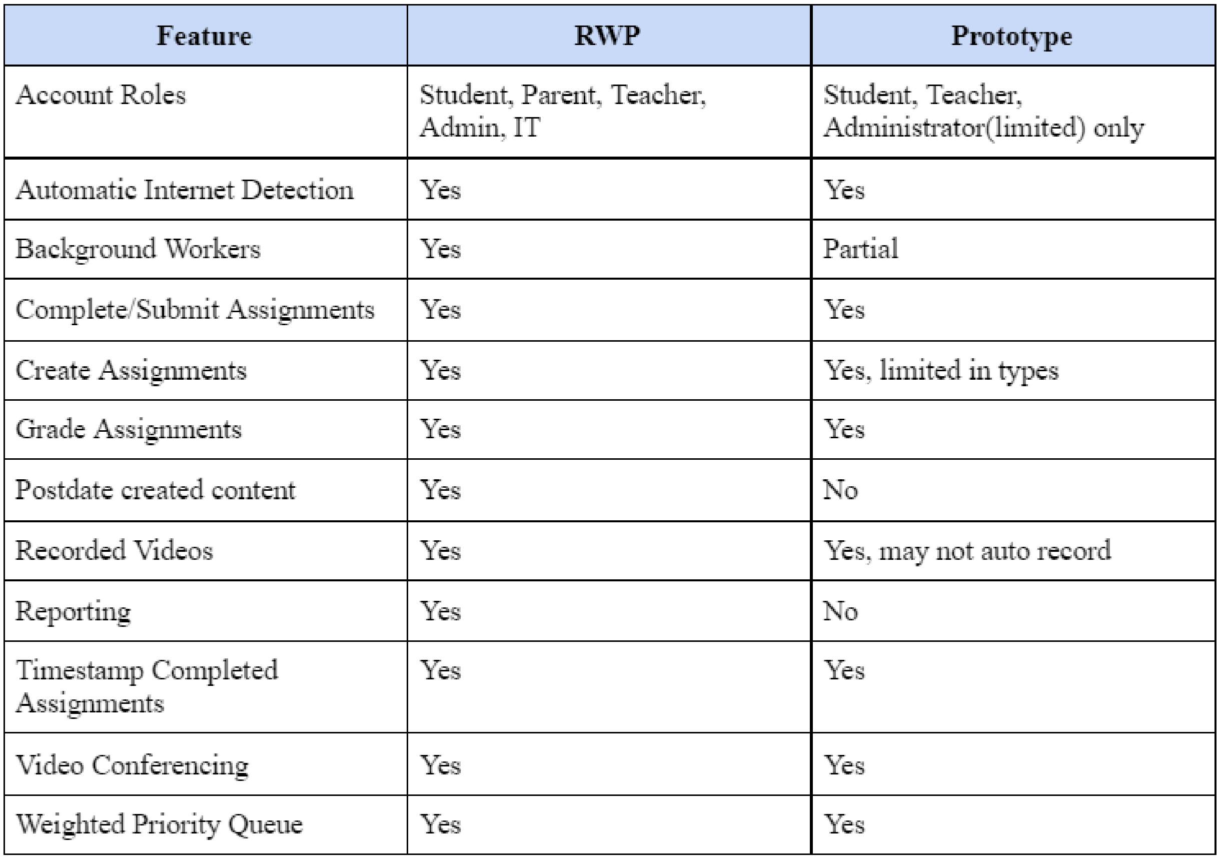
Students will use the student dashboard view as the overall view to reach the other sections they need. The prototype will focus on displaying current assignments, links to assignments, and links to videos.

4.2.3 Teacher Dashboard

The teacher dashboard for the prototype will be a limited functionality version of the production. The dashboard will include sections to access assignments and video sessions. For functionality that will not be included in the prototype, the dashboard will contain placeholder components to ensure that it has the same look and feel as the production version. One of the main sections that will be placeholder and not functional in the prototype is the reporting section.

4.2.4 Assignment View

The assignment view will be an area that both teachers and students have access to, but the content and functionality will adjust based on the user role. Teachers will have access to creating and grading assignments. The type of assignments they can create will be limited to homework and quizzes. The prototype will not offer an exam homework type like the production version will. The format for questions inside of the assignments will be limited to multiple choice, short answer, and true or false. Their will not be a place in the prototype to create essay questions or to allow uploading of external documents. Students will be able to use this view to complete and submit assignments.



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

4.2.5 Video Conference View

The video conference view allows teachers to create a new video conference and record the conference session. Students will be able to join a live session that the teacher created or view a recording of that session later through this view.

4.3 Prototype Architecture

The architecture for the prototype will have many of the same properties as the production version. These are broken into two sections, the hardware required to run the application and the software that will be developed by the Inclusive Classroom team.

4.3.1 Required Hardware

The backend API for the prototype will use Amazon Web Services (AWS) for the infrastructure. Individual AWS products include S3 buckets, E2C compute instances, Elastic IP, and RDS. Docker will be used to containerize the code and it will be deployed using Kubernetes.

The client-side application will be required to operate on Windows 10 PCs, Macs, and Chromebooks. A physical example of each will be used during development to ensure compatibility.

4.3.2 Software to be Developed

The prototype functionality will be broken into two sections, a client-side application that have local copies users interact with directly, and a server hosted API that will act as a gateway for many instances of the client-side application. An overview of the software components and how they relate to the infrastructure can be seen Figure 5.

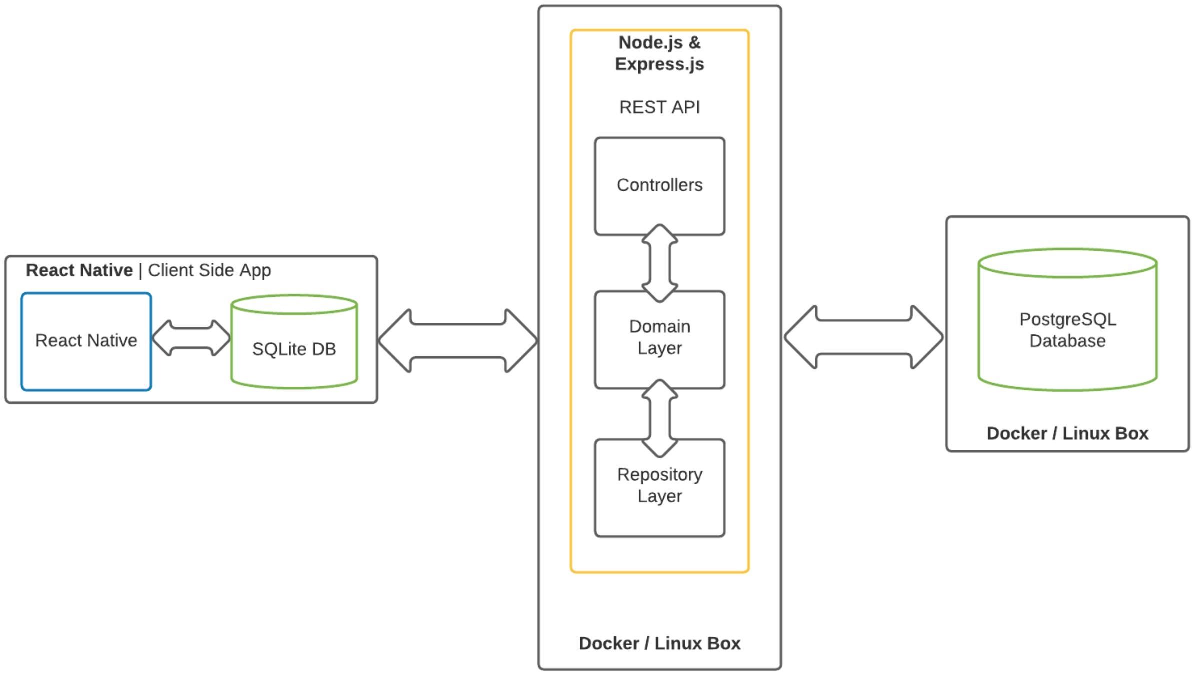


Figure 5: Prototype Component Diagram

4.3.2.1 API

The API for the prototype will be implemented using the Express micro framework built on NodeJS. The API will provide an access point for the different instances of the client-side app to connect to and store data. There will be endpoints to upload and download assignments, handle user authentication, and limit user role access through authorization. It will connect to a PostgreSQL database. The database will store assignments directly and will store reference links to recorded videos that will be stored in Amazon S3 buckets. The API will also be tasked with setting up the priorities of assignments in the queue. Additionally, the API will provide push notifications to the client-side application when a live event is happening.

4.3.2.2 Client-side Application

The client-side application will serve as the interface between the user and the API. Both the student and the teacher will utilize the client application. Each user type will have to login to authenticate. The client-side app will be implemented using React Native to build a cross platform application that will run on a PC, Mac, or Chromebook. The prototype client-side application will be able to schedule, view, record, and upload live video sessions. It will be able to create, view, download, upload, and grade assignments. The application will have functionality to create and manage classes.

4.4 Prototype Features and Capabilities

The prototype will have limited functionality compared to the production version. The MVP for the prototype will contain the following functionality:

- Teachers and students will be able to create accounts and login.
- Teachers will be able to create classes and add students to classes.
- Teachers will be able to create assignments.
- Teachers will be able to schedule and open zoom sessions for students to join.
- Students will be able to view the classes they are in.
- Students will be able to view and submit their assignments before the due date.
- Students will be able to watch a lecture live or watch the recording after.

The interactions that each user role is authorized to do on the client-side application are illustrated in the site maps for the student and teacher. The student site map can be viewed in Figure 6 and the teacher site map is Figure 7.

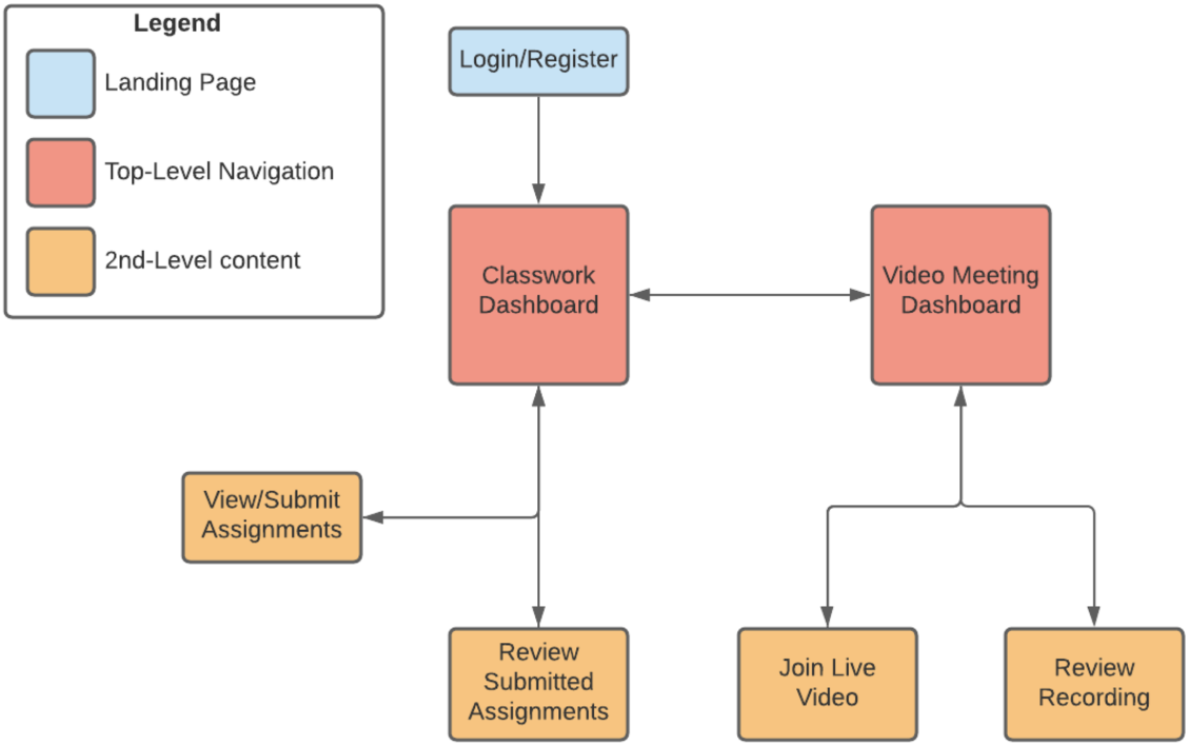


Figure 6: Student Site Map

[This space intentionally blank.]

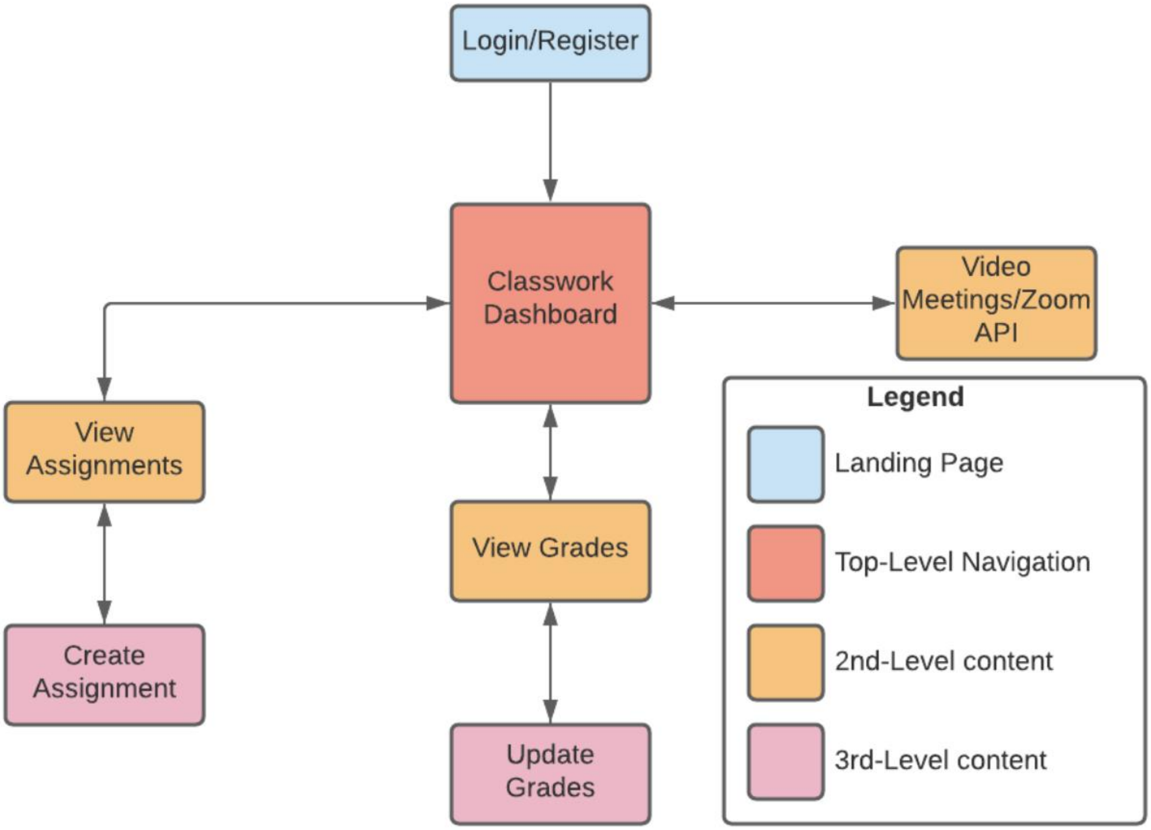


Figure 7: Teacher Site Map

[This space intentionally blank.]

5 Glossary

English as a Second Language (ESL) – Students whose primary language is other than English

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)

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

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

6 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.