Lab 2 - Product Specification Outline

CS 411W Lab II

Product Specification

Colton Hurst, Team Gold

October 29, 2021

Table of Contents

List of Figures

1 Introduction

1.1 Purpose

Young underprivileged students are negatively 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. Our product, Inclusive Classroom, aims to solve this issue.

When the COVID-19 pandemic started around April 2020, school systems were forced to adapt to new styles and delivery methods of teaching. While colleges have been using resources for online learning for decades, grade schools are not so fortunate. The shift in teaching style is mitigating students' ability to learn. Out of those afflicted, mostly younger students are struggling the most with the shift to virtual or hybrid learning. If the younger student encounters a technical problem, there isn't 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. These connections tend to have slow connections when many people are on at one time, and frequently there is 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 possible.

Our solution, Inclusive Classroom, is a student facing app and a teacher/admin app. 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 via login and password. The teacher interface will be designed with convention over configuration in mind to keep things simple for the student.

The entire community of teachers, students, parents, and all others involved in helping teach children should benefit from Inclusive Classroom. We hope that this solution will bring joy to learning again.

1.2 Scope

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. One of the biggest goals for IC is that the client app of IC will be able to operate completely without the internet. As long as the software is set up by the school or parent(s) with internet access initially, 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. In a situation where 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, IC will be an option for nearly all school systems across the United States.

If a case study was done on our software, this is the MVP (minimum viable product) case study results we are aiming for:

- Inclusive Classroom is being developed for...
  - Low income students who do not have access to stable and reliable internet.
- Inclusive Classroom will be used for...
  - Live Streaming, Viewing, and Recording Lectures
  - Uploading, Downloading, Timestamping, and Grading Assignments
- Who else might use Inclusive Classroom in the future?
  - Families who want a fallback mechanism in the possibility of faulty internet.

[This space is intentionally left 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. This is the name of Team Gold’s software package, aimed to solve the issues we see with underprivileged students in schools within the K-5 age range

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

[This space is intentionally left blank]

1.4 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.
16. Team Gold. (2021, October 19). Lab 1 – Inclusive Classroom. Retrieved October 29, 2021 from the private Team Gold Google Drive.

[This space is intentionally left blank]

1.5 Overview

The Inclusive Classroom software will be a client side application built with React Native. It will communicate with an API built with Node.js and Express.js, hosted within a Docker container on Linux. This API will point to a PostgreSQL database, which is also hosted within a Docker container on Linux. This is summarized by the following Major Functional Component Diagram [Figure 2].

Below the required hardware and software is delineated:

- Required Hardware
    - Client App: Windows 10/Mac/Chromebook
    - API: Amazon Web Services' S3, Docker, and Kubernetes
- Software
    - 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.
        - We will be developing a React Native (Android/iOS) application to do the following…
            - View Live Streams/Recordings
            - Record/Upload Live Streams
            - Download/Upload Assignments for Teachers and Students
            - Manage Classes
            - Assign/Upload Grades for Assignments
    - API
        - Will communicate via the HTTP protocol as a REST API
        - Be built with Node JS, Express, Postgres
        - Save videos to the database
        - Be a gatekeeper to a user who would like to download a video from the database
        - Upload/Download Assignments
        - Get assignment lists as a priority Queue
        - Handle user authentication
        - Send Push notifications

Based on the technology above, Inclusive Classroom with support the following features and capabilities:

- 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 their assignments and submit to them before the due date.
- Students will be able to watch a lecture live or watch the recording after.

We can envision student interaction with the following Student Site Map [Figure 1] below, and the teacher interaction with the following Teacher Site Map [Figure 2] below.

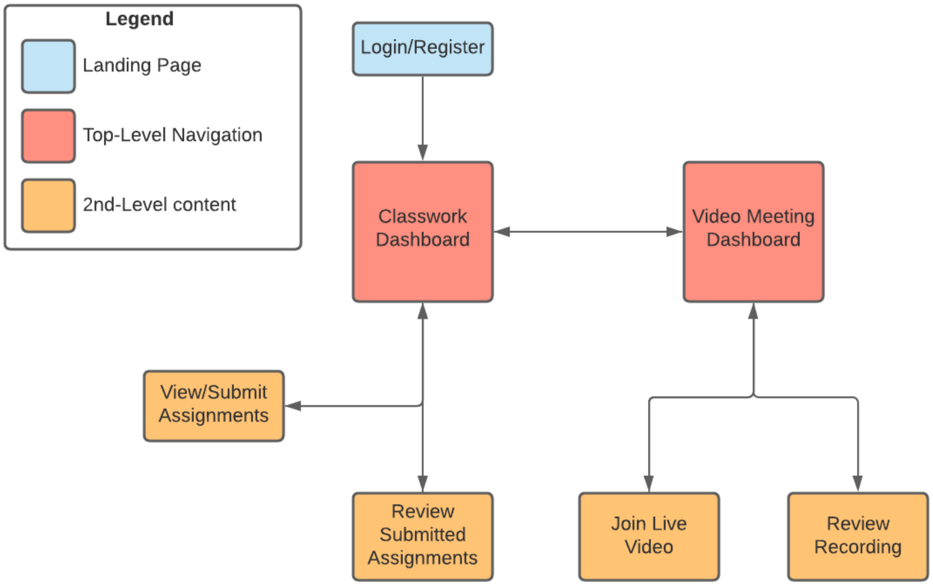

Figure 1 – Student Site Map

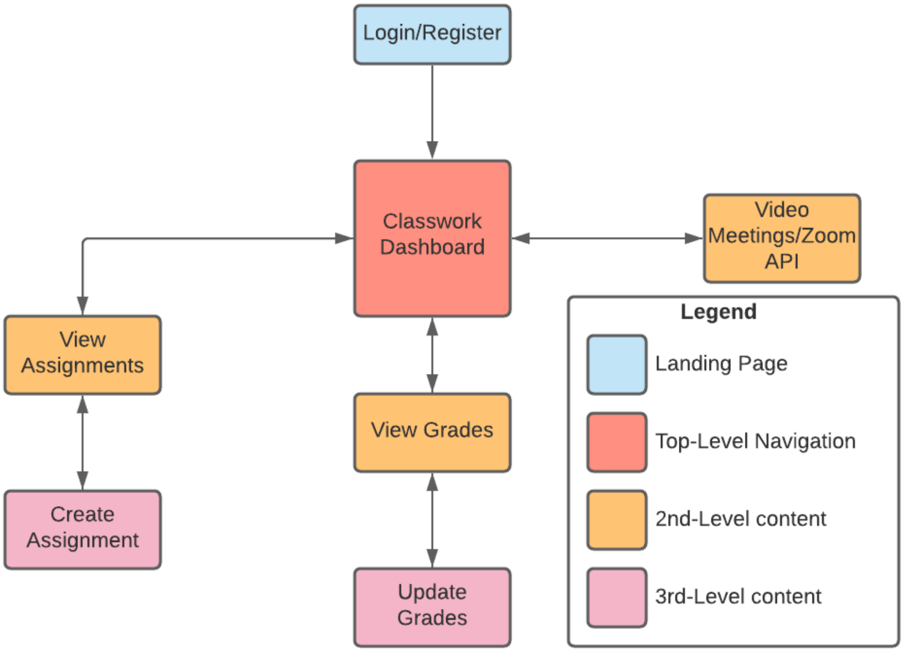

Figure 2 - Teacher Site Map

2 General Description

2.1 Prototype Architecture Description

We can initially summarize the main pieces of the software with our Major Functional Component Diagram [Figure 3] below:

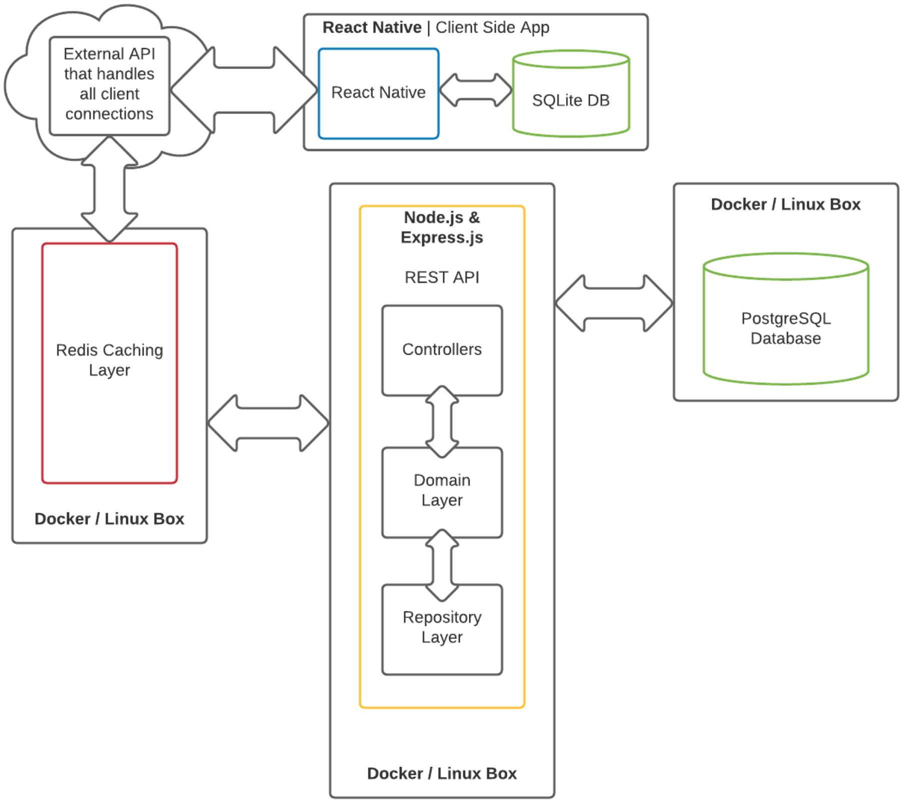

Figure 3 - Major Functional Component Diagram

We can split the architecture into the following pieces:

- Database
    - The database will be a PostgreSQL database. The database for our concept will be within Linux, but provided with a professional team, the database will become an entire database layer. It will scale according to the needs of the incoming traffic. Overall, the database layer will hold all data for the system in a relational database style. It will also be responsible for holding the usernames, emails, and passwords for user authentication.
- API
    - The API will be a REST API accessed via the HTTP protocol. The main function of the API is to provide an interface for the client app, so it can successfully

communicate with the database safely. The API will contain business logic, will sanitize input, and will reject bad queries.

- Redis Cache
    - The Redis Caching Layer will be the first layer between the client and the API. The Redis Caching Layer and API are hidden behind the cloud in Figure 3, as the client will not see these as different. The Redis Caching Layer will cache the most frequent calls and responses, helping reduce the load on the API and ensuring the client app runs as fast as possible.
- Client
    - Finally, the client will be written with React Native. It will rely on SQLite as a local database. This is the user interface that the users (teacher, student, parent, and school admin) will interact with. It will be the layer on the OS, handling all initial input, local internet issues (such as caching submitted assignments until the user has internet), and more.

[This space is intentionally left blank]

2.2 Prototype Functional Description

We can easily see how the prototype will function with the following flows. These flows will explain the logic and will summarize the functions based on which part of the client app a user is in.

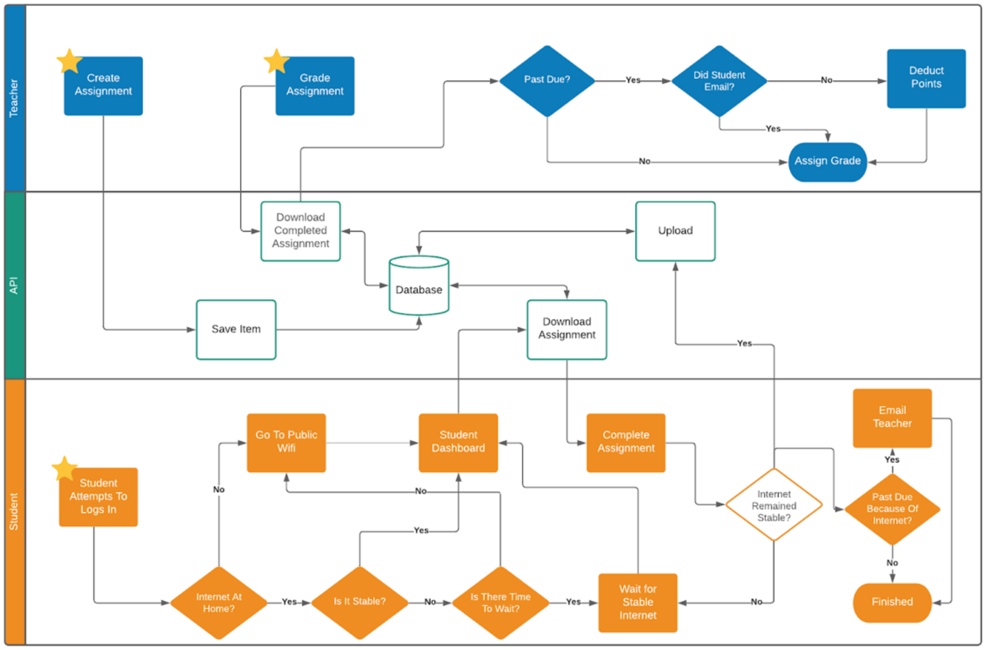

Figure 4 - Assignment Process Flow

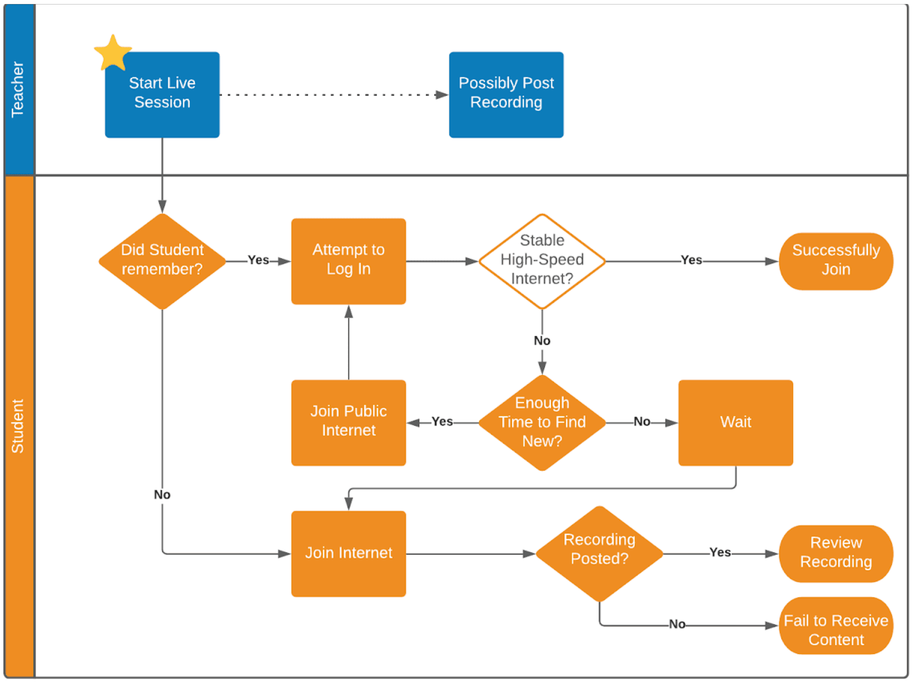

Figure 5 - Live Video Process Flow

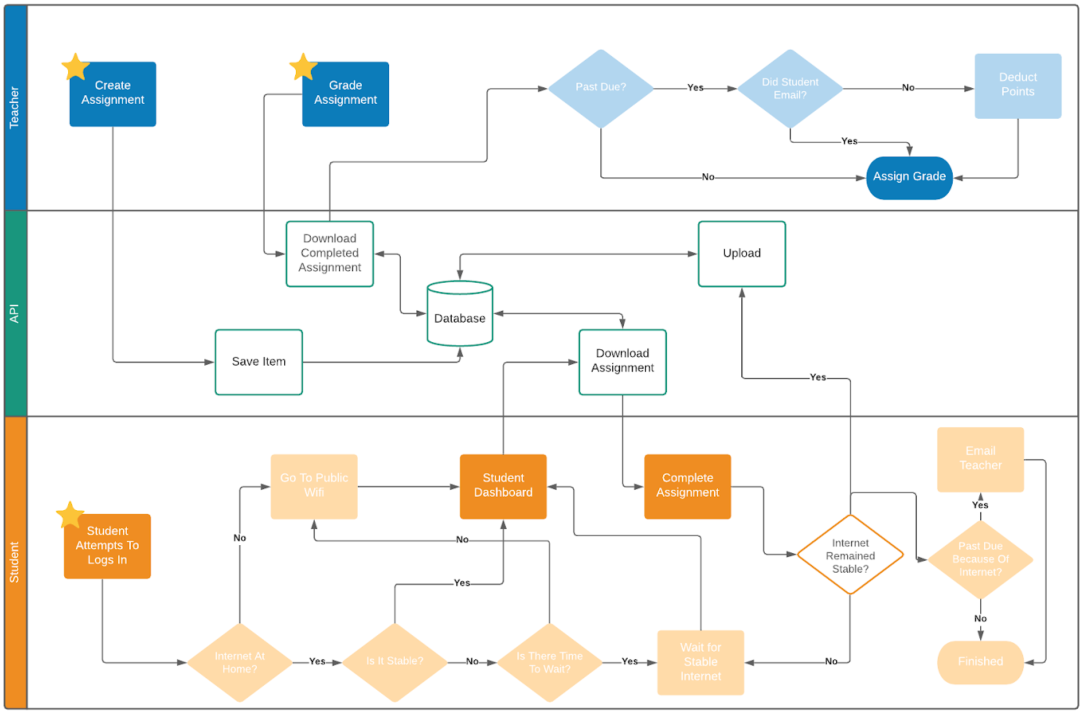

Figure 6 - Current Process Flow

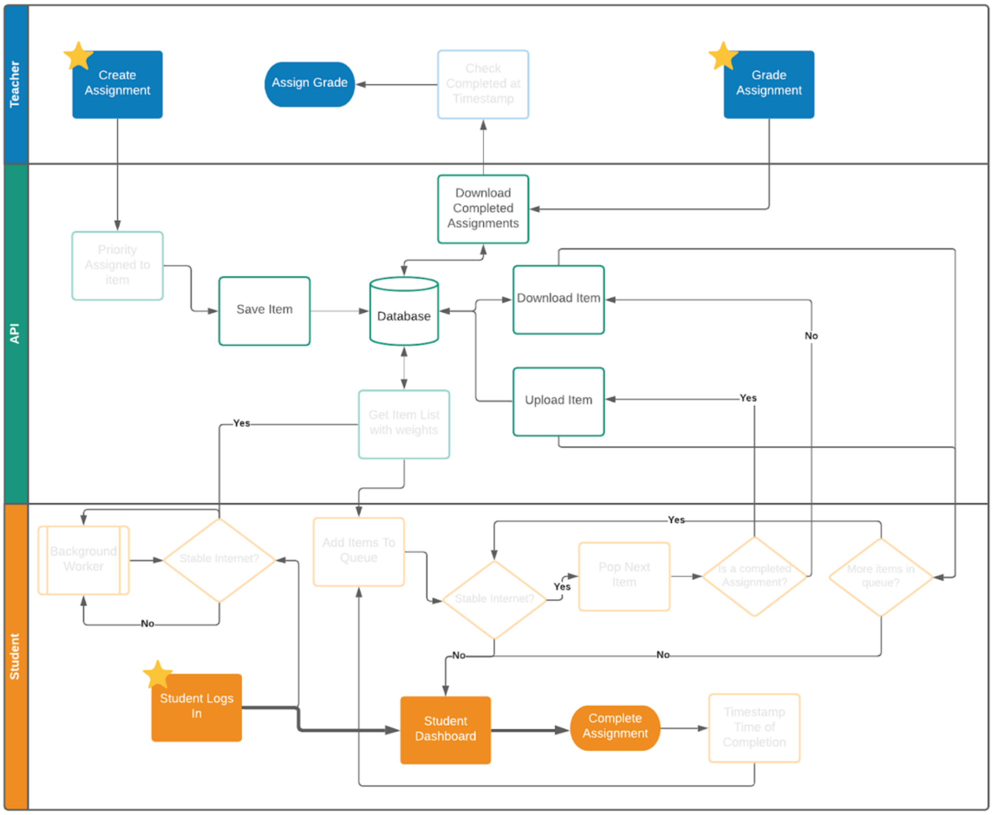

Figure 7 - Solution Process Flow

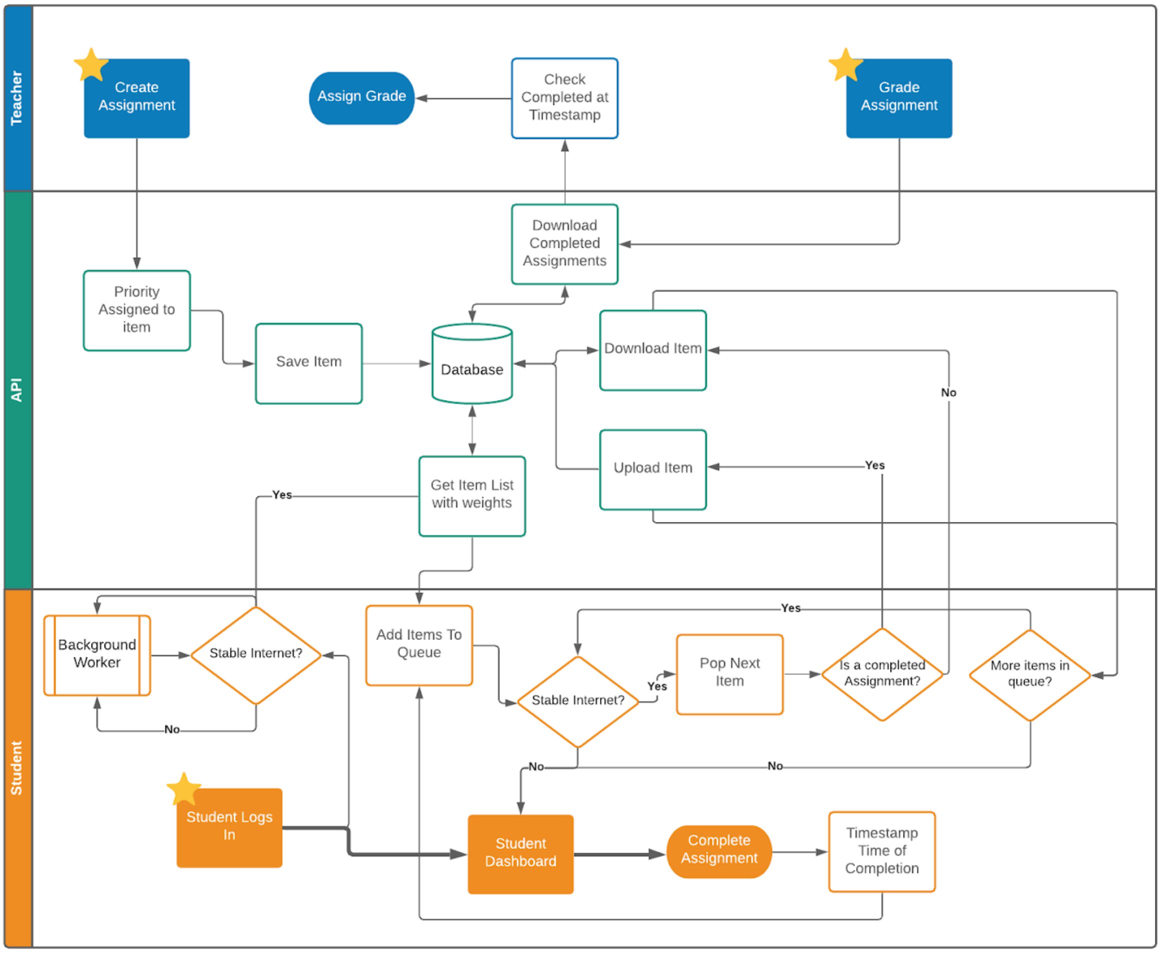

Figure 8 - Algorithms Process Flow

[This space is intentionally left blank]

2.3 External Interfaces

- No external interfaces will be needed, as the product is a software only solution.

2.3.1 Hardware Interfaces

- No hardware interfaces will be needed, as the product is a software only solution.

2.3.2 Software Interfaces

- The API will communicate via the HTTP protocol.
- The client application will communicate with the API via the HTTP protocol.
- The client application will run natively (with compiled code) on the user's operating system (OS). This will vary by the OS type, as mentioned and delineated above.

2.3.3 User Interfaces

- The user will work with a mouse and keyboard with their machine and operating system to interact with the software.
- The hardware may vary, and as long as the hardware will run the following operating systems, the software will be supported:
    - Windows
    - Mac OS
    - Linux (specifically the Ubuntu distro will be supported)

2.3.4 Communications Protocols and Interfaces

- Protocols and interfaces used will be TCP/IP via 100mb Ethernet, or wireless internet specified by the IEEE 802.3/802.11 standard. This should be covered by the hardware, on which the software will be installed.
- As our product is a software only solution, the school system will be required to assist with hardware related issues. However, our software is built to aid in many different circumstances where hardware may become an issue, such as when internet is not present.