WHAT'S UP DOC

# Technological Feasibility Analysis

3/11/2022

## Project:

C & I Doctoral Tracking Tool

## Project Sponsor:

Gretchen McAllister

## Faculty Member:

Michael Leverington

## Team name:

What’s Up Doc

## Team Members:

Adam Larson (Lead), Brandon Shaffer, and Eddie Lipan

## Team Mentors:

Tomos Prys-Jones<br>
Anirban Chetia

# Table of Contents:

Pages

# 1.0 Introduction

Attending and graduating college constitutes some of the most formative and adversarial years of a person's life. According to data reported in 2021, 3.1 million post baccalaureate students are enrolled in graduate programs at universities in the United States. [1] Generally these students are striving to learn and better themselves each day of attendance but many will likely face similar adversities, such as timeliness, deadlines and examinations, all while balancing social lives. A vital aspect of learning is feedback. For undergraduates, this can be achieved through a simple three-step process; take an exam, turn in said exam in, and receive a grade. Feedback facilitates the process of learning from one's mistakes, a key part of personal and intellectual development.

Most undergraduate courses are implemented through Learning Management Systems (LMS) such as Blackboard Learn or Canvas, which act as course delivery and grade feedback tools. This is possible as most undergraduate courses classify as "pass/fail" or fall under some version of the standard A-F grading system. Assignments and examinations from class to class are normally quite rigid in that they pertain to lectures or chapters of an assigned reading, thus they may be outlined and planned before a semester even begins. This rigidity is absent in graduate school as graduate students perform activities that are less quantifiable, such as shadowing mentors and researching. Therefore, it is considerably harder to deliver the graduate program using a LMS.

The Coordinator of the Curriculum and Instruction (C&I) doctoral program at Northern Arizona University (NAU), Gretchen McAllister, Ph.D., and Administrative Services Assistant, Michele Benedict, have both experienced the shortcomings of Blackboard Learn firsthand. As a result, neither is able to track the milestones of their graduate students. In order to attain candidacy in the C&I Ph.D program, students must complete a minimum of 60 graduate-level course units, professional development requirements, comprehensive written and oral exams, a qualifying research paper, approval and assignment of a dissertation committee, and submission of an approved dissertation prospectus.

At present, Gretchen and Michele are having graduate students funnel all deliverables to a single computer via email which are then held locally in folders labeled with the student's name. No further organization is performed other than storing the deliverable into the appropriate students' folder. Graduate student grading and submission inquiries are achieved by emailing Michele directly, who then browses the files locally, and provides feedback via email. The current feedback process is inefficient and taxing for both parties involved.

Team What's Up Doc's solution is an intuitive website application, allowing graduate students to track their progress and receive daily motivational messages up until candidacy. The website application will also allow faculty to view student progress and create detailed reports based on their remaining milestones. Given that some students may not have a technical background, the app would require an intuitive, visually appealing dashboard making it easier to upload deliverables, track progress, all while avoiding the time-consuming correspondence with Michele. Another goal of the website application is to avoid manual data entry for either Gretchen or Michele, a responsibility that often falls to Michele currently.

The purpose of this Technological Feasibility document is to identify foreseeable technological challenges, explore possible alternatives and solutions, prove feasibility, and guide future implementation.

# 2.0 Technological Challenges

Team What’s Up Doc is in the early stages of this project and we have currently identified four key technological challenges that need to be addressed:

- **Database Management System:**
  - Currently there is a demand to store documents from the C&I doctoral program, which will be uploaded from the website application by the graduate student. Considering the current state in which all deliverables are being stored, we need to organize, populate and maintain a database system so that we may reliably call upon the necessary data when needed.
- **Authentication and User Privileges:**
  - Authentication is an important part of any application because it determines user privileges. User privileges encompass any quantifiable action that an authenticated user can make in an application. This will be a College of Education website application, therefore there is a high level of security and privacy entailed to each user. Students must only be allowed to track and view their own milestone progress, which is reached using their respective NAU credentials. Faculty must be allowed to create student dashboards, remove student dashboards and view reports on any and all students involved in the graduate program, thus there is a clear separation and authentication that must be implemented.
- **Service Framework:**
  - Service frameworks are crucial to ensuring that a website application will run properly. If a graduate student submits a document through the website application, the website app must know which database system and which student table is to be populated. When a student wants to view a graded document or access a submission, the website must make a request to the database system for the appropriate document. Frameworks facilitate this exchange of information and ensure that the data is properly populated and can be accessed later on by said requests.
- **Front End:**
  - Ultimately, our goal is to create an approachable and intuitive front end system that is easy to use and visually appealing, so that graduate students can maintain their milestone dashboard up until candidacy without causing more unnecessary stress.

The challenges highlighted above will be meticulously scoured and explored, along with any alternatives, so that each proposed solution is of sound quality and properly sets us and our client up for success.

# 3.0 Technology Analysis

As for the challenges identified in the previous section, analysis has been performed to gauge the feasibility of available technologies. Analysis requires desired characteristics of each challenge to be addressed and awarded a point value ranging from 0 to 5, with 0 being the lowest and 5 being the best. Each technological challenge requires careful consideration and consistent scoring, therefore a detailed explanation of how each point value will relate to said challenge can be found in their corresponding Desired Characteristics section. Totals are calculated and the resulting highest score determines the technology most beneficial to the product.

## 3.1 Database

Databases are common storage mediums for data that can easily be accessed to update and retrieve information. The role of it within this product is specifically to hold the completion status of students for the different milestones they have completed, as well as their role within the system and basic identifiers. Milestones are the name of the objects that need to be completed and noted with a boolean value. The basic identifiers are necessary to ensure the user is viewing the appropriate data. Finally the role field is present for noting student and administrator status, allowing varying displays and additional administrator accessibility.

### 3.1.1 Desired Characteristics

The database for this product needs to be fully accessible with ease of communication with the service framework and the ability to add or remove milestones over time. The relevant requirements are as follows:

- **Scale:** The scale of the database for the product is relative to the amount of students and milestones present. Students currently attending, previous graduates, and all upcoming students will be present within. A constantly increasing number of students, and a varying amount of milestones makes the scale of the database very important.
    - A score of 0 would show it is a fixed size and cannot be modified.
    - A score of 5 would indicate the database's ability to accommodate any amount of information necessary.
- **Relevance:** This is the impact of the database within the development community. A database that is more relevant is one that is going to have a much longer use and support life than one that is less relevant, as well as boosted connectivity options.

    - A score of 0 would go to unknown and obscure databases with little to no support and no preexisting way to connect to them.
    - A score of 5 would be for the most popular, modern and supported databases that display long term support and viability. Databases with this score also have existing module or library support that increases connectivity.

- **Implementation:** The ability to implement and construct the database to match the data, as well as knowledge and familiarity with the process of creating, retrieving and removing data.
    - Scoring of this section is subjective to the developer working on it, unlike the objectivity of previous characteristics.
    - Scoring a 0 would be no understanding of the database, how it is managed or accessed.
    - Scoring a 5 would display a strong understanding of the database type and specific previous experience managing it.

### 3.1.2 Alternatives

- **MySQL:** One of the most popular and well known relational database management systems (RDBMS). Data within the database is based on the relational model and presents it in an intuitive table and uses SQL language to retrieve data. [7]
- **MongoDB:** Non-relational database that provides enormous data storage that does not use SQL language, and instead of storing data based on the relational model (table), data is stored in a document style format. [8]
- **PostgreSQL:** Another SQL based database that provides additional functionality such as rollback, cascade functions, advanced data-types, and many others. [9]

### 3.1.3 Analysis

- **MySQL**
    - **Scale:** MySQL has the ability to scale up to any necessary size, and is only limited by the system running it. Extremely large amounts of data does slow a full view of all data, but this is easily negated by limiting the amount of data retrieved. This product could show all students, but limiting it to current or graduated students would increase speed with a large scale.
        - **Score: 4 out of 5 points**
    - **Relevance:** MySQL is arguably the most well known and supported option. Many frameworks and tutorials are available to set up quickly and the SQL language is familiar.

        - **Score: 5 out of 5 points**
    - **Implementation:** Creation of the schema, tables, and additional options are made very easy through either the mysql command line or the MySQL Workbench program. Previous usage of MySQL and the Workbench make this very easy to set up and accommodate any amount of data necessary.
        - **Score: 5 out of 5 points**
- **MongoDB**
    - **Scale:** MongoDB was a solution to storing an enormous amount of data and being able to access specific items very quickly. It is easily the best example of large scalability and would provide the best option to store a significant amount of data.
        - **Score: 5 out of 5 points**
    - **Relevance:** A very popular option with plenty of community support. Many other proprietary options, but there is a community supported option with many modules and guides on set up.
        - **Score: 5 out of 5 points**
    - **Implementation:** Unfamiliarity with MongoDB hurts its ability to quickly be implemented. There are tutorials and plenty of documentation available through the mongodb website, but no previous usage would make it less desirable for a short turnaround product.
        - **Score: 2 out of 5 points**
- **PostgreSQL**
    - **Scale:** PostgreSQL is identical to MySQL in terms of scale, as both are relational and size can be what it needs to be but will hurt the response time if it is too large or too much data is being returned.
        - **Score: 4 out of 5 points**
    - **Relevance:** While not irrelevant, this is not one of the commonly mentioned databases for non-commercial use. Usage would be similar to MySQL, but with considerably more features. Advanced data-types, rollback, cascading functional changes, and many of the other options would not be useful for this product.
        - **Score: 3 out of 5 points**
    - **Implementation:** Similar to MongoDB, no experience with this affects its ability to be implemented quickly. Creation of the Schema and tables are very similar to MySQL, however additional options create some confusion with declarations while providing no benefit over primitive types.
        - **Score: 2 out of 5 points**

**Table 3.1.3 Database Analysis**

| | Scale | Relevance | Implementation | Avg. Score |
|---|---|---|---|---|
| MySQL | 4 | 5 | 5 | 4.7 |
| MongoDB | 5 | 5 | 2 | 4.0 |
| PostgreSQL | 4 | 3 | 2 | 3.0 |

### 3.1.4 Selected Approach

As indicated in table 3.1.3, MySQL proves to be the best choice of database technology. This is due to its ability to scale to any size necessary, and the speed of data table retrieval is largely negligible for a product of this scale. It is one of the most popular open source databases with a wide range of support, both available online and professionally through Oracle. The implementation of MySQL is what sealed it in terms of usage. MySQL has been used for multiple applications by the developer in charge of implementing it into the project. This greatly reduces turnaround time as no additional language, platform, or system connection are required.

### 3.1.5 Feasibility

Noted above was previous experience with MySQL, and how it was chosen for use. A great amount of experience with it, both within the computer science department and personal use, gives us the edge to implement it well and properly. Local deployment for testing and connectivity with other relevant technologies will be done through a simple python script, providing the ability to test

## 3.2 Authentication and User Privileges

Authentication of users is an important part of any program that deals with user data and access. It uses user credentials and tokens to provide limited access to the database and only return data that that person should have access to. Proper authentication allows specific users access to information that only they have access to. In this case, students are the primary users with administrators having the ability to update information and access student information necessary, as well as the ability to download and update local records. Differentiating these two groups is very important to the integrity of the product, as well as filtering out those not a part of either.

### 3.2.1 Desired Characteristics

The desired product will need to use the NAU login information to provide the graduate students with access to the web app providing seamless access to all school related information. The relevant requirements are as follows:

- **Documentation:** Proper documentation will be required to set up and authenticate web tokens. This is necessary as there are many ways to produce these tokens. Consistent, verifiable access is crucial to protecting student data.
    - A 0 score would indicate a closed system, with no documentation available and no way to connect.
    - Scoring a 5 in this category indicates significant documentation is available, documentation is clear and concise, and provides developers additional source code to connect more easily.
- **Implementation:** Implementation of the product includes how easy it will be to integrate using provided documentation. This includes processing web tokens and providing accurate user information to other systems.
    - Primarily subjective based on developer's understanding of the documentation.
    - Scoring a 0 would be no understanding of the API or token processing.
    - A score of 5 would indicate a clear understanding of how to integrate the API, token contents and necessary data to be collected from them.
- **Connectivity:** Connectivity pertains to the ability to connect to the users NAU accounts and use their normal login credentials.
    - Scoring 0 would be no connectivity to NAU accounts
    - Scoring 5 on this would allow the users to log in once on any NAU platform (BBLearn, Louie, etc.) and access the app without the need to sign in again.
    - Connectivity is an all or nothing point evaluation.

### 3.2.2 Alternatives

- **NAU CAS:** School authentication system directly tied to student accounts, and uses the NAU login page to directly log in with school credentials.
- **Google:** Authentication using Gmail to access all Google based accounts. A variety of authentication options are available for different types of applications. [10]
- **Facebook:** Uses Facebook's authentication system to allow login with a Facebook account. Any email domain can be used provided there is an account created using it with Facebook. [11]

- **Apple:** Functionally the same as facebook, however provides authentication through Apple's servers instead. Any email domain is accepted as long as it is associated with an Apple account. [12]

### 3.2.3 Analysis

- **NAU CAS**
    - **Documentation:** No documentation is available to students.
        - **Score: 0 out of 5 points**
    - **Implementation:** No implementation options are available to students.
        - **Score: 0 out of 5 points**
    - **Connectivity:** Being directly connected to the school account, it would provide the most effective and direct way of authenticating users.
        - **Score: 5 out of 5 points**
- **Google**
    - **Documentation:** Google documentation provides an extremely clear means of how to connect to their API's, how to select the proper one to be applied, code and examples to implement into a web page, and many additional resources to technology they use. Of particular note, JSON Web Tokens (JWT) are explained in great detail as they are a primary part of authenticating a user.
        - **Score: 5 out of 5 points**
    - **Implementation:** Given the large amount of documentation that Google provides, ease of implementation will be increased. There are many additional tutorials and guides available online as well.
        - **Score: 5 out of 5 points**
    - **Connectivity:** All student and faculty NAU accounts are managed by Google and are considered as such for authentication purposes. While the inner workings of the CAS system are unknown, implementing Google's authentication allows an automated redirect to the NAU login page. This redirect allows students to access NAU related pages without having to log in again.
        - **Score: 5 out of 5 points**
- **Facebook**
    - **Documentation:** Facebook's documentation was excellent in discussing how to implement it and connect to their API. However, it was lacking in visible demo's, sample code, and additional resources. This made their API rather unclear for someone unfamiliar with their system directly, and has not done similar work before.
        - **Score: 3 out of 5 points**

  - **Implementation:** Lack of effective entry level developer documentation increases the amount of time it would take to implement due to additional research and understanding. Additionally, no provided sample code or token information creates a longer period between research and application.
    - **Score: 2 out of 5 points**
  - **Connectivity:** Facebook requires a Facebook account, and this can be associated with any email domain. All students do not necessarily have an account with Facebook, and this option would not always use NAU credentials. Seamless transitions between NAU related sites would also be unable to occur as the school does not use a Facebook based system.
    - **Score: 0 out of 5 points**
- **Apple**
  - **Documentation:** Apple's documentation is very detailed and provides significant information regarding connectivity and gives many options for sample code. It does not however provide additional resources outside of their own system and how to connect or what to send.
    - **Score: 4 out of 5 points**
  - **Implementation:** The documentation is extremely helpful and implementing this system would be straightforward. That said, the missing additional resources would require slightly more time to implement. Without researching other technologies, it would have been unclear how their token system works and additional research would have been necessary.
    - **Score: 4 out of 5 points**
  - **Connectivity:** Apple has the same issues as Facebook for connectivity. Any domain is possible, not everyone has an Apple account, and it would not allow a single login between this and the NAU systems.
    - **Score: 0 out of 5 points**

**Table 3.2.3 Authentication Analysis**

| | Documentation | Implementation | Connectivity | Avg. Score |
|---|---|---|---|---|
| NAU CAS | 0 | 0 | 5 | 1.7 |
| Google | 5 | 5 | 5 | 5 |
| Facebook | 3 | 2 | 0 | 1.7 |
| Apple | 4 | 4 | 0 | 2.7 |

### 3.2.4 Selected Approach

Given the results, it is clear that using Google's authentication system is the only option to make sure students are able to log in with their school credentials. Google provides the best documentation, samples, and resources to effectively implement a connection between our system and their servers. The authentication token received will also be able to be broken down for various permission checks, ensuring that the proper users are accessing what they should.

### 3.2.5 Feasibility

While this is not a familiar area to any of the group members, it is something practical and relevant that will be implemented efficiently. Applying requests through Javascript to interact with the Google API, interact with the received token for authenticity, and applying token based constraints to data access will be performed. The main objective is a secure way of managing student data so only the appropriate users will have access, and using Google's system will provide the proper means to achieve this.

## 3.3 Services

Our website application will have to successfully call and store data from a related database system, assure proper authentication, and present an intuitive user interface for both graduate students and faculty. This will require multiple coding languages that have dependencies on one another and must be adequately integrated or we risk build failure. For an application of this magnitude, it is commonplace in industry to utilize frameworks, such as Spring, that eliminates having to write boilerplate code and instead provides structure to your application, which then allows for more focus on business and design logic.

### 3.3.1 Desired Characteristics

The chosen framework must have the following characteristics:

- **Documentation:** Having adequate documentation is crucial to understanding a framework and using a framework, especially when it is open source and used by a community. Any package, plug-in, or project with the intention of being open source must have extensive documentation otherwise no developers will be able to fully utilize or understand it. Highly accessible documentation and a large community of active developers will prove to be a valuable resource during future implementation.

    - A score of 0 would indicate there is absolutely no documentation for the given framework and little community activity.
    - A score of 5 would indicate there is an extensive amount of documentation available to the developer and an active community updating the documentation.
- **Implementation:** The framework should intuitively integrate into our project and must coincide with the languages chosen for our front-end user interface and database system. A desirable framework would ultimately reduce the amount of boilerplate code needed to have a web application make requests to a database and populate a database. The time gained could then be used to focus on the user interface.
    - A score of 0 would indicate great difficulty implementing said framework within its respective language.
    - A score of 5 would indicate that the framework can easily be implemented within its respective language and many additional languages as well.
- **Functionality:** A framework's functionality relates to the number of available projects and dependencies that can be utilized to properly integrate all the pieces of a given project.
    - A score of 0 would indicate the framework has no supporting projects or dependencies that aid in the integration of the framework.
    - A score of 5 would indicate the framework has an abundance of innate projects and dependencies that can be used accordingly during integration.

### 3.3.2 Alternatives

Here I will illustrate three possible alternative frameworks that could be used in our website application.

- **Spring:** Spring Framework is a Java focused framework that “came to existence somewhere in 2003 at a time when Java Enterprise Edition was evolving fast”. [2] It quickly became favorable amongst developers due to its Dependency Injection (or Inversion of Control) feature in which “rather than the application taking control of the flow sequentially, it gives the control to an external controller who drives the flow.” [3] Since its inception, Spring has been well received by developers and as such there are many projects, documentation and resources readily available for developers to learn from. A great example of one of these projects is Spring Boot which “provides us with a set of highly opinionated but extensible template[s] for creating various projects based on Spring in almost no time.” [2]
- **Django:** Django is a Python focused framework that allows for fast development and launch of web applications. According to their website, “Django was invented

to meet fast-moving newsroom deadlines, while satisfying the tough requirements of experienced web developers.” [4] Django’s main focus is launching applications from concept to completion in a short amount of time to allow for more focus on design and business logic. As it stands, Django is ready to manage things like “user authentication and content administration, right out of the box.” [4]

- **Bootstrap:** Bootstrap is a framework designed for HTML, CSS, and JS with the intention of “make[ing] front-end web development faster and easier. It's made for folks of all skill levels, devices of all shapes, and projects of all sizes.” [5] If utilized, Bootstrap would allow a greater focus on the front end side of the project and is easily usable with any of the mentioned web programming languages. It also has exceptional documentation “for common HTML elements, dozens of custom HTML and CSS components, and awesome jQuery plugins.” [5]

### 3.3.3 Analysis

In this section the alternatives described in section 3.3.2 will be fleshed out in terms of the desired characteristics and graded on a scale of 1 - 5, where 5 is an exceptional display of said desired characteristic and 1 being an exceptionally low display.

- **Spring:**
    - **Documentation:** Spring is a highly venerated framework in the industry and has meticulous documentation. Spring’s website is a great starting point for documentation but because the framework is so prevalent, many third party guides and documentation exist, such as baeldung, which has proven to be a great resource in approaching Spring.
        - Spring’s documentation **scores a 5 out of 5** due to the sheer prevalence of updated documentation and guides available to the developer.
    - **Implementation:** Through keen use of Dependency Injection, the amount of documentation available, and the many Spring projects available to the developer, implementation of Spring should be quite configurable and manageable within the Java realm.
        - Spring’s implementation **scores a 4 out of 5** due to the nature of dependency injection; if your project has a large scope and requires many dependencies, it could be daunting having to implement each one and assuring that each dependency is accounted for.
    - **Functionality:** Thanks to the continually growing Spring community, there are many open source projects, built on top of the Spring framework, that can aid a developer and thus provides high functionality for many different purposes and languages. Spring Boot is a great example of a community

made project that “provides us with a set of highly opinionated but extensible templates for creating various projects based on Spring in almost no time.” [2]. Spring Framework can also facilitate access to MongoDB data, GemFire data, and JPA data with REST or MySQL which allows the developer great flexibility.

        - Spring’s functionally **scores a 5 out of 5** due to the extensive amount of projects and dependencies available to the developer.

- **Django:**
    - **Documentation:** Django has a vast amount of reliable documentation via the project website itself. Their website states plainly, “Django has a lot of documentation. A high-level overview of how it’s organized will help you know where to look for certain things.” [4] Outside of the project website, there are many third party documentation and guides available for developers to use.
        - Django’s documentation **scores a 5 out of 5** as its project website has a vast amount of documentation readily available from the start and many third party websites provide additional documentation and guides.
    - **Implementation:** In the case of our project at hand, “Django comes with an object-relational mapper in which you describe your database layout in Python code,” thus the service implementation would be require a degree of boilerplate Python code to be written by the user during development. This is because Django introduces the model layer, “an abstraction layer for structuring and manipulating the data of your web application.” [4]
        - Django’s implementation **scores a 4 out of 5** because the developer does not have access to any initializers or boot packages at startup, which means that a degree of boilerplate Python code must be implemented.
    - **Functionality:** Django has a Community page that claims they have been “building the Django Community for 16 years, 8 months.” [4] There is a fair amount of Django-based projects and packages that can be found straight from the project website. It is, however, a Python framework above all and is designed for the rapid development of a web application. Developers looking for more flexibility and package options may find it troublesome to sift through the vaguely named packages on the website.
        - Django’s functionality **scores a 3 out of 5** because there are a number of packages and frameworks available to developers but they are not easily accessible or readable.
- **Bootstrap:**

- **Documentation:** The Bootstrap website displayed a lengthy amount of documentation for each language implementable by Bootstrap. There are three usable versions of Bootstrap to choose from, the earliest version being a stable version and the newest version being dynamic. Bootstrap also has extensive tutorials on popular programming websites such as W3 Schools.
    - Bootstrap’s documentation **scores a 4 out of 5** because it is not nearly as verbose as the previous alternatives, however, because development is mostly carried out through HTML, CSS, and JS, the information being documented is quite trivial.
- **Implementation:** Bootstrap provides options for HTML, CSS, and allows the use of JavaScript through the use of plugins. This gives the developer great variety when it comes to creating a front end user interface.
    - Bootstrap’s implementation **scores a 4 out of 5** because HTML and CSS code are not very demanding.
- **Functionality:** Bootstrap may offer many options on the front end but the functionality in relation to the back end is negligible. At its core, Bootstrap is a front-end presentation framework and is meant to allow websites to adapt to any screen. It does not have any database management capabilities.
    - Bootstrap’s functionality **scores a 1 out of 5** because you can’t manage or populate database data through its framework.

**Table 3.3.3 Services Analysis**

|  | Documentation | Implementation | Functionality | Avg. Score |
|---|---|---|---|---|
| Spring | 5 | 4 | 5 | 4.7 |
| Django | 5 | 4 | 3 | 4.0 |
| Bootstrap | 4 | 4 | 1 | 3.0 |

### 3.3.4 Selected Approach

Table 3.3.3 indicates that Spring has the highest applicability and functionality for our project at hand. Spring’s documentation was extensive, approachable and clearly community driven, therefore it scored a perfect 5 out of 5. Spring’s implementation will place a large focus on dependency injection, which can facilitate a smooth integration if used correctly, therefore it scored a 4 out of 5. Spring’s functionality was most impressive due to the sheer quantity of initializers, projects, and packages available to the developer, therefore it scored a perfect 5 out of 5 here. We have elected to move forward with Spring Framework due to its vast community, extensive documentation, project repository, and its high functionality with Java and JavaScript, which we will be using on the front end.

### 3.3.5 Feasibility

Spring is still deeply rooted in Inversion of Control (Dependency Injection) so we must ensure that all the proper dependencies are accounted for. Spring Initialzr, Spring Boot, MySQL, and Gradle present us with quick and easy ways to begin demoing Spring web applications. From there we can properly scale our database to meet our needs and begin fleshing out the more important details regarding the user interface.

## 3.4 Front-End Web App

Web applications are common interfaces with databases. The front end application for this project is going to facilitate users accessing data from the database based on their login privileges.

### 3.4.1 Desired Characteristics

These are the characteristics that determine the approach for the front end’s programing language:

- **Ease of Implementation:** How simple it is to incorporate the language into the website, and how universal the language is to work with browsers.
    - A score of 0 would show it is either incompatible with most browsers or has a requirement preventing its use.
    - A score of 5 would indicate the language’s ability to be added to the website without requiring additional package installations.
- **Functionality:** Available libraries that can provide support for desired functionality.
    - A score of 0 would show it is limited to very specific functions.
    - A score of 5 would indicate the language's ability to implement any amount of functions necessary.
- **Documentation:** How well the language’s libraries are documented online.
    - A score of 0 would show the libraries are either not recorded or locked behind a large paywall.
    - A score of 5 would indicate the language’s various libraries are easily available for free for reference in a single space.

### 3.4.2 Alternatives

- **JavaScript:** The most common website language for more intricate web designs.
- **Python:** A popular scripting language with web coding support libraries.
- **HTML:** The base web programming language.

### 3.4.3 Analysis

- **JavaScript**
  - **Ease of Implementation:** JavaScript is used as a client-side programming language by 97.9% of all the websites. [6] To include JavaScript in a website it only requires an extra line of code in the html file.
    - **Score 5 out of 5**
  - **Functionality:** There are several database specific libraries available. Ie TaffyDB, ActiveRecord.js
    - **Score 4 out of 5**
  - **Documentation:** Due to its popularity there are many sources of documentation for the various libraries and functions of JavaScript from various websites.
    - **Score 4 out of 5**
- **Python**
  - **Ease of Implementation:** Requires third party frameworks to implement python into a web application.
    - **Score 4 out of 5**
  - **Functionality:** Python is a high functioning programming language. I am also more familiar with Python over the other two languages.
    - **Score 5 out of 5**
  - **Documentation:** Python documentation is spread out but readily available on the internet.
    - **Score 4 out of 5**
- **HTML**
  - **Ease of Implementation:** HTML is the base level language, and is essentially required for web applications.
    - **Score 5 out of 5**
  - **Functionality:** Was originally designed for formatting and displaying academic papers. Very little functionality beyond formatting text.
    - **Score 1 out of 5**
  - **Documentation:** Widely available, though spread out, online documentation is available.
    - **Score 4 out of 5**

**Table 3.4.3 Front End Analysis**

| | Implementation | Functionality | Documentation | Avg. Score |
|---|---|---|---|---|
| JavaScript | 5 | 4 | 4 | 4.3 |
| Python | 4 | 5 | 4 | 4.3 |
| HTML | 5 | 1 | 4 | 3.3 |

### 3.4.4 Selected Approach

Based on table 3.4.3, most of the front end implementation should be conducted with JavaScript or Python. Due to JavaScripts ease of implementation, JavaScript will likely be the primary language, with Python being used to fill in any functional gaps if necessary.

### 3.4.5 Feasibility

A user-friendly front-end application is vital to the sponsor's goals. A successful implementation will not only communicate with the database, but also be easily accessible and understandable for users with varying computer literacy.

# 4.0 Technology Integration

Tying all of the previous technologies together yields a fully functional product with proper communication between all parts. Communication between all parts requires an internet connection and will fail if communication between each step is not available.

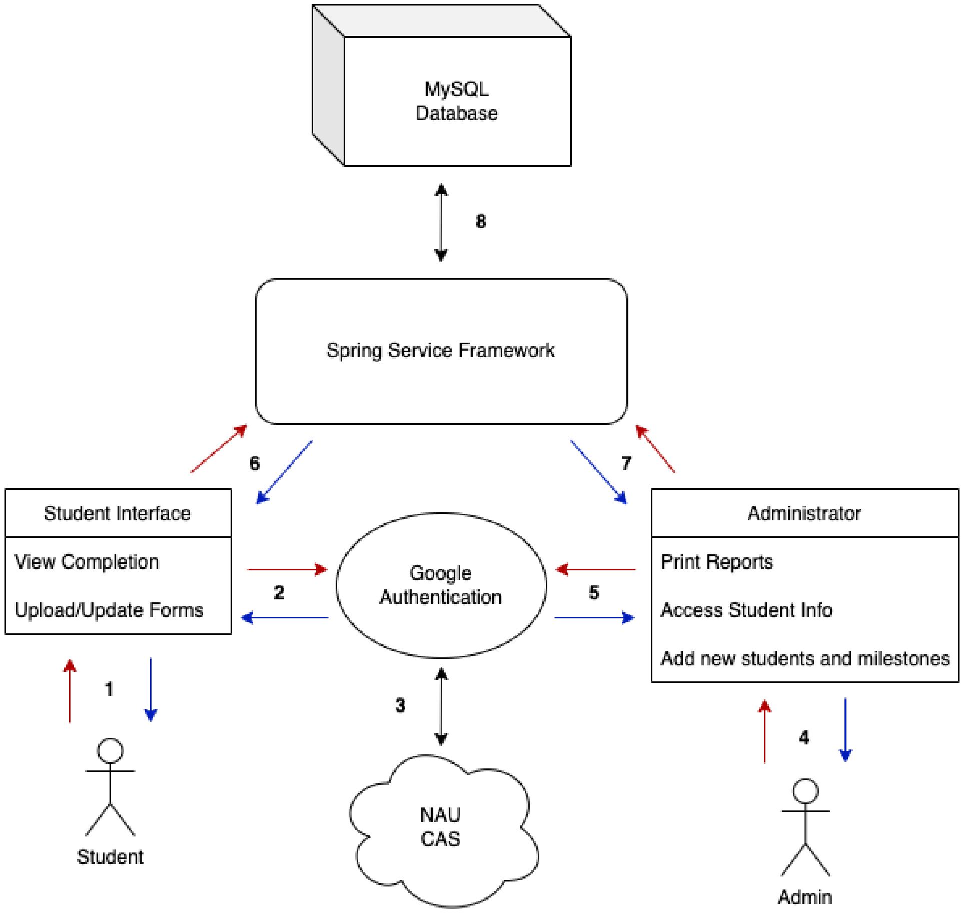

**Figure 4.0 Application Data Flow**

As shown in Figure 4.0, there are many points of communication throughout the program. Color coding of the arrows indicates direction of data, with red as being sent and blue received all relative to the user. Numbers have been provided to ease in explanation of where the data moves, and will be explained below. There are effectively two basic paths; the different paths, while very similar, provide varying access between students and administrators. Administrators have additional permissions allowing them to add or remove students and milestones, access reports and update local files if necessary.

Data flow for the product begins with the students and administrator opening the web app interface and selecting “Log In with Google” (displayed by **1** and **4** respectively).

This will redirect to a Google page to input the user's email (displayed by **2 for students** and **5 for administrators**). Both user types will be redirected to the NAU CAS login page to submit their normal school login info (displayed by **3**). From there, data will be returned to the Google authentication server, then back to the interface (students: **3** → **2**, administrators **3** → **2**). The token would be processed within the web app interface for authenticity and user information. Once processed, and the user makes a request of any kind, it is sent to the service framework (displayed by **6 for students** and **7 for admin**). The framework processes the request and retrieves data from the database (send request and received data displayed by **8**). This requested data is sent from the framework to the interface to be displayed (**6 for students** and **7 for admin**) and finally it is available and viewed by the student and admin (**1** and **4** respectively).

# 5.0 Conclusion

Learning from one’s mistakes is an invaluable part of the education process, however it must be facilitated through a feedback process. If said feedback process is denied, there is a clear barrier the student must pass through before they can reflect on their work. Team What’s Up Doc aims to reimplement the feedback process for graduate students enrolled in the NAU Curriculum and Instruction doctoral program through an accessible website application. The team started by addressing four key technological challenges we might encounter when creating our website application. The first challenge addressed which database management system better suits our project needs. The team chose to move forward with MySQL as our DBMS due to its ease of implementation, scalability, and popularity. The second challenge involves authentication and user privileges. The team chose to use Google authentication here which can easily be set up to work with students and administrators NAU emails. The third challenge entails implementing a service framework that can properly integrate all the front-end code with the corresponding back-end database. The team chose to use Spring Framework here due to the exceptional amount of documentation, projects, and the ease of implementation with a variety of languages. Lastly, the fourth challenge addressed how we would tackle our front-end web application. The team decided to move forward with JavaScript as our primary front-end language and Python for some of the functional gaps. Armed with our proposed solutions, Team What’s Up Doc will look to create a viable software solution for our clients. The next feasible task will be to begin developing a web application demo that properly showcases our chosen technologies working in unison. A working College of Education website application would introduce an efficient feedback mechanism for C&I graduate students and offload a tremendous amount of stress for our clients Gretchen and Michele.

**Table 5.0 Chosen Technologies**

| Technological Challenge | Chosen Solution |
|---|---|
| Database Management System | MySQL |
| Authentication and User Privileges | Google Email Sign In |
| Services | Spring Framework |
| Front End | Python, JavaScript |

# 6.0 References

[1] Hanson, Melanie. "College Enrollment Statistics [2022]: Total + by Demographic." *Education Data Initiative*. 22 Jan. 2022. <https://educationdata.org/college-enrollment-statistics#:~:text=Report%20Highlights.,students%20are%20in%20graduate%20programs.>.

[2] Chandrakant, Kumar. "Why Choose Spring as Your Java Framework?" *Baeldung*. 11 Mar. 2022. <https://www.baeldung.com/spring-why-to-choose>.

[3] Shankar, Ramya. "10 Best Java Frameworks to Use in 2022 [updated]." *Hackr.io*. <https://hackr.io/blog/java-frameworks>.

[4] "Django Overview." *Django*. <https://www.djangoproject.com/start/overview/>.

[5] Mark Otto, Jacob Thornton. *Bootstrap · the World's Most Popular Mobile-first and Responsive Front-end Framework.* <https://getbootstrap.com/docs/3.3/>.

[6] "Usage Statistics of JavaScript as Client-side Programming Language on Websites." *Usage Statistics of JavaScript as Client-side Programming Language on Websites*. W3Techs. Web. 11 Mar. 2022. <https://w3techs.com/technologies/details/cp-javascript/>.

[7] "MySQL Documentation." *MySQL*. Web. 8 Mar. 2022. <https://dev.mysql.com/doc/>.

[8] "Release Notes for Mongodb 5.0." *Release Notes for MongoDB 5.0 - MongoDB Manual*. <https://www.mongodb.com/docs/upcoming/release-notes/5.0/>.

[9] "PostgreSQL 14.2 Documentation." *PostgreSQL Documentation*. 10 Feb. 2022. Web. 8 Mar. 2022. <https://www.postgresql.org/docs/current/>.

[10] "Using Google ID Tokens to Authenticate Users | Cloud Endpoints with Openapi | Google Cloud." *Google*. Google. Web. 8 Mar. 2022. <https://cloud.google.com/endpoints/docs/openapi/authenticating-users-google-id>.

[11] "Facebook Login: Developer Documentation." *Documentation - Facebook for Developers*. Web. 8 Mar. 2022. <https://developers.facebook.com/docs/facebook-login/>.

[12] "About Sign In With Apple." *Apple Help Library*. Web. 8 Mar. 2022. <https://help.apple.com/developer-account/?lang=en#/devde676e696>.