Team DigiLearn

Requirements Specification:

The Digital Backpack

Doctor Morgan Vigil-Hayes

Volodymyr Saruta

Caitlin Abuel

Grace Shirey

Israel Bermudes

Kristine Hermosado

Sebastian Kastrul

11 November, 2020

Accepted as baseline requirements for the project:

Client Signature: _____ **Date:** _____

Team Signature: _____ **Date:** _____

Team DigiLearn

Table of Contents

Team DigiLearn

1.0 - Introduction

The COVID-19 pandemic has led to a sudden shift to remote learning. Unfortunately, many students across America don't have access to a reliable Internet connection. The phenomenon known as “the homework gap” affects nearly 12 million students that are unable to fully participate in their coursework due to a lack of sufficient Internet access. Such a situation disproportionately affects disenfranchised communities. These students must rely on public hotspots to complete their assignments.

Dr. Vigil-Hayes runs the Community aware Networks & Information Systems Laboratory (CANIS) in the School of Informatics, Computing, and Cyber Systems at Northern Arizona University. CANIS Lab focuses on network analysis and community-centered design. Team DigiLearn is working with Dr. Vigil Hayes and CANIS labs to bring to life The Digital Backpack. The Digital Backpack or DigiPack is an app that will allow a fluid transition between online and offline learning. When a user comes into range of a wifi connection, the DigiPack will automatically download the requested content for offline use later. The app will also automatically upload completed assignments for the user. These upload and download requests can be queued offline to be performed when a network connection is available. The app will interface with popular Learning Management Systems such as Google Classroom.

This document serves to establish the major requirements necessary for the usability of this product. Section 1.1 further elaborates on the problem statement, followed by the solution vision in Section 1.2. Section 2.0 describes the main requirements of the system, divided into subsections for functional requirements, performance requirements, and environmental requirements. Section 3.0 discusses the potential risks associated with the project. The Team’s project plan is outlined in Section 4.0. Lastly, Section 5.0 concludes the requirements document, with a review of the overall plan for the system. References to studies and other research can be found in Appendix A at the end of this document.

1.1 Problem Statement

CANIS lab at NAU focuses primarily on network analysis and community-centered design. One of the cornerstone projects of CANIS is PuebloConnect, which seeks to expand internet access to traditionally underserved communities through innovative network architectures. The project focuses on hardware solutions to expand network connectivity. The Digital Backpack

project, envisioned by Dr. Vigil-Hayes, focuses on potential software solutions to expand connectivity.

Variable or prolonged delays in connection can be common in areas with poor networking infrastructure. Typical network communication relies on a complete end-to-end path between two connections. This may not always be the case in rural areas, as unpredictable disruptions can occur. Because of this, students often utilize public hotspots to complete their assignments. Still, time is limited when using these hotspots, and a student may spend a lot of that time trying to remember what tasks need to be done while a connection is available.

For this project, the main problems to be addressed are as follows:

Connectivity - In rural areas, where network connectivity may be unreliable, data transfer suffers from problems such as intermittent connectivity, variable delays, and high error rates. The Digital Backpack must be able to accommodate for these connection issues using software to supplement the available network coverage.

Transitioning Between Online and Offline Use - To further aid the transition from online to offline learning, the Digital Backpack must be able to automatically queue upload and download tasks for the user. Automating the relationship between remote students and web-based learning content will provide the student with more time dedicated to learning that might have otherwise been spent trying to organize these tasks themselves.

Integrating with Third-Party Applications - Being able to integrate with 3rd party applications is a crucial part of the Digital Backpack system. This project will focus primarily on integrating with Google Classroom as well as Google Search. Google Classroom has been one of the most widely adopted Learning Management Systems in the United States following the switch to remote learning. Many students and teachers could benefit from this assistive integration.

Accessibility - The Digital Backpack must be accessible to accommodate a wide variety of students. Accessibility can expand to many facets regarding the design of the system. The application will target students that may be of various ages as well as backgrounds. It is important that the interface is designed with this in mind. Additionally, the application should be as widely available on as many platforms as possible.

1.2 Solution Vision

The Digital Backpack is a software solution to the issue of unreliable network connection. The Digital Backpack is a system that consists of two main components. The first component, the DigiPack app, is a user-facing application for Android, iOS and the web. The DigiPack app will enable the users, students, to interact with scholastic materials seamlessly between online and

offline environments. It is the second component of the Digital Backpack, the proxy server, that enables the DigiPack application. The primary purpose of the proxy server is to act on the behalf of the user while the DigiPack is offline. During this time, the DigiPack will complete such operations as resolving Google search requests, retrieving education materials from third party sources such as Google Classroom, and preparing these materials for transfer to the DigiPack application.

Consider the following high-level diagram of the Digital Backpack system architecture in Figure 1.1.

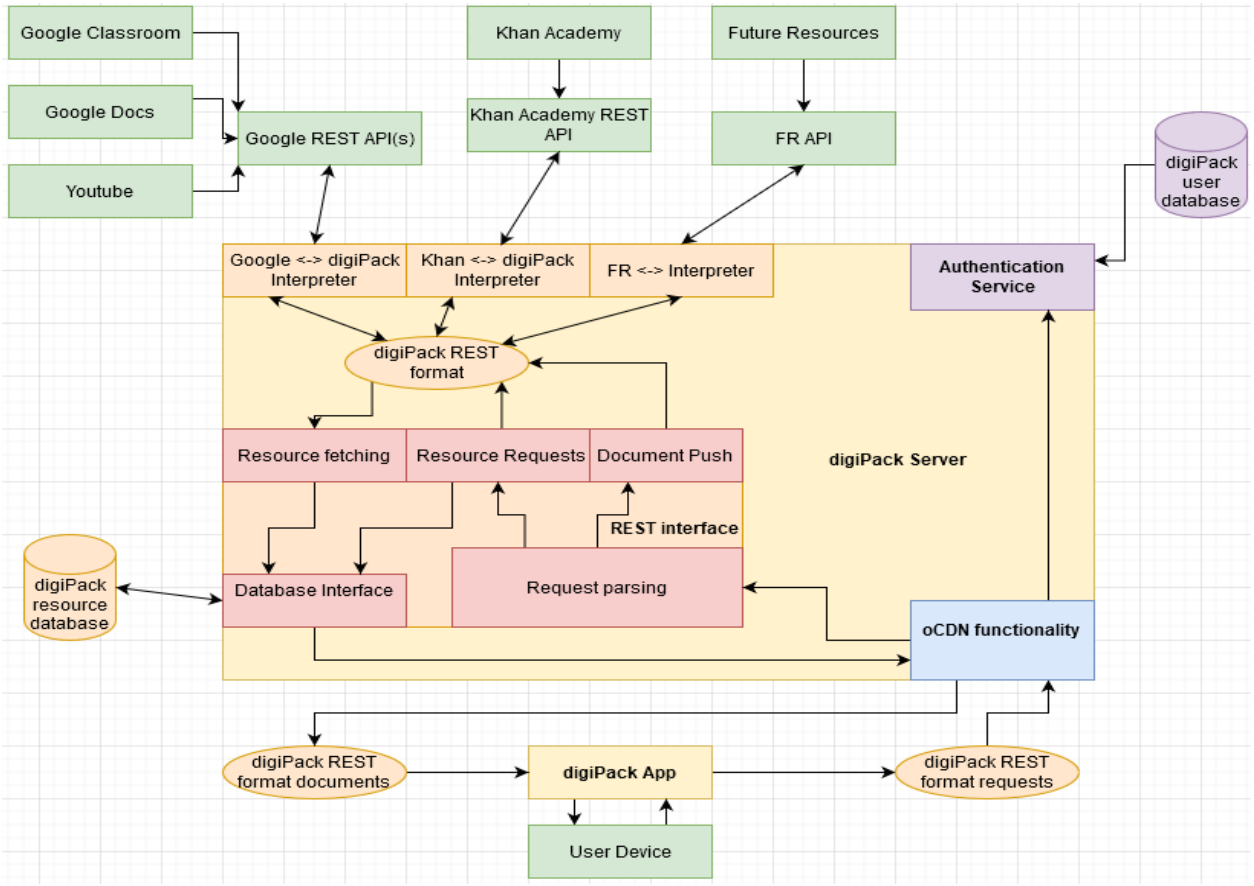

Figure 1.1: Digital Backpack System Diagram

In the following subsections, the way that this architecture addresses the main problems detailed in Section 1.1 is clarified:

Connectivity - As illustrated in Section 1.1, the proxy server leverages the functionality of an opportunistic Content Delivery Network to deliver and receive content over an unreliable connection. The key features of this connection are as follows:

Both the proxy server and the user device will prepare all files for transfer while offline to maximize the use of an opportunistic connection

If a connection is interrupted, the state of the connection will be recalled and can be resumed once the connection is reestablished.

Transitioning Between Online and Offline Use - Key to smooth transitioning between online and offline learning is the variety of actions that the proxy server takes on the behalf of the user while they are offline. Some of the key actions that the proxy server takes are as follows:

The proxy server interacts with third party services through a REST interface. This allows the proxy server to submit assignments, retrieve educational materials, resolve search queries, and more.

The proxy server prepares user materials for transfer to the DigiPack app in a queue for rapid transfer once connection is established.

Integrating with Third-Party Applications - The Digital Backpack must interface with a number of external, third-party applications. These third-party services are going to have a variety of formats for the same data. For this reason, the proxy server will implement a REST interface so that The Digital Backpack has an internal file format. This same REST interface will translate this internal file format back into the various formats used by third-party services.

Accessibility - At the core of The Digital Backpack is the need to improve internet access for as many people as possible. To this end, the DigiPack app needs to be near -universally accessible for students. As such, the app will be cross-platform developed for iOS, Android, and Chrome. In addition, multiple user interfaces will be developed to serve the needs of multiple age groups.

Team DigiLearn

2.0 - Project Requirements

This section describes the key requirements necessary to make the Digital Backpack a usable product. These requirements are broken down into functional, nonfunctional, and environmental requirements. The domain-level requirements of this project establish the overall goals of the Digital Backpack. These are listed as follows:

- The Digital Backpack will aid remote education.
- The Digital Backpack will provide seamless transitions between online and offline functionality.
- The Digital Backpack will utilize oCDN functionality in order to facilitate asynchronous connections and provide a delay-tolerant network.
- The Digital Backpack will support three types of end users: students, teachers, and parents.
- The Digital Backpack will be highly accessible in both design and device requirements in order to accommodate a wide range of users.
- The Digital Backpack will securely handle all sensitive personal and scholastic user data to prevent unauthorized access.

2.1 Functional Requirements

The functional requirements describe the features of the Digital Backpack system and it's requirements for the user.

F.R.1: The Digital Backpack will implement a proxy to work on the user's behalf.

F.R.2: The Digital Backpack will display downloaded content for use offline.

F.R.3: The Digital Backpack will interface with Google Classroom and Google Search.

F.R.4: The Digital Backpack will require user login or create a new profile.

F.R.5: The Digital Backpack will manage sensitive data and prevent unauthorized access.

F.R.6: The Digital Backpack will authenticate all user actions.

F.R.7: The Digital Backpack will have a configurable upload and download queue.

F.R.8: The Digital Backpack will have a domain filtering system for teachers and parents.

F.R.9: The Digital Backpack will use a uniform RESTful format for storage & transfer of data.

F.R.10: The Digital Backpack will be available on Android, iOS, and web browsers.

F.R.11: The Digital Backpack will have similar performance on each platform.

F.R.12: The Digital Backpack will have multiple easy to use interfaces for different ages.

F.R.13: The Digital Backpack will manage downloaded content storage on the user's device.

F.R.1: A proxy to work on the user's behalf

The proxy server in the Digital Backpack System will act on behalf of the user. When they are offline, the server will begin pulling data that was requested when the user was online. When a connection is finally achieved, the requested data that is ready will be pushed to the user's device. At the same time, the user's device will push data such as new search query requests and completed assignments. During the connection time, the user's device will communicate with the server, sharing information on the last connection session. This is to determine whether the data that was transferred in the previous session was successful. If the transfer was incomplete, the server will push the missing pieces to the user.

F.R.2: Display downloaded content for use offline

The main feature of the application is to be able to be used even offline. Students must be able to navigate through the application to access the downloaded content without an internet connection and must be able to view that downloaded content so students can work on them. Using the data transfer described on F.R.9.3, the Digital Backpack must be able to show the downloaded content even if it is offline.

F.R.3: Interface with requested services

In combination with F.R.9, RESTful libraries provided by the services requested by the client (Google Search, Google Classroom, Google Docs, and Khan Academy) will be used to provide delay-tolerant connectivity with said services. These interfaces will allow the Digital Backpack service to interact with these services on behalf of the users, downloading content provided or requested by teachers, uploading content created by the students such as completed assignments, test answers, and search queries.

The Digital Backpack server will use a collection of REST interpreters, a set of programs that are specific to each resource mentioned above, that will make requests to each resource and take the data sent back to the server and convert it to the local format described in F.R.9. Because of the wide range of data and content that each resource could provide for each user, interpreters for each individual resource are necessary to ensure that the transition from their format to the Digital Backpack server format is seamless.

This interface, along with the supporting architecture described by F.R.9, will form the core of the Digital Backpack service.

F.R.4: User login and creating a new profile

The application DigiPack is offered to individual users as everyone has a different level of education needs. For users to use the mobile application, they are required to make an account for themselves. As the account will hold personal information, the account will be protected by a password and is limited to unauthorized users.

- **Creating a new account:**
    - ◊ **Age verification:** At the start of the application, the users would be asked about their age as it determines if that user needs to have a parental supervision account. The user will need a parental supervision account if they are under the age of 14. It would also help determine the type of interface they would get.
    - ◊ **User information:** After determining the age, the user would be asked to fill in the account information such as first and last name, username, email, password, password confirmation, and a security question as authentication.
- **User login:**
    - ◊ When the user login into the application, the application would ask the user to input their username and password. More detail about the authentication on F.R.9.

F.R.5: Manage sensitive user data and prevent unauthorized access

The protection of scholastic and personal data of users is an ethical and legal responsibility of the Digital Backpack. As the Digital Backpack system can be broken down into three major components, so too can the security requirements of the system be broken down into the same three major categories. These categories, and their associated requirements, are as follows:

- **Security on the proxy server**
    - ◊ The proxy server will implement the Uncomplicated Firewall (UFW) as an additional layer of protection
    - ◊ User data that is stored on the proxy server will be encrypted using AES encryption.
    - ◊ The proxy server will be supplied encryption keys at runtime and hold these keys in active memory so that they are never written. This will decrease vulnerability to unauthorized access.
    - ◊ Encryption will be handled autonomously by the server with no need or permission for user interference.
- **Security on the DigiPack app**
    - ◊ User data that is stored on the user device will be encrypted using AES encryption.
    - ◊ The mobile app will store encryption keys and hashed passwords in the secure storage environments provided by iOS and Android. Encryption keys for the browser app will be password protected.
- **Security over network connections**
    - ◊ Communication over the network will be implemented using https with TLS.
    - ◊ Highly sensitive information, such as authentication tokens or keys, will be additionally encrypted before transmission and decrypted upon receipt.

F.R.6: Authentication

All connections between devices in the Digital Backpack system will be authenticated. Furthermore, communications with external services such as Google classroom will also be authenticated.

Connections between the DigiPack app, the proxy server, and third-party sources will be authenticated using OAuth2 authentication tokens. These authentication tokens will be initially exchanged over the network and then stored locally for repeat use. The use of these tokens will prevent any unauthorized device from communicating with the Digital Backpack system. This will also prevent unauthorized communication within the Digital Backpack system, such as the DigiPack app communicating directly with third party services.

User access to the DigiPack app itself will be authenticated using a username and password. These credentials will be required every time the user opens the DigiPack app on any device. The user may manage this password through any mechanism that they see fit. In the case of the mobile app, this password will be compared to a hashed password stored in the device's secure storage environment. In the case of the web app, the hashed password will be stored locally.

F.R.7: Configurable upload and download queue

The Digital Backpack works on the user's behalf while they are offline, automatically queueing requested content to be uploaded or downloaded when connectivity is established. The default setting for the queue will be First In First Out. The download queue should be configurable in the user preferences.

Different queue configurations include:

- First In First Out
- Smallest Files given priority
- Largest Files given priority
- Soonest due date given priority

Access to configuring the queue settings will vary based on the age-appropriate interfaces as detailed in F.R.12. The young (6-12 age) and mid-range (13-14/15) queue settings will be limited and require parental guidance. This is to prevent making accidental changes as younger users may not understand the purpose of the queue. Older (15/18) age range users will have full access to the queue settings that they may change as they see fit.

F.R.8: Domain filtering system for teachers and parents on student search queries

The Digital Backpack must be able to integrate with Google Search. These search results can be highly variable and there must be a way to limit the content that is shown. A configurable filtering system will be available on the teacher and parent interface for the application. This will allow them to whitelist or blacklist certain domains, as well as phrases. This will be done by utilizing Google Search operators and appending the requested features onto a student's search query. The filtering interface will be abstracted for end users. The specific search operators that will be implemented for the filtering are as follows:

Google Search Operator	**Function**	**Interface Abstraction**
-	Exclude terms from search results. *(E.g. jaguar -car)*	Blacklisted Words
-site:	Exclude specific websites or domains from search results.	Blacklisted Websites Blacklisted Domains
site:	Search within specific websites. (E.g. site:.gov or site:khanacademy.org)	Whitelisted Websites* Whitelisted Domains

Table FR.8: Search Operators for Use in Filtering Interface

**Because the "site:" operator will exclusively search for the entered site, a warning will be displayed to the user, informing them how whitelisting websites will limit websites.*

F.R.9: Uniform RESTful format for storage and transfer of data

Within the functionality of the Digital Backpack proxy server, a uniform REST format will be used to store, organize, and distribute data pulled from the requested resources to users.

F.R.9.1: Uniform Format

Using the interface described in F.R.3 content will be pulled from the services and converted to a local (to the Digital Backpack server) format that will contain metadata about the content. This metadata will include but is not limited to:

- ◊ Date the content was acquired by the server
- ◊ What type of content was acquired (file type, generic type)
- ◊ Size of the content (in bytes)
- ◊ Where the content is stored on the server
- ◊ What classrooms (groups of students), individual students, teachers, or topics the content is associated with
- ◊ When the content expires or should be updated

This local format will be a json file containing all of the metadata that will form the basis for the database.

F.R.9.2: Storage

Using data pulled from Google Classroom the database will be populated with tables describing the classroom structure (what teacher, what students are in the class, school name, etc.) and, once content is added by the teachers or requested by students, how that content is related to either whole classrooms or individual students. The tables describing content will also include where that content is stored on the server.

Databases can only store text based data, therefore content will be stored in an MBR file structure (for compatibility) and the filepath associated with the content will be stored in the database to be referenced when needed.

F.R.9.3: Transfer of Data

F.R.9.2 will allow the Digital Backpack server to be developed

independently of the Digital Backpack user interfaces and allow for each interface to modify the content locally (on the user device) as needed to fit device specific requirements.

The transfer of data will be handled by the functionality described by F.R.1, however the data transferred to and from the users will be contained in the format described by F.R.9.1. This agreement or protocol between the application(s) and the server will allow for all of the necessary metadata to be transferred without need for the need for multiple protocols for the large range of devices that could be used to connect to the Digital Backpack server.

F.R.10: Android, iOS, and chrome browser

The Digital Backpack application needs to be able to run on Android, iOS, and chrome browser. Flutter includes a widget using their language, Dart, that allows the developers to integrate views on each platform. There will be a back-end and front-end side for the chrome browser, Android, and iOS, and then the widget that detects which platform the user is using, and adapts to that platform.

F.R.11: Comparable performance on each platform

While the Digital Backpack will be using Flutter to adapt to multiple platforms, the architecture for each of those platforms as well as the way they both run, needs to be similar. Using the Dart language Flutter provides for the back-end, the Digital Backpack must not be more advanced on one platform than another, since each platform will be coded separately. On the front-end, the UI for each platform must look as similar and simple as possible to not be confusing for the users.

F.R.12: Interfaces that are easy to use for different ages

One type of end-users of the Digital Backpack will be K-12 students with varying ages ranging from 6-18. Research states that each age group has different behavior, physical, cognitive capabilities, and expectations. Three different user interfaces will be developed to account for the variety of end-users (will also include a tutorial guide throughout the interface), which are:

- **Young (6-12 age):**
    - ◊ As the number of times children using mobile devices increases,

these users would have some experience interacting with mobile devices or tablets. The interface would be designed to the user expectations of having high interactivity [6].

- **Mid-range (13-14/15): (A tricky age-range)**
    - ◊ As the younger rage grows, the experience using mobile devices gets higher, these user expectations would vary on the exposure and experience. The interface would be designed with a mixture of some interactivity and mature design (search bar).
- **Older (15/18):**
    - ◊ In this age range, users like an adult are goal-oriented when it comes to using websites. However, research done by NN/g stated teens are not tech-savvy as stereotypes suggested. The study indicated that teens have poor reading skills, less sophisticated research strategies, and a lower level of patience. The interface would be designed with age-appropriate content and neutral graphics [6].

F.R.13: Manage downloaded content storage on the user's device

The Digital Backpack is responsible for downloading content onto the user's device. Resources for older homework assignments may be unnecessary after that assignment is turned in. To ensure a user does not run out of storage space on their device, the Digital Backpack must manage previously downloaded content. This content will be given an expiration date, which will vary based on the type of content. For example, a downloaded assignment that has been turned in will expire after a short time (1-2 days). Other content, like content pulled from a Google search, may have a longer expiration date. This is because a student may need that content for a prolonged time if it pertains to a specific unit being taught.

2.2 Performance Requirements

The performance requirements describe the metrics in which the application is expected to perform. The following subsections will describe the measurable standards for the functional requirements outlined in section 2.1.

P.R.1: A proxy to work on the user's behalf

An important performance metric for the proxy server is its overall efficiency for performing upload and download tasks for the user. By reducing the number of connections that need to be made by the user device and the physical distance between the Digital Backpack server and the user device, the time needed for uploading or downloading data can be reduced by half the amount of time in comparison to the amount of time needed to upload or download without prefetched data. A study done at the Nara Institute of Science and Technology suggests that data that is prefetched can be accessed around 1.5 times faster than cached data [4].

- **Target Metric:** Download time as a ratio of prefetched data vs not prefetched.
- **Benchmark Value:** Prefetched data should be retrieved 1.5x faster than cached data
- **Testing Method:** Sample data will be downloaded and timed on two devices using the same network. One device will download the data through the Digital Backpack server.

P.R.2: Display downloaded content for use offline

As most of the system users will be K-12 students, they have different attention spans of various ages. In research conducted by Common Sense Media, they found in 2017 that children between the age of 0-8 had a mobile device and that 45% of that age group own their tablet. Assuming that the users were already tech-savvy and exposed to some mobile applications, the downloaded contents should be displayed on the screen in less than 1 seconds. It gives users a sense of control over the application [6]. The target metric would be:

- **Target Metric:** The latency of the application when the user goes to check the downloaded content.
- **Benchmark Value:** The downloaded content should display around 1 second.
- **Testing Method:** Once the application is built, the application would be downloaded in two phones and tested by creating accounts and requesting contents to be downloaded.

P.R.3: Interface with requested services

Documents and data that will be pushed to users from each of the requested services will be downloaded and stored in the format described in F.R.9 as soon as it is available on each service or tagged by a teacher for a set of students. The performance of this functionality will be tested in combination with F.R.1 and F.R.9. Proving that the servers ability to get the data flagged for a user to them is faster than the user could acquire the same data through “traditional” means, i.e. using each services respective mobile or online applications to download and view the data on their device.

- **Target Metric:** 25% or less of the time it would take for a user to download the same content through that resource's respective app. See: P.R.9
- **Benchmark Value:** Time to open and download multiple different content types from each resource.
- **Testing Method:** Once the REST interface has been built, a testing account will be created and used to pull content of different types and sizes from each resource. The time for the server to receive it will be compared against the time to open and download the same content from each resource.

P.R.4: User login and creating a new profile

As mentioned before, the application will need to have account creation and login pages. The DigiPack is offered to three different users which are students, teachers, and parents but the application would be mostly used by students. With the idea that everyone takes time understanding account creation, the time to create an account and login is as follows:

For first-time users, 68% of users stated that having complex logins can cause the application's deletion in a study done [2]. Login creations are the main factor that contributes if the user continues using the application, so the account reaction should take less than two minutes.

Entering the required specification, such as password, email, and username should take less than one minute.

The other account requirement should take another minute.

With user account login, the user should be able to login within 40 seconds for new users and less than 20 seconds with experienced users. It could vary depending on how fast the user remembers their credentials.

- **Target Metric:** The time a user takes to input their credentials.

- **Benchmark Value:** It would vary to the user's experiences and exposure to using mobile applications.
- **Testing Method:** User testing is required, users will be timed on how long it takes them to input their credentials.

P.R.5: Manage sensitive user data and prevent unauthorized access

Within the DigiPack app, all security operations must be performed in a way that does not significantly affect the responsiveness of the app. Specifically, security must be implemented in a way that remains compliant with P.R.2.

All security must be implemented in a way that does not significantly impact the network connection. That is, there should not be a significant difference in speed between secure and unsecure transfer. The unauthorized access of encryption keys must be impossible under any circumstances.

- **Target Metric:** Transmission time as a ratio between a secured connection and an unsecured connection.
- **Benchmark Value:** As most security is not handled at the time of transmission, a near one-to-one ration between secured and unsecured connections is expected.
- **Testing Method:** Trials will be conducted where the same data will be transferred and timed over both a secured connection and an unsecured connection. These trials will occur when all devices have already completed all relevant prep-work. That is, authentication tokens have already been exchanged and any encryption has already been performed. The transmission times in both cases will be compared.

P.R.6: Authentication

Authenticating a password should take no longer than five seconds in any circumstance. This number has been selected as an acceptable time frame that will not bother the user.

- **Target Metric:** The time between when the user inputs their password and the authentication result is returned to the user.
- **Benchmark Value:** At a maximum, the password authentication process should take at most five seconds.

- **Testing Method:** Trials will be conducted where the device times from when a password is sent to when the password is successfully or unsuccessfully authenticated.

The exchange of OAuth2 tokens between any given user device and the proxy server should occur only once as to prevent redundant use of limited bandwidth. After this exchange, there should be no significant increase in network transaction time when including OAuth2 tokens.

- **Target Metric:** Transmission time as a ratio between a secured connection and an unsecured connection.
- **Benchmark Value:** As most security is not handled at the time of transmission, a near one-to-one ration between secured and unsecured connections is expected.
- **Testing Method:** Trials will be conducted where the same data will be transferred and timed over both a secured connection and an unsecured connection. These trials will occur when all devices have already completed all relevant prep-work. That is, authentication tokens have already been exchanged and any encryption has already been performed. The transmission times in both cases will be compared.

P.R.7: Configurable upload and download queue

The upload and download queue will implement several different scheduling methods. In order to test the success of the queue, the output will be examined. The expected output is defined by the style of queue that is being implemented. For a FIFO queue, it is expected that items are processed first in first out. For a smallest file priority queue, it is expected that the smallest files are output first. For a largest file priority queue, it is expected that the largest files are output first. For a due date priority queue, it is expected that all the contents associated with an assignment with the soonest due date are output first.

- **Target Metric:** The output of the queue order
- **Benchmark Value:** The tested queue output should match the expected output with 100% accuracy
- **Testing Method:** Create different target user scenarios to test queues and scheduling approaches

P.R.8: Domain filtering system for teachers and parents

The filtering system relies on Google's built-in search operators, therefore the filter's performance will be based on how easy to navigate the parent/teacher filter interface is. Since the end users for this filtering interface will be adults, there should not be a large learning curve associated with navigating the interface.

- **Target Metric:** Time it takes a user to understand the filtering interface.
- **Benchmark Value:** Depending on the user, it should take the amount of time it takes to parse the interface for the user to understand it.
- **Testing Method:** User testing is required to determine the efficiency of the filtering interface.

P.R.9: Uniform RESTful format for storage and transfer of data

While most of the user side performance testing of this requirement will be handled by the testing described in P.R.3, testing for the subsections of F.R.9 will be as follows:

P.R.9.1: Uniform Format

Once the data and associated metadata is acquired by the server via F.R.3 the "time till stored" or time it takes to convert the RESTful data and content from each service to the local format described in F.R.9.1 will be calculated for comparison with the total time to acquire and send data to a user.

P.R.9.2: Storage

Similar to P.R.9.1, the time it takes to input new data and references into the database described in F.R.9.2 and compared to the total time from acquisition on the server to getting to the end user.

P.R.9.3: Transfer of Data

This section will be broken down further into two different testing sections. The first will calculate the time it takes to pull all of the content from a certain time period for a specific user and get it queued to be sent once the user connects. The second will calculate the time needed to send that data to the user which will also be a portion of the testing for P.R.1.

As an extension of the testing described in P.R.3, the calculated times mentioned above will be compared to the total time needed to acquire

and serve the data to the user and against the time needed to perform the same operations manually through each service's respective mobile and web applications. These comparisons will be used to improve performance and efficiency where possible within the database and "back end" code supporting F.R.3 and F.R.9.

- **Target Metric:** The total time to open and download content of different types and sizes from each resource.
- **Benchmark Value:** 75% or less of the time to open and download the same content from each resource.
- **Testing Method;** Similarly to P.R.3, a testing account will be created on each resource and content will be requested, converted to the DigiLearn format, stored on the server, and then transferred to a test device. The time that this process takes will be compared to the benchmark value. In combination with P.R.3, the process described above should take at most the same amount of time as it would through each resource's respective applications, however the goal is to reduce this total time.

P.R.10: Android, iOS, and web browsers

Users should be able to use the application of multiple platforms which are Android, iOS, and web browsers. As mentioned in F.R.10, Flutter includes a widget which combines all of these platforms. When a user logs into the application, the software should be able to detect which platform the user is on, and use the code for that platform.

- **Target Metric:** Time it takes for the software to determine which platform the user is on.
- **Benchmark Value:** Varies depending on the platform the user is on, should at most be a 2 second difference between the three platforms.
- **Testing Method:** Opening the Digital Backpack will be timed on Android, iOS, and Google Chrome.

P.R.11: Comparable performance on each platform

Users that use the mobile app as well as the web app for the Digital Backpack should have a seamless experience between the two interfaces. Visually, the application will have the same format across platforms. Additionally, load times between the iOS and

Android app will have an average difference of at most 2 seconds. In a study done by Blaze Software (Blaze Software, n.d., 1), when comparing Android and iOS browser load times, Android had a medium time of 2.144 seconds vs. iOS's medium load time of 3.254 seconds.

- Target Metric: Load time in seconds
- Benchmark Value: The difference in load times between Android and Apple devices should be at most 2 seconds.
- Testing Method: Opening the Digital Backpack application will be timed on an Android device and an Apple device, then compared.

P.R.12: Interfaces that is easy to use for different ages

Each of the three interfaces will use grade-appropriate language associated with the three target age ranges. Students' time to become familiar with the interfaces would vary from the three age ranges.

- **Target Metric:** The time each target age range gets familiar with the interface design.
- **Benchmark Value:** As each age range cognitive abilities varies, the time of getting familiar with the interface would also vary.
- **Testing Method:** (This is a tricky part) Once the application is built, it would be downloaded and be tested by different users that fall under the age range (most likely with the older group).

P.R.13: Manage downloaded content storage on the user's device

To ensure that garbage files do not pile up on the user's device, it is important that the storage management fully deletes unnecessary files from the user's device. For example, if a particular downloaded resource takes up 1.2GB of memory on a user's device, then deleting that resource should free 1.2GB of memory.

- **Target Metric:** Storage in Gigabytes
- **Benchmark Value:** After deleting data from a user's device, the gigabytes of freed data should be equivalent to the amount of gigabytes that data took up
- **Testing Method:** Storage on a device will be observed, before data is downloaded, after data is downloaded, and after data is deleted.

2.3 Environmental Requirements

The environment requirement describes the constraints that were imposed on the Digital Backpack by either the client, Dr. Vigil-Hayes, or by federal laws. These limitations are as follows:

2.3.1 Federal laws

As the DigiPack is a mobile application K-12 students, the constraints are laws that protects children privacy such as:

- COPPA (Children's Online Privacy Protection Act), which protects children's privacy under the age of 13. The act manages how the application would collect and store personal information about the user. Since the Digital Backpack would require account logins, it is essential that the system is aware of what is taking from the users and does not violate the law [3].
- FERPA (Family Educational Rights and Privacy Act), which gives parents the right to have access and revise their children's (who's under the age of 18) education records and have authority over what can be disclosed about the student regarding their education record. Since the Digital Backpack would be accessing the student's grade (which is under the protection of FERPA because it is directly related to the student, which is also maintained by an institution), the system must comply with the rules set by FERPA [5].

2.3.2 Robust network usage

Due to the nature of opportunistic connections to serve users with poor internet connections, it is likely and unavoidable that the user’s connections to the proxy server will be brief, sparse, and sporadic. For this reason, these connections must be used to the fullest possible extent:

- Transmission must begin as soon as connection is established so that no time goes unused.
- Any possible work related to the transmission of data, such as queue said data or otherwise preparing it for transfer, should be handled while the user is offline.
- In the case that file transfer is interrupted, the best effort must be made to ensure that said transmission can be resumed, instead of restarted, when connection is reestablished.

Team DigiLearn

3.0 - Potential Risks

This section will explore the various potential risks associated with the usage and development of the Digital Backpack system. Each of these risks is analyzed in terms of severity, likelihood, a mitigation solution, and the feasibility of said solution. The severity of a risk refers to the impact that the risk could have on the user or on the ability of the larger Digital Backpack system to function. Severity has been assessed in terms of damage done and how easy the risk is the correct. The likelihood of a risk refers to how often the risk is expected to occur. Likelihood has been assessed strictly in terms of the possibility and frequency of the risk occurring.

Below is a table that summarizes the assessed risks associated with the Digital Backpack (Figure 3.1). The table on the following page contains descriptions of each risk that are further described later on in this section.

RISK	SEVERITY	LIKELIHOOD	MITIGATION
1. User forgets username or password	Low	Moderate	Users will be asked security questions to recover the account.
2. Student accesses inappropriate content from Google Search results	Moderate	Low	A default list of blacklisted domains and websites will be included. (Configurable by parents or teachers)
3. Data transfer between user device and server is interrupted or incomplete	High	High	When connecting to the server, the user's device will send information to the server about what data was received at the last connection and what data is missing.
4. Unauthorized individuals or software makes requests to the proxy server or push data to the DigiPack app	High	Low	All network connections within the Digital Back system will be authenticated using OAuth2 tokens. This will prevent unauthorized devices from being interacted with by the system. The proxy server will be further hardened by the Uncomplicated Firewall.
5. Unauthorized individuals may access sensitive user data stored on the proxy server or on the DigiPack app.	High	Low	All sensitive data stored on the DigiPack app or the proxy server will be encrypted. As a result, this sensitive data will be inaccessible even in the case of unauthorized access to either of these devices. In the case of the DigiPack app, decryption of these materials will require a login.
6. Content requested by a user is unable to be interpreted by the REST service(s).	High	Moderate	Thorough unit testing will be done for each of the interpreters to attempt to mitigate the possibilities of this occurring. However, in the event that it does happen an error will be logged so that amendments can be made to the interpreter.
7. Downloaded data from the DigiPack occupies too much room on the user's device, causing them to run out of storage.	High	High	Data will be sent to the user's device in a compressed format to reduce the size of the files. Additionally, data that is no longer needed will be deleted from the user's device.
8. Data transfer is interrupted by hardware failure on the user's device, such as when a mobile device's battery runs out.	High	High	Where possible, the user device will send the relevant data needed to resume transmission as described in risk 3. In cases where this is not possible, it will be necessary to discard the partial transmission and restart it the next time the device connects to the proxy server.
9. The proxy server crashes or otherwise needs to be reset	High	Moderate	Due to how the proxy server carries its encryption keys, it will need to be manually reset. Once the server has been reset and has been supplied any and all necessary keys for operation, it can resume normal operation.
10. Data on the server is unexpectedly lost	High	Moderate	Data backups will be maintained by utilizing DigitalOcean's weekly backup service.

Figure 3.1: Potential Risks Summary Table

RISK 1: User forgets username or password

SEVERITY: Low

The user would need to reset their password using their email.

LIKELIHOOD: Moderate

As the survey by HYPR stated, passwords are commonly forgotten, and that 78% of people reset their forgotten passwords in the past 90 days. A study conducted by Rutgers University and Aalto University suggests that users who create unique passwords that frequently use it will most likely remember their password, unlike users who create a simple password but rarely it.

MITIGATION: Users will be asked security questions to recover the account.

FEASIBILITY: When the user creates an account, they would be asked to pick what security questions they would want and use that as authentication when they need to recover the account.

RISK 2: Student accesses inappropriate content from Google Search results

SEVERITY: Moderate

Given that there are endless query results that can show up on a Google search, a student may come across age inappropriate material. The Digital Backpack is intended to be used as an educational tool.

LIKELIHOOD: Low

If the application is used in its intended manner, it is highly unlikely that inappropriate content will be shown by accident. There is a small possibility that a student may choose to abuse the Google Search functionality of the Digital Backpack.

MITIGATION: A default list of blacklisted domains and websites will be built into the search query filter. This list will be configurable by parents and teachers, who may add or remove domains as they see fit.

FEASIBILITY: Google Search has a list of special operators that can modify searches to exclude certain sites or phrases. To implement this, a string of the blacklisted items will be appended to the end of the student's search query. This will be done on the backend and abstracted from all front end users.

RISK 3: Data transfer between user device and server is interrupted or incomplete.

SEVERITY: High

A student may not be able to complete their assignments if they do not have all the necessary content downloaded.

LIKELIHOOD: High

The target users of the Digital Backpack are likely to live in areas with poor internet connection. Because of this it is likely that frequent interruptions will occur.

MITIGATION: When connecting to the server, the user's device will send information to the server about what data was received at the last connection and what data is missing. The server will then push the data that the user is missing to the user's device.

FEASIBILITY: Implementing a method of communication between the user's device and the Digital Backpack server should be relatively straight forward.

RISK 4: Unauthorized Network Access to the Proxy Server or DigiPack App

SEVERITY: High

Unauthorized access to the end devices could result in sensitive user data being compromised. It is for this reason that this is a high-severity risk.

LIKELIHOOD: Low

The relevant details of the proxy server and the user device are not publicly available, so this is a low-likelihood risk.

MITIGATION: All network connections within the Digital Back system will be authenticated using OAuth2 tokens. This will prevent unauthorized devices from being interacted with by the system. The proxy server will be further hardened by the Uncomplicated Firewall.

FEASIBILITY: The mitigation solution described above is standard in terms of security best-practice, so this solution should be highly feasible to implement.

RISK 5: Unauthorized Access to Sensitive User Data

SEVERITY: High

An unsecure system could lead to violation of FERPA privacy laws and the leaking of sensitive user data. For that reason, user data being accessed without authorization is a high severity risk.

LIKELIHOOD: Low

The mitigation solution proposed under Risk 4 is the first layer of protection against this sort of intrusion, so Risk 5 has a low likelihood of being relevant.

MITIGATION: All sensitive data stored on the DigiPack app or the proxy server will be encrypted. As a result, this sensitive data will be inaccessible even in the case of unauthorized access to either of these devices. In the case of the DigiPack app, decryption of these materials will require a login.

FEASIBILITY: The mitigation solution described above is standard in terms of security best-practice, so this solution should be highly feasible to implement.

RISK 6: Content requested by a user is unable to be interpreted by the REST service(s).

SEVERITY: High

Users being unable to access requested data (within the confines of P.R.8) could mean that they potentially lose part of their grade, or miss some key detail to their understanding of a subject. Errors within the interpreters could also lead to missing data, incomplete database queries, or corrupted files which could present many issues for the server and applications functionalities.

LIKELIHOOD: Moderate

Given the wide range of content and data types each resource can serve, there is a good possibility that certain instances could simply be overlooked during development or data in transit from a resource to the server could be corrupted.

MITIGATION: Thorough unit testing will be done for each of the interpreters to attempt to mitigate the possibilities of this occurring. However, in the event that it does happen an error will be logged so that amendments can be made to the interpreter.

Furthermore, each interpreter will consist of three parts, the first will ensure that the content received from a resource contains all of the information needed to be converted to the Digital Backpack format, the second will be the interpreter itself, and the third will be a checker to make sure that the Digital Backpack formatted file produced by the

interpreter has all of the information needed to be put into the database or sent to the user directly.

FEASIBILITY: Unit testing and the interpreters internal architecture described above are key steps in the development process for these interpreters. Unit testing will be done with the Python unittest library to ensure that test coverage is adequate and thorough. These unit tests will also form the basis for the first and third parts of the interpreters described in the mitigation section of this risk. Error logging will be a must for all parts of this project and will be strictly enforced by the team release manager.

RISK 7: Downloaded data from the DigiPack occupies too much room on the user's device.

SEVERITY: High

Using the entirety of the user's storage would render the app unusable. Once the device hits capacity, no more items can be downloaded.

LIKELIHOOD: High

The target users for the Digital Backpack are those that live in rural areas. It is likely that these users would not have devices with large storage space. The resources needed to complete an assignment may be a significant amount of data.

MITIGATION: Data will be sent to the user's device in a compressed format to reduce the size of the files. Additionally, data that is no longer needed will be deleted from the user's device. After a certain amount of time has passed, data that has not been used will be trashed (unless otherwise specified by the user).

FEASIBILITY: Compression is a standard practice for reducing file sizes to be transferred. Additionally, it should be relatively easy to keep track of files and delete them as necessary according to time.

RISK 8: Data transfer is interrupted by hardware failure on the user's device

SEVERITY: High

In any case where data is not reaching the user's device is a high-severity issue due to this transmission of data being at the core of the Digital Backpack's purpose.

LIKELIHOOD: High

The user will be often using the DigiPack app without a source of power. Due to

this fact, it is inevitable that there will eventually be an instance where the user's device loses power during transmission.

MITIGATION: Where possible, the user device will send the relevant data needed to resume transmission as described in risk 3. In cases where this is not possible, it will be necessary to discard the partial transmission and restart it the next time the device connects to the proxy server.

FEASIBILITY: Restarting the transmission is simple and highly feasible. Recovering the partial transmission will instead depend on the state of the transmission at the time of device shutdown, but will be possible in some circumstances.

RISK 9: The proxy server crashes or otherwise needs to be reset

SEVERITY: High

Any time that the proxy server is unavailable, all networked functionality of the Digital Backpack will be unavailable. As this is the core purpose of the Digital Backpack, this is a high-severity problem.

LIKELIHOOD: Moderate

There is no foreseen circumstance that will result in the server needing to be reset; however, when working with software it should be assumed that such a circumstance will eventually occur.

MITIGATION: Due to how the proxy server carries its encryption keys, it will need to be manually reset. Once the server has been reset and has been supplied any and all necessary keys for operation, it can resume normal operation. A team member will be informed of this outage to minimize downtime.

FEASIBILITY: Resetting the proxy server manually is a quick and simple process, so this is a highly feasible solution.

RISK 10: Data on the server is unexpectedly lost

SEVERITY: High

Loss of data on the server would disrupt the flow of the Digital Backpack system.

LIKELIHOOD: Moderate

Server failures can occur as a result of unexpected failures, so the likelihood of data loss can be unpredictable.

MITIGATION: Data backups will be maintained by utilizing DigitalOcean's weekly backup service. The most important data to backup would be user information. The loss of other data pulled from third party sites is less important because this data can simply be pulled again. Because of this, weekly backups should be sufficient, as an influx of new daily users is not expected.

FEASIBILITY: Since this service is offered by Digital Ocean, it just needs to be enabled in order to be put into place.

Team DigiLearn

4.0 - Project Plan

Section 4 discusses the plan the team has created in order to execute the project. The project plan is a key part to the software project. The plan assists the team with staying on track and advancing the project on a strict timeline. Every week the team meets numerous times to discuss the deliverables and make sure every member is on schedule and on the same page. A schedule has been created to track the progress of the project as the semester comes to an end, and the principal milestones for the second semester to complete the project.

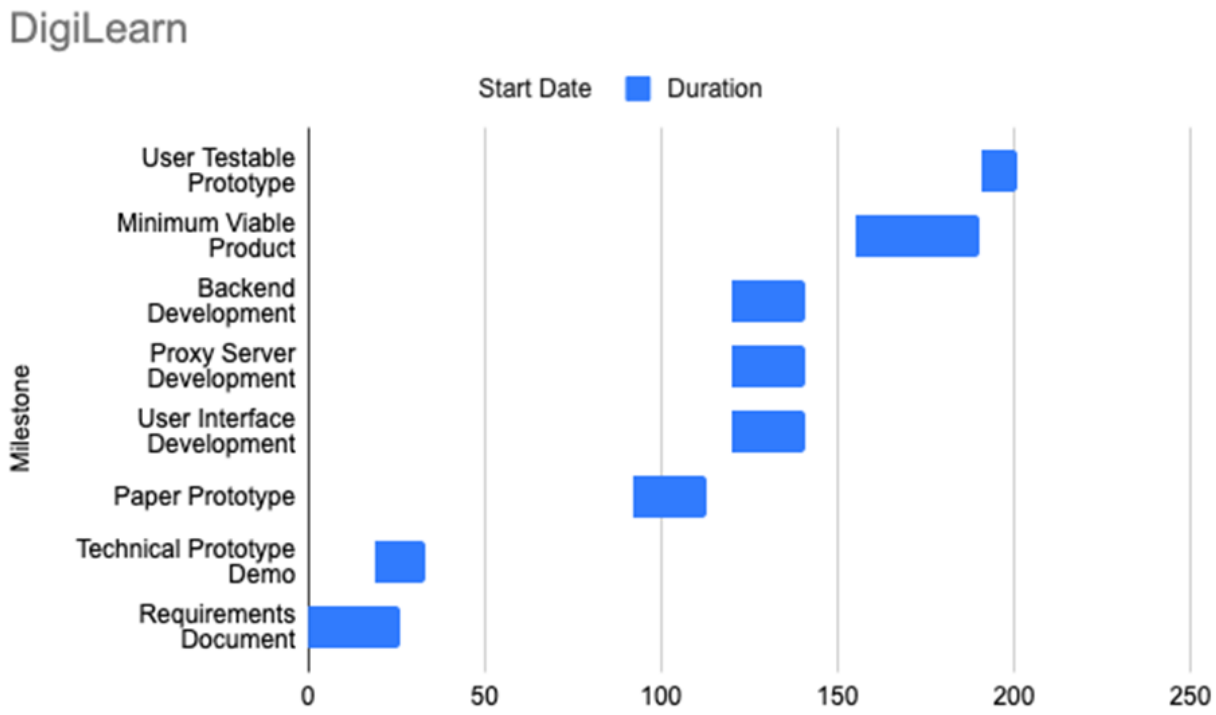

Figure 4.0: DigiLearn Spring Semester Gantt Chart

Currently, the requirements specification document is in progress, and is set to be finalized by mid November. After the document is completed, the technical prototype demo will be started. It is crucial that the technical prototype demo is finished by the end of the first semester in order to prove the feasibility and start the development of the project.

The tasks for the development of the project for the second semester have been divided into

six milestones. The first milestone starts off simple with the paper prototype of the DigiPack. By early February, a sketch of the prototype will be finalized. This will show the desirable design of the DigiPack along with the plan for constructing the back-end.

Once the prototype has been designed on paper, the software with the user interface will be built. It is important to start the user interface early on in the semester to allow time to make improvements and develop the highest quality user interface. The first iteration of the user interface will be developed by early March.

After the user interface is developed, the next key part to the project is developing the proxy server. The proxy server for this project will be used for connecting the user to the internet as well as security. The proxy server will be developed by early March.

The final key for the project to be completed as a whole is to develop the backend. The backend of the DigiPack will be developed by early March.

With the frontend, backend, and proxy server developed, the team is now able to start creating the desired project. The features desired for the DigiPack will be implemented. The minimum viable product of the DigiPack will be completed by the end of April. The minimum requirements and expectations for the project will be completed.

From the minimum viable product, the team will be adding more features, upgrading the software, and improving the product in any way possible. At this point in the plan, the focus is the final product. As the semester comes to an end, the team will be able to present our user testable prototype.

Team DigiLearn

5.0 - Conclusion

As the internet has become more and more of a necessity for students of all levels of education the “digital gap” has grown considerably. Students without regular or reliable internet access are at a significant disadvantage compared to others that are able to access the internet consistently. The Digital Backpack project aims to aid students struggling with this issue by offering an opportunistic Content Delivery Network or oCDN for educational content. The Digital Backpack application will give users the ability to download and upload homework assignments, tests, educational videos, and even search for resources to support their learning in the background, any time they are able to connect to the internet. By storing these things on the user’s device they will be able to take educational content home and still be able to participate similarly to students with constant internet access while being completely offline.

This document aimed to refine the requirements specified by the client, Dr. Morgan Vigil-Hayes, define more specific functionalities of the Digital Backpack service at a granular level, describe the overall architecture of the Digital Backpack service and the testing that will be needed to ensure that each piece functions properly, address any potential risks associated with the project as a whole, and start to plan out the development process for the second semester.

Section 1.1 and Figure 1.1 give a general overview of the architecture while Section 2.1 goes into more detail about the functionality and requirements for each of the objects in Figure 1.1. Breaking down the requirements in Section 2 has allowed the team to get a better understanding of the high level architecture of the project and how each member is key to completing a functional and useful product in the coming semester.

This document, acting as a sort of contract with the client, has not only laid out the goals of the DigiLearn team for next semester but also kick started thoughts and planning for development in the coming months.

Team DigiLearn

Appendix A: References

[1] Blaze Software. (n.d.). iPhone vs. Android – 45,000 Tests Prove Whose Browser's Faster. https://www.wired.com/images_blogs/business/2011/03/Android-vs-iPhone-F1000-Paper.pdf

[2] Brianna Miller. “First-Time User Experience: How to Keep New App Users Coming Back.” CleverTap, clevertap.com/blog/first-time-user-experience/.

[3] “Children's Online Privacy Protection Rule (‘COPPA’).” Federal Trade Commission, 6 Mar. 2020, www.ftc.gov/enforcement/rules/rulemaking-regulatory-reform-proceedings/childrens-online-privacy-protection-rule.

[4] Chinen, K. (1997). An Interactive Prefetching Proxy Server for Improvement of WWW Latency.

[5] “Family Educational Rights and Privacy Act (FERPA).” Home, US Department of Education (ED), 1 Mar. 2018, www2.ed.gov/policy/gen/guid/fpco/ferpa/index.html.

[6] World Leaders in Research-Based User Experience, Jakob Nielsen. “Powers of 10: Time Scales in User Experience.” Nielsen Norman Group, www.nngroup.com/articles/powers-of-10-time-scales-in-ux/#:~:text=Most