NORTHERN ARIZONA UNIVERSITY

School of Informatics, Computing, and Cyber Systems

Requirements Specification

November 28, 2022

Team Teacher To-Do

Project Sponsor: Chris Aungst

Graduate Mentor: Italo Santos

Faculty Mentor: Michael Leverington

Version: 3.0.0

Team Members:

Sam Gerstner (Team Lead)

Alexander Frenette

Noah Nannen

Shlok Sheth

Accepted as baseline requirements for the project:

Client:	For the team:
Date:	Date:

Table of Contents

Introduction 3
Problem Statement 4
Solution Vision 5
Functional Requirements 7
Non-Functional Requirements 10
Environmental Requirements 12
Potential Risks 13
Project Plan 15
Conclusion 16

Introduction

Arizona and states across the nation are facing an unprecedented teacher shortage; Teachers are a critical part of the development of any society and states are taking swift action to address this shortage. To help fill vacant teaching positions in the state, the Arizona Department of Education is now allowing education students in their final semester to fill these vacant teaching positions in lieu of the traditional student teaching experience. This benefits the students of Arizona, as well as the teacher who gains valuable classroom experience even before earning their degree.

While this program has many benefits, it carries a heavy administrative burden for the NAU and College of Education officials who must track all of the requirements associated with the program. The College of Education is currently using a manual process in an Excel spreadsheet to track student requirements. This process is extremely cumbersome and consumes an estimated 200 man-hours each semester that could be used to provide other support services to students. Because this process is so time consuming, the College of Education is looking for a web application to help them track and maintain documentation related to student success in this program.

There are approximately 75 students each semester participating in this program, and each degree program has different requirements that must be met. In addition to the general requirements to become a teacher, there are subject certifications exams that teachers must pass to be able to teach a subject. Documentation of exam scores, fingerprinting requirements, and more are all needed for a student to participate in this program and these documents must be maintained by the College of Education in the event of a state audit.

College of Education staff are looking for a streamlined way to store and track documentation, as well as generate reports for individual or groups of students to see

what requirements have/have not been met. By freeing up the time that is currently used to maintain this documentation, COE staff can provide more support services to their students and reduce the chance of human error which can be costly during an audit.

Problem Statement

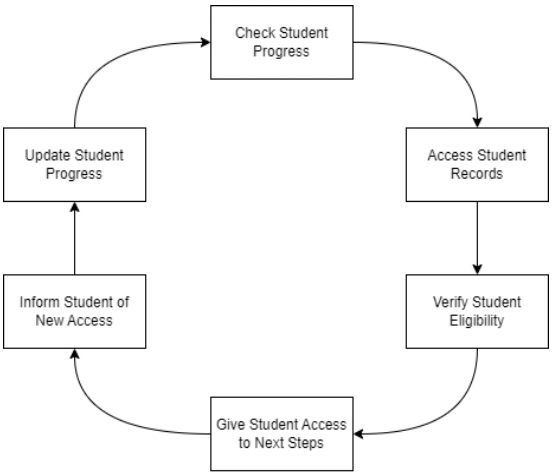


Figure 1.1

The main issue that the client has is that the current system process they are following is very outdated. As shown in Figure 1.1, the clients use a Microsoft Excel spreadsheet to monitor the students' progression towards their goals and throughout their courses. This current system cannot send or receive notifications, so it relies on exterior administrators to send email notifications to students and other administrators. This becomes a problem when a timely response is needed by the administrator, as they would be unable to access the data without requesting it from the selected source. It is also an issue with validation, ensuring the data is accurate if it is not coming from a firsthand source.

Another main issue with the current system is the nuanced versioning of the process. It is incredibly difficult to understand without the proper training, and not many people even have access to the system to be able to learn its capabilities. The process being run through an outdated system such as Microsoft Excel, especially when it is responsible for a portion of the graduates of the College of Education, is incredibly dangerous. It relies on a very precariously balanced system of equations and tracking to ensure that students have the qualifications to become student teachers, however any user can accidentally change or alter those equations and ruin the entire system. The client has expressed interest in a more definitive system that is not so easily altered.

The last issue of usability becomes an issue too, considering the client wants this program designed for faculty and students. Learning all the equations and cell requirements for the current document will undoubtedly be a time commitment to anyone who is looking to access the data for a certain student or update a student's status on their progress Likewise, If a student wants to be able to access where they are in the process of ensuring they are on track to becoming a student teacher, a Microsoft Excel document filled with equations and numbers will hardly be a clear and concise statement on their progress as opposed to a system created for this task specifically.

Solution Vision

We are planning to create a website using Spring Boot Java that is going to have a user-friendly interface. The website works as a to-do list that contains all the requirements to enroll in the Student Teacher Intern Certificate and helps the student to keep track of and follow through with the list of requirements. STIC requires a hefty list of requirements in order to make a student eligible to teach in a school, obviously, you don't let anyone with a degree teach students and become a teacher.

The Schema of the database is quite complex, and the table is 46 pages long, which shows the number of requirements a person must fulfill. Due to this level of complexity, we also want our website to handle subquery requests comfortably as there will be a lot of variables involved in resolving the data. Considering all the scenarios we plan to create a website that will calculate and plan for the student based on their goal and completed requirements.

The primary goal of the application is to track completed and missing requirements for students participating in the STIC program. By using a systematic approach to pull student data, COE staff can avoid hours of manual work to load student data into an excel spreadsheet and manually track requirements. Because the application keeps track of all requirements, COE staff can easily log in and generate reports to see which students have missing requirements, etc.

We are planning to keep the student information secure by introducing a hashing algorithm into the website backend, this will keep the student information secure by encrypting the FERPA related information. Currently the department is handling the condition by manually checking for the completed requirements for each student and laying out the plan. Whereas the website we are building is going to create that in a few seconds for the students on demand, with no room for human error.

The website consists of two types of users, one being a student who is our user and the other being an administrator who is going to manage the schema of the database if there are any changes in the structure in the future. Our website is going to give the benefit of doubt to the data stored in the database rather than checking its credibility. One of the stretch goals described by the client is that they want the user to get notified about the pending and upcoming tasks on their phone via SMS, this can be integrated with our system although it requires us to set up an SMS server. We look forward to working on this feature once the goals specified by the client are achieved.

As one of the risks listed in this document, we don't plan to provide any access to change or re-write the FileMaker Pro database. Even if the administrator does not have access to change or update the table, the only way to do that is by doing it manually in FileMaker Pro. Although this has created a dilemma. The problem with this type of system is that once the user finishes the next task, there is no input stream into the data that updates the change and creates the latest calculation. We plan to solve this by making the department manually update the table by verifying the information.

We don't plan on storing or caching any personal information in the website backend, it is going to use CAS to let the NAU student log in into the portal which will directly pass the information to the place holders and the website is going to use these placeholders to search through the database and encrypt the information by not storing but just displaying it directly to calculate the plan for individual student.

Functional Requirements

Currently our application is not expected to be used by members external from NAU. Our current users are staff from the NAU College of Education, and the students of their program. Because of this we can shift the burden of user authentication and account creation to the existing NAU identity accounts and their authentication system. A valid staff member, or student are therefore automatically valid members of our system. There will be no need for an additional account creation process on behalf of our users. There is interest in possibly providing access to our system to external parties. To satisfy this requirement we would need to implement a separate flow for secure account creation for these parties.

As we know, the major functionality of our app is to track eligibility of a student to participate in the STIC program, it might be of value to provide a general view for all individuals that will list all 22 programs, and their requirements. This is not to say that we will be tracking this requirement, but instead provide resources that do not require authentication to view.

Administrator User Requirements

Administrator/staff accounts do not have the same requirements as students. Administrators will use the application more as a monitoring tool, instead of actively maintaining the records. This means that administrators will need to be able to view the status of every student. We will first provide a general list of students that will act as quick summaries of each student as well as an individual view of a student. We intend for the general view to provide the ability to transition to the singular student view upon selection. This singular student view will then provide more detailed information.

These two styles of views of student data for administrators will be provided by the ability to filter. The first of which is to filter an individual student, by their NAU student ID, or their email, both of which are unique identifiers. Another filter parameter that has great value is completion status. To be able to filter out students that have or have not

completed their requirements allows for staff members to handle these students accordingly. An example being, a staff member may wish to filter students that have not met all requirements, and possibly follow up with the student to address any impediments. The filtering systems will also need to be flexible enough to provide the staff with the ability to filter students in ways that may not yet be considered.

Student User Requirements

As mentioned earlier, students will not have the requirement of account creation, as this is handled by external services through the NAU. They will however need to be able to sign up using these systems.

The main requirement for students is simply to track the status of their individual requirements as per the program selection in the STIC program. The determination of which program and their accompanying requirements should be independent of student interaction. The intent is for the system to use existing data to provide an automatic selection for the student.

Once a selection for their requirements has been made, there needs to be a logical suggestion for tasks to be completed. If there are tasks that are unable to be completed until a prerequisite has been met, these prerequisites must be made clear. The tasks themselves must also have persistence; once a task has been marked as completed, this state must persist.

Some tasks require the student to take external testing and report the scores. For all external testing, along with any other document associated with said tasks, we need to offer the ability to store said documentation and associate it with both student, and task. This ability to store documents using our application will act as a centralized repository for all necessary documentation. Student and administrator alike should also be able to retrieve said documents and store them on their local device. While not critical, it would be a nice to offer the ability to export all documents for a student profile as a single zip. Another non-critical requirement, but a nice feature, would be to allow users the ability to

view stored documentation within the web application itself without the need to download the file.

Within the STIC program there is student-teacher acquire multiple signatures from multiple individuals when signing an agreement for the student teacher intern. This document requires signatures from students, NAU staff, and members of the intended school. The process of acquiring so many signatures is a difficult and time consuming one. That is why a desired feature is for students to upload this document and send requests to the pertinent parties for them to review and sign electronically. The process of signing will be independent of accounts, thus allowing external parties to still participate. There are requirements for legally binding documents and their e-signature. This will be discussed later.

Communication with administrators and students also remains a critical task. While not critical, there is desire to offer communication functionality from within the app. There are several options to be considered. There is the possibility of offering messaging directly from within the app, another option is to involve email communication, and finally there is the option to involve text messaging. Currently we see no need for this functionality to be extended to all users, but instead should be between administrator and student.

Speaking of communicating, it would be a useful feature to offer a directory of administrator accounts to each student. This will enable students to resolve questions and issues quickly and seamlessly. One option is to simply provide a directory of all administrators and give users the ability to initiate communication with them directly. Another option would be to assign a point of contact for students. It may also be a combination of both. A student may have access to a particular “advisor”, but then be able to reach out to others if said advisor is unavailable.

General User Requirements

When using our application, it is necessary that the state of a user being logged in is maintained. We want the user to be able to access pages without the need to reauthenticate for every page.

In respect to authorization, due to the sensitive nature of student records, it is critical that authorization be strict. We do not want a user to have the ability to access the records of a fellow student. A student will only have access to their personal records. Due to the needs of an administrator, they will continue to have access to all student records and accounts.

Non-Functional Requirements

There are several general requirements for our system. They are high availability, redundancy of data, and fault tolerance. Our System will be using external services such as AWS to provide a great deal of our infrastructure. We intend on using these services to host a virtual server, a managed database, and our document store. Because of our reliance on these external services, we will assume their guarantee. As per AWS's SLA (Service Level Agreement) it is expected that EC2 server instances will be guaranteed uptime of 99.99%. This satisfies our need for high system availability.

RDS (Relational Database Service), Amazon's managed database services provide guarantees of 99.95% uptime. In addition, failover support in an option, enabling one system to go offline, and have its place taken by another. Both of these facts mean that our requirement for high system availability is being met. Another feature of their service is the backup and restore functionality in case of catastrophic scenario. This functionality helps guarantee data redundancy and safety.

For our blob/document storage we intend on using Amazon's S3 (Simple Storage Service). With this data we expect extremely high durability, and availability. This is critical as we expect data that is stored to never be lost, and always be accessible. As per

Amazon's SLA S3 provides "99.999999999% durability and 99.99% availability". This guarantee is exceptional and meets our application needs.

A general Requirement for any app as well as ours is the ability to scale. It is the case that this program continues to grow. Over the past few years, it has grown by roughly 12x. If this continues, we have the expectation to meet these needs. The issue of data storage is not a concern, as the services we are using will satisfy the needs for additional future space, as they are dynamic in nature. Our server and response infrastructure will also need to be able to scale. This will be possible through the use of a gateway/proxy which will allow for the distribution of requests amongst a collection of servers.

Focusing more on usability requirements, there should not be any need for a user, administrator or student, to need training while using our application. The experience should be intuitive and allow for independent exploration without the worry of making unwanted modifications. We can verify this through iterative deign involving feedback from expected administrators, as well as students.

In respect to responsiveness of our application we know that research shows that user engagement of web pages dramatically degrades after 4 seconds of the page loading. This will be our upper end for acceptable load times. Our system being dependent on FileMaker Pro, we will have to consider the time it takes for our response to be processed by this external service. Currently we have not yet interacted with the environment, but we believe a reasonable assumption for response time would be 150ms; that is time between sending initial request and receiving response. Our time constraint needs to be mindful of networking time, which we will assume a 150ms delay. That is time of initial request, there will be a 150ms delay to when the server will begin receiving the beginning of the request. The time delay will be applied to the response. In total that is 300ms. With these constraints, we have a total of 450ms that have been consumed, leaving our system just over 3.5 seconds to generate the page on our backend. We are able to track and ensure that this responsiveness requirement is being met by using services provided by Spring to log incoming requests, and their associated responses. By analyzing this log data, we are then able to make modifications if necessary.

Another requirement of our system is security of sensitive data. One of the most important pieces of data a user has is their password. We are able to offload this through the use of CAS of Google's OAuth services, which handle all of the user's sensitive account information including password.

A major requirement for our application is modularity and portability. As our client expressed, the initial goal is to develop a web page. As of now it is a requirement that our site works both on mobile web browsers, as well as desktop browsers. To maintain simplicity, we will also not support any version of Internet Explorer. We will however have the requirement that our site will work on 95% of web browsers. This data can be obtained using publicly available stats regarding browser prevalence, and the supported features of each browser (https://caniuse.com). In the future however it may be of interest to expand the scope of the project to include desktop of mobile apps. By developing a backend that is separate from the views we are able to then replace the view and treat it is an independent product. This will be accomplished by using a REST API which will provide all requesting services with the necessary views of the data, as well as a means by which to manipulate the state. As discussed earlier with horizontal scalability and how our web page generation could be scaled using a proxy, our API could be scaled by using an API Gateway, which allows for multiple instances to process incoming requests, and ac t as a single URI endpoint.

Environmental Requirements

Because the application is going to be accessing FERPA protected student data, we need to be extremely conscious of how we are accessing data and where that data is used within the application. There are no set guidelines for protecting FERPA data which makes determining exactly what access controls to put in place extremely difficult.

Our application also assumes that we know what user is currently logged in to the application so that we can accurately pull data. This requirement will be handled by

our authentication mechanism, which will provide us the UID or other unique identifier of the currently logged in user. For these purposes, how the authentication mechanism gets that identifier is not important, but rather that the identifier provided is guaranteed to be the logged in user.

The student data that will be accessed by the application is maintained by the College of Education in a program called FileMaker Pro. FileMaker Pro contains its own database which we will be using as the single source of student data for this application. FileMaker Pro is not an ideal database solution for these needs, but it is what is currently used by COE and therefore must be used by our application instead of an alternative data source.

Two Additional existing constraints pertain to user authentication. That is CAS (Central Authentication Service) and Google's OAuth 2. Our constraint is such that we must be able to verify that an individual is in fact a student, and the two systems currently in place to assist in doing so are these.

While we have the stretch goal of enabling electronic signatures, it is critical that we conform to all applicable laws and regulations. Currently the two major laws in place are the federal United States Electronic Signatures in Global and National Commerce (ESIGN) Act and the Uniform Electronic Transactions Act (UETA). To ensure the validity of all signatures it is critical that any solution we use is in compliance with the stands set forth by both acts.

Potential Risks

Our project risks mostly involve the use of outside systems to provide data. The main system we would be using is FileMaker Pro, which is the university's choice of data storage. However, that may not always be the case. If the university ever decides to change their main source of student data to a different database, the project will have to be radically altered to ensure it can handle the change. Our front end would mostly be ok,

but the backend systems would need to be able to adapt to any new storage system that may be implemented. The project would also need to be radically altered to be up to date with the newest schema, which would require modifications across the board with our backend database. While it is a risk, we want to ensure our project works well with FileMaker Pro and can securely transfer student data back and forth between our system.

That data then also becomes a risk. Accessing any student's private information requires security measures to be put into place to ensure that information is not available to any user to be able to see. We would need to consistently ensure that all student data is stored and encrypted so no other user or administrator would be able to access it all at once. This is quite a severe risk as well, making our project responsible for NAU student data and ensuring that data is as secure as possible.

A less severe risk involves our ability to learn new technologies in time. Our team is working to learn several new technologies over time. While we do believe that we are comprehending them, it is possible that we do not fully understand the solution we are aiming for in the time that we have available to us. As it would be detrimental to our project's overall usability and usefulness, it would simply limit us in our ability to be able to use our system as effectively as possible.

We also have the issue of our project being online. If we were to lose internet access, we would not be able to maintain the database. Many would consider that to be a large issue, but our project is reliant on the university internet, which if it does go out, will automatically disable the database and the login services, which are things we cannot control or fix. Therefore, they are of a lower severity than the other situations on this list.

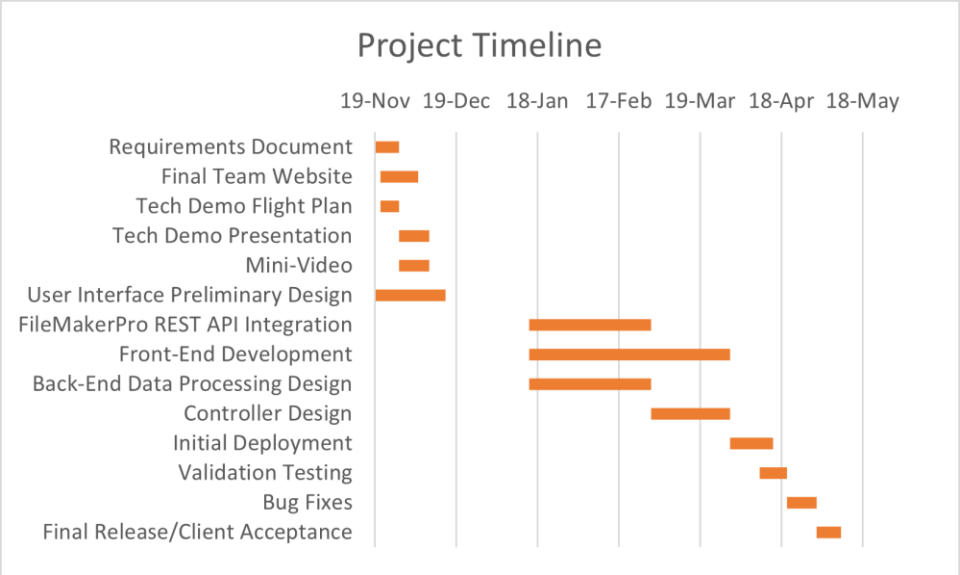
Currently, our largest risk to the project is the project itself. If our system is not fully accurate, we may be classifying students incorrectly and directly impacting their educational and possibly future careers. As a team, we will need to make sure that our system is fully accurate and able to correctly identify any errors that may occur, which can be mitigated with large amounts of project testing before rollout. We will also need to

have a person dedicated to just bug testing and fixing, to ensure that the project has no oversights or missed code within that could cause the system to have errors.

Risk	Description	Level of Severity (1 – 5)
Loss of Data	Losing FileMaker Pro and being unable to access student data.	3
Data Security	Being Unable to successfully protect and encrypt student data.	4
System Accuracy	Having the system fulfill requirements wrong, or not at all	5
New Technologies	Being unable to learn technologies needed to make the system more efficient	1
Loss of Internet	Not having access to the internet to be able to update the student data	2

Project Plan

Until this point, we have managed to solve the feasibility of the capstone project. In the security aspect of the website, we plan to implement SHA encryption for protecting the FERPA-related information of every student. We have achieved a decent level of security with this feature and have reached a point where the website user interface is friendly and in accordance with the NAU theme. From the Gantt Chart, we can see that we are on track to finish the capstone project by the end of April. Right now, we are on our way to working on the user interface and API side by side. We have divided our team in such a way that we could work on different parts of the website simultaneously. We plan on choosing the layout of the webpage with the consent of our client.



The Gantt chart here represents the different phases of development of the project, we also plan to work during the winter break to get a head start and so there is no break in the chart. Our main goal in this journey is to achieve the minimum required goals provided by the client. The deployment process also includes testing the database with the website and ensuring correct results. Once that is done, deployment will take place on the server.

Conclusion

We believe that as our project progresses, we will become closer to finding a solution to the teacher shortage in Arizona. It is imperative that the next generation of teachers are properly trained and better equipped to deal with the challenges of the next generation of students. We believe that our system of tracking software will not only assist the College of Education, and specifically our sponsor, in finding new teachers, but also allow them to be able to handle more important issues without a cumbersome system of management for these students. This document outlines the basics of what we want this project to be capable of, and how we plan to accomplish the goals to make sure it is the best version of itself. To create a software capable of organizing and planning the requirements of a student and be able to not only express to the student where they are in their academic plan, but where they are going and to do so in such a way that it can be

easily understood, but also express to an administrator of the College of Education where their students are in this process to be able to better help them see where aid is needed and to be able to free up their time elsewhere. With the help of this document, through analyzing risks, adopting requirements, adding functionality, and following our project plan, we are well on our way to accomplishing the goal of improving the College of Education and aiding students that may have a hard time along their academic journey.