CS 411W Lab 1

Inclusive Classroom

Grant Ralls

13 September 2021

Table of Contents

Listing of Figures

1 Introduction

Young underprivileged students are adversely affected by being required to learn online because of a lack of stable, high-speed Internet. They also lack the knowledge and support at home needed to adapt when the Internet fails. School systems were forced to adapt to new styles of teaching due to the COVID pandemic starting around April of 2020. 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 with the shift to virtual or hybrid learning. If the student encounters a technical problem, 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 have slow connections when many people are on at one time, and there is frequently a time limit as well. A young student relies on their parents or a guardian to take them to public places, which might not always be feasible. The solution is a student facing app and a teacher/admin app called Inclusive Classroom. The student view will have a simple UI that is easy to use for children. It will be a native app 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 easy for the student as possible. The teacher view will not need to be constrained to a certain device since a teacher will likely have good internet access at work. Teachers will be able to log in on any device. The teacher interface

will be designed with convention over configuration in mind to keep things simple for the student.

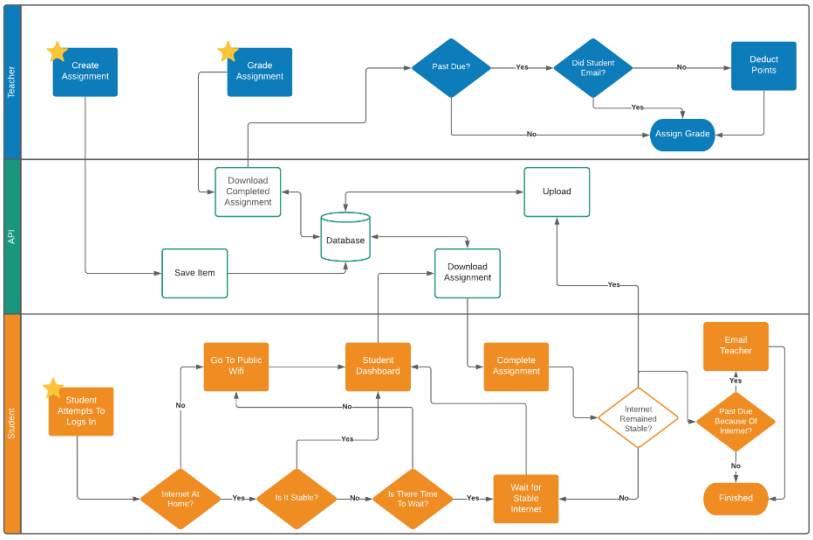


Figure 1: Assignment Process Flow Diagram

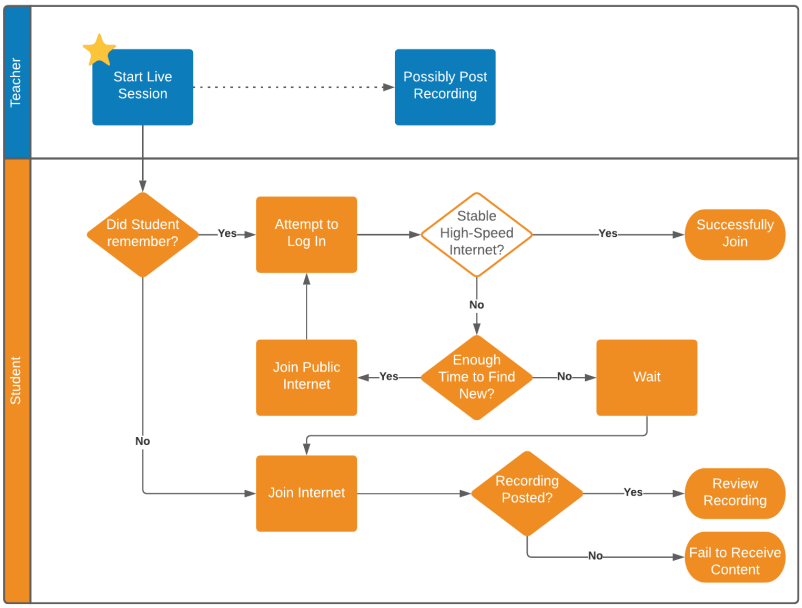


Figure 2: Live Video Process Flow

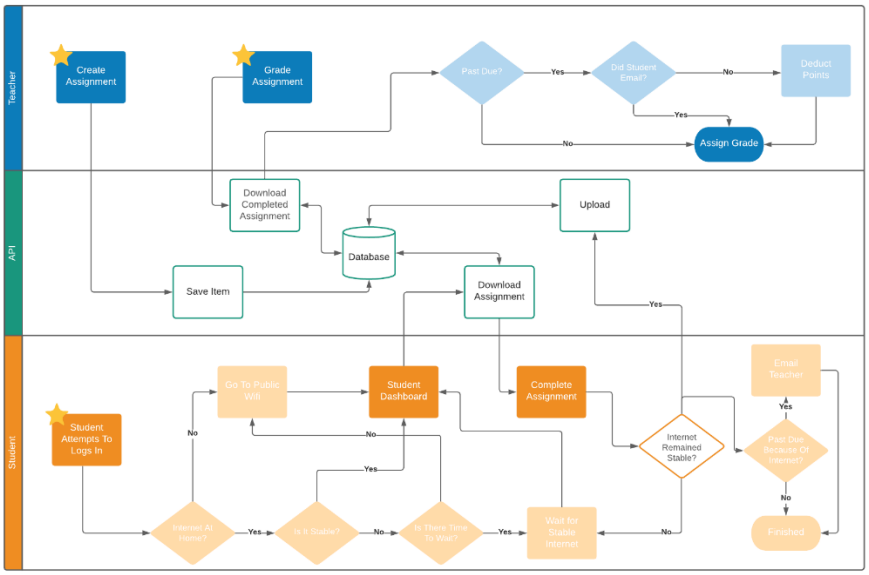


Figure 3: Current Process Flow Revisited

2 Inclusive Classroom Product Description

Inclusive Classroom's (IC) primary goal is to increase the accessibility of online learning to low-income students. The solution is a two-pronged approach. The two primary sections will be the student flow and the teacher flow. The student will have the ability to passively download and upload, both assignments and lectures. Inclusive Classroom will attach a timestamp to completed assignments to enable teachers to determine if an assignment has been completed in time. The major goal of the teacher 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 (IC) software will have key features and capabilities that distinguish it from the rest of the competition. The biggest concept behind IC is the idea that many students learn in environments without stable internet connections. The client application of IC 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 flow is first set up, it will load 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, IC will be an option for nearly all school systems across the United States.

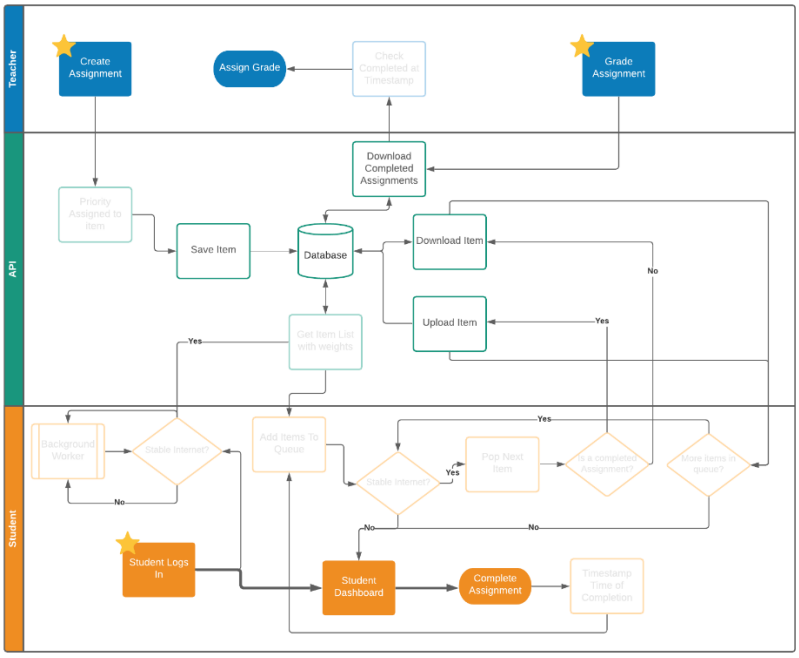


Figure 4: Solution Process Flow: Assignments

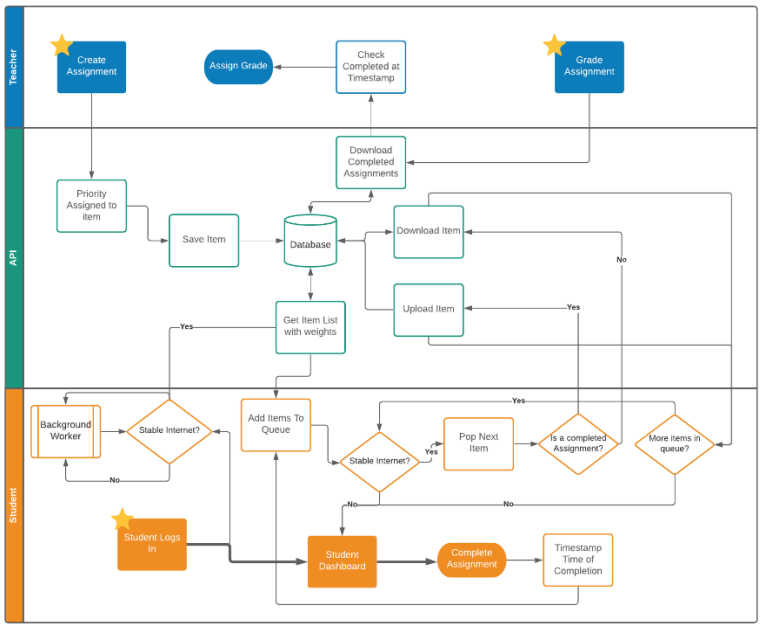


Figure 5: Algorithms Process Flow

2.2 Major Components (Hardware/Software)

Inclusive Classroom will be software developed with hardware in mind. Inclusive Classroom will be cross-platform, but it is expected that certain parts of the software to be used by specific hardware. The student flow is expected to be accessed from a Chromebook while the teacher workflow is expected to be accessed from a Mac or a Windows 10 device.

The software to be developed will be split into two major parts, the client-side application and the API. The client application will serve as an interface between the user and the API. Both the student and the teacher will utilize the client application. The application will have an authentication flow for both types of users. The client-side application will be developed with React Native. This allows us to have access to the Android/iOS APIs while also being able to build the app for the web. The React Native application, paired with a SQLite store, will be able to view livestreams/recordings, record/upload livestreams, download/upload assignments for teachers and students, manage classes, assign/upload grades for assignments.

The API will be built using Node, Express, Postgres, and Redis. It will have the functionality of saving videos to the database, authenticate users, upload/download assignment data, get assignment lists as a priority queue, and send push notifications. Mocha will be used as the testing framework. The API will be shipped and hosted from Docker containers. Docker eases the development process by eliminating inconsistencies between environments.

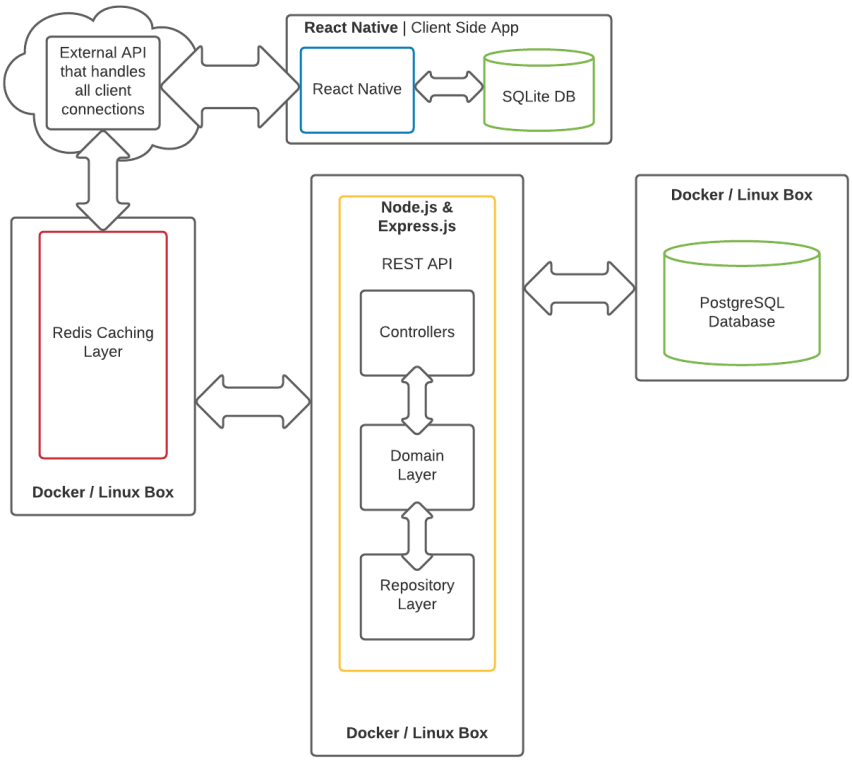


Figure 6: Major Functional Component Diagram

3 Identification of Case Study

The target demographic for IC is low-income students who do not have access to stable, reliable, high-speed, internet. Teachers will also use the software to provide content for the students. The software will be used to livestream, view video, and record video. The software will also be able to upload, download, and timestamp assignments. A family in the future could use Inclusive Classroom as a second option if faulty internet is a concern.

4 Product Prototype Design

When developing the prototype, emphasis will be placed on the automation aspects of the proposed solution. This includes the weighted priority queue, automatic detection of the internet, and automatic submission of assignments when internet access is detected.

The design of the prototype will have the views necessary to demonstrate the following. Displaying current assignments/video, create/grade assignments, start video sessions, and login flow. Only a selection of assignment types will be available in the prototype. These include homework and possibly quizzes with true/false, multiple choice, and short answer questions. The technology used to develop the prototype will be the same as the technology used for the final product.

4.1 Prototype Architecture (Hardware/Software)

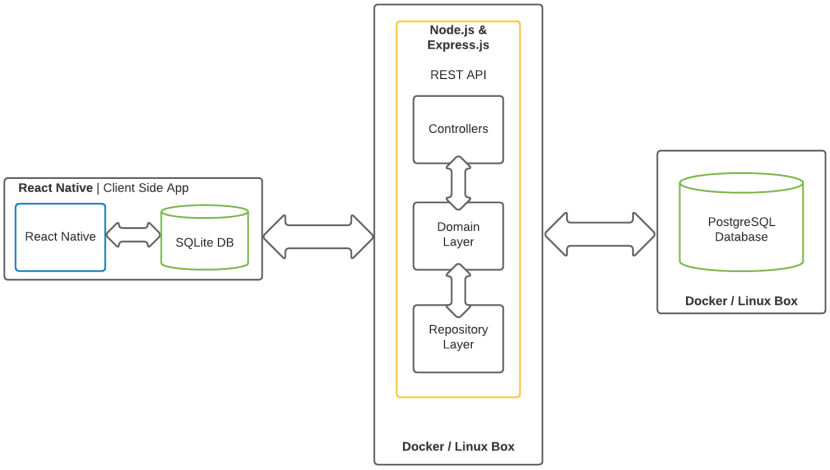


Figure 7: Revised Major Functional Component Diagram

The required hardware for the client application is only a device to access the application itself. This can be a Windows 10, Mac, or Chromebook device. The API will need Amazon Web Services' S3 along with Docker and Kubernetes.

The required software includes the client application as well as the API. The client application will serve as an interface between the user and the API. Both the student and the teacher will be intended users of the client application. The application will be developed using React Native. This allows us to build to specific environments. The final application will be able to view livestreams/recordings, record/upload livestreams, download/upload assignments, manage classes, and assign/upload grades for assignments.

The second half to the required software is the API. The API will be developed with Node, Express, and Postgres. It will save videos to the database, download/upload assignments, authenticate users who want to download/upload assignments, get assignment lists as a priority queue, and send push notifications.

4.2 Prototype Features and Capabilities

Both teachers and students will be able to create accounts and log in. Teachers will be able to create classes, add students to classes, create assignments, and schedule/open zoom sessions for students to join. Students will be able to view their classes/assignments, submit assignments, and watch a live or recorded lecture.

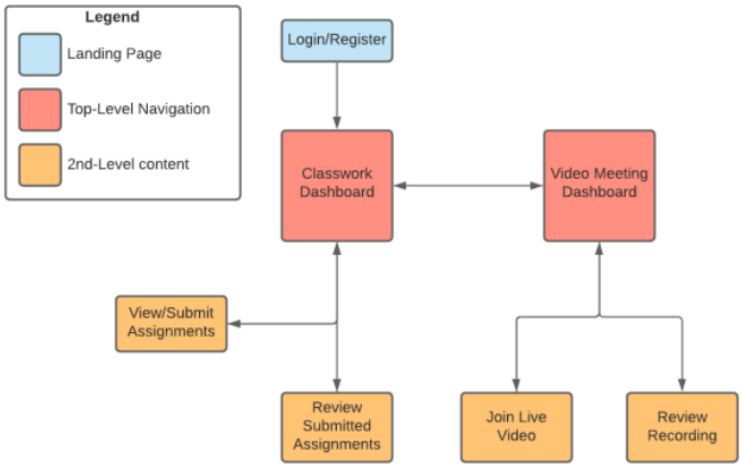


Figure 8: Student Site Map

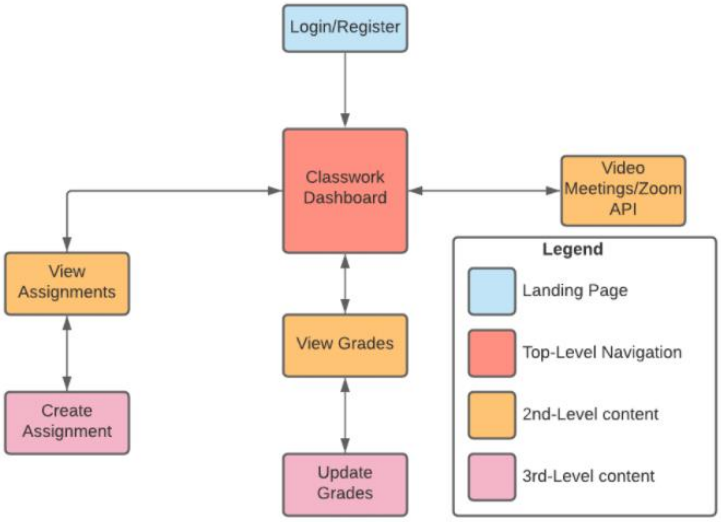


Figure 9: Teacher Site Map

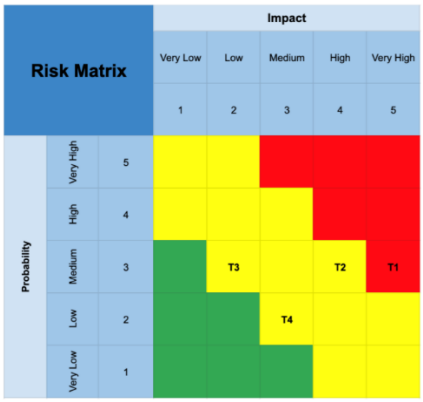
Student Privacy: (C1)
Mitigation: Student data will only be available to the administrators that absolutely need it, and teachers can request permission for student data if needed. Teachers and school administrators will need to go through privacy training to follow FERPA guidelines.

Teacher Privacy: (C2)
Mitigation: Teacher data will only be available to the administrators that absolutely need it, and teachers can request permission for teacher data if needed. Teachers and school administrators will need to go through privacy training, and sign waivers to be recorded during lectures.

Missed or Ignored Reminders and Due Dates: (C3)
Mitigation: The Inclusive Classroom software can include an 'acknowledge' button that proves a student and/or parent, via email or the UI, has seen an announcement, due date, or other important piece of information.

Poor English: (C4)
Mitigation: Inclusive Classroom will be English only, but for students in ESL, a section of the UI can be translated to their selected language so that they can access ESL support through their language.

Figure 10: Customer Risks



Loss of internet during stream: (T1)
Mitigation: Flag session as unseen and automatically downloaded recorded version when internet is restored. Optionally, notify the teacher so they can follow up with the student for any extra help needed.

Loss of Internet in mid file transfer: (T2)
Mitigation: Automatically detect dropped packages and resend when internet is restored. A date will be stored for all file transfer attempts.

Assignments can't be sent until after due date: (T3)
Mitigation: A timestamp will be created when student completes an assignment, and this timestamp will be used when comparing the submission date to due date. This will help ensure that the student can reliably submit assignments on time without internet.

Class updates not visible to students without internet: (T4)
Mitigation: Teachers will enter the syllabus into the system before the start of the year. All assignments and dates will be cached locally on the student side. Students can then receive updates throughout the semester without internet, and class updates will cache and change whenever connected if there are updates from the server.

Figure 11: Technical Risks

SQL Injection: (S1)
Mitigation: Databases will be parameterized though typed parameters and careful use of stored procedures. All web components (libraries, plug-ins, web server, database server software) will be kept up to date for security patches, and input will be sanitized.

Data release: (S2)
Mitigation: Sensitive student/teacher data will be encrypted on the backend in correspondence with privacy and FERPA laws. All transferred data will also be via HTTPS.

Broken Access Control: (S3)
Mitigation: Enforcement on the server-side logins will prevent students from accessing teacher permissions. Additional tests will be made to ensure specific URLs will be "hidden" without proper authentication.

Broken Authentication: (S4)
Mitigation: Authentication will be primarily based on school provided tokenization of student logins. Student passwords will be properly formed in a secure matter. Periodic password updates will be required to ensure security.

Figure 12: Security Risks

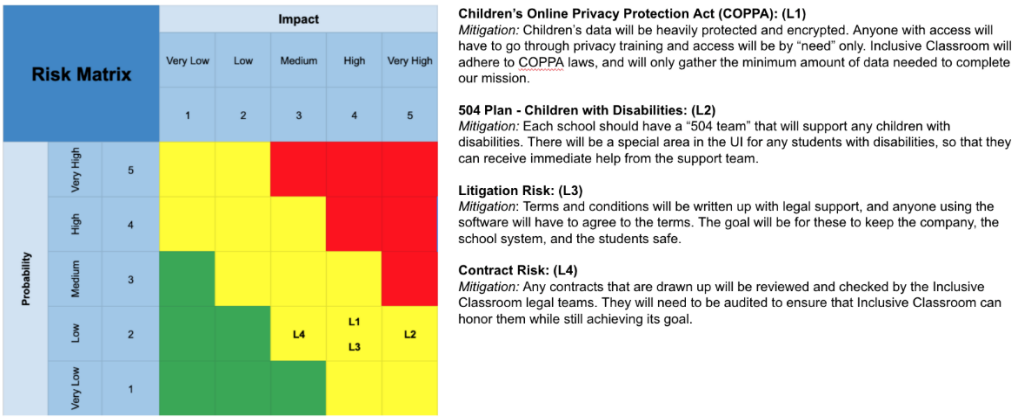


Figure 13: Legal Risks

4.3 Prototype Development Challenges

There are some development strains that are foreseeable. The time constraint for the length of the course will be difficult to work with. The development team may have to learn new technologies and frameworks that are required for the implementation of IC. Deployment of Docker containers to servers may require some funding. Emulating a Chromebook for the student application could also prove challenging. Finally, setting up a system where testing can be done on multiple operating systems will need to be done.

5 Glossary

1. High-speed Internet - Internet with consistent download speeds of at least 3.8 Mbps (Zoom)
2. English as a Second Language (ESL)
3. Family Educational Rights and Privacy Act (FERPA) - Federal law that protects the privacy of student education records
4. Google Classroom - "Free web service developed by Google for schools that aims to simplify creating, distributing, and grading assignments" (Google)

5. littleLearners - Former CS 410 group solution that emphasizes simple UI for students in the K-5 age range (Del Razo)
6. Stable Internet - Internet with less than 1% dropped packets (ICTP)

6 References

1. 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/.
2. “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.
3. “Chromebook Support.” LCS, www.lcsedu.net/departments/information-technology/chromebook-support.
4. “Classroom FAQ - Classroom Help.” Google, Google, support.google.com/edu/classroom/answer/6025224?hl=en&ref_topic=7175444.
5. Del Razo, Gabriel, et al. “LittleLEARNERS.” Team Orange, www.cs.odu.edu/~cpi/old/410/orangf20/.
6. 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.

7. 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.
8. 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.
9. 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.
10. “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.
11. “The 504 Plan.” The Center for Children with Special Needs, 2018, cshcn.org/childcare-schools-community/the-504-plan.
12. VBCPS. “VBCPS Adds 19,000 Chromebooks to Achieve 1:1.” Virginia Beach City Public Schools, www.vbschools.com/news/archived_news/2019/chromebooks.
13. 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

14. 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/.
15. “Web Applications with Spring.” Spring, spring.io/web-applications.