

Requirements Specification

Version 2

18 November 2020

Team DigiFolio
Dr. Andy Wang
Fabio Santos

Burbank, Logan(lead)
Braudaway, Jackson
Chen, Kailin
Marschel, Parker
Yang, Zhenyu



Accepted as baseline requirements for the project:

Team Lead Signature: Logan Burbank

Date: 11/19/20

Client Signature: [Signature]

Date: 11/20/2020

Table of Contents

1. Introduction	2
2. Problem Statement	2
3. Solution Vision	4
4. Project Requirements	6
4.1 Functional Requirements	6
DR1: User System	7
DR2: Digital Portfolio	9
DR3: Goal Dashboard	9
DR4: Pulling from LinkedIn	11
DR5 (Stretch Goal): Job Posting Page	11
4.2 Nonfunctional Requirements	12
5. Potential Risks	14
6. Project Plan	16
7. Conclusion	18

1. Introduction

Having industry/internship experience prior to graduating can be a determining factor in whether a graduate lands a job or not. Unfortunately, this leaves those who do not have this prior experience still searching for a job after they have graduated or even forcing graduates to look into other fields not related to their degree so they can find work. This is not only just a problem for students attending Northern Arizona University; it is also an underlying problem for millions of other graduates looking for work.

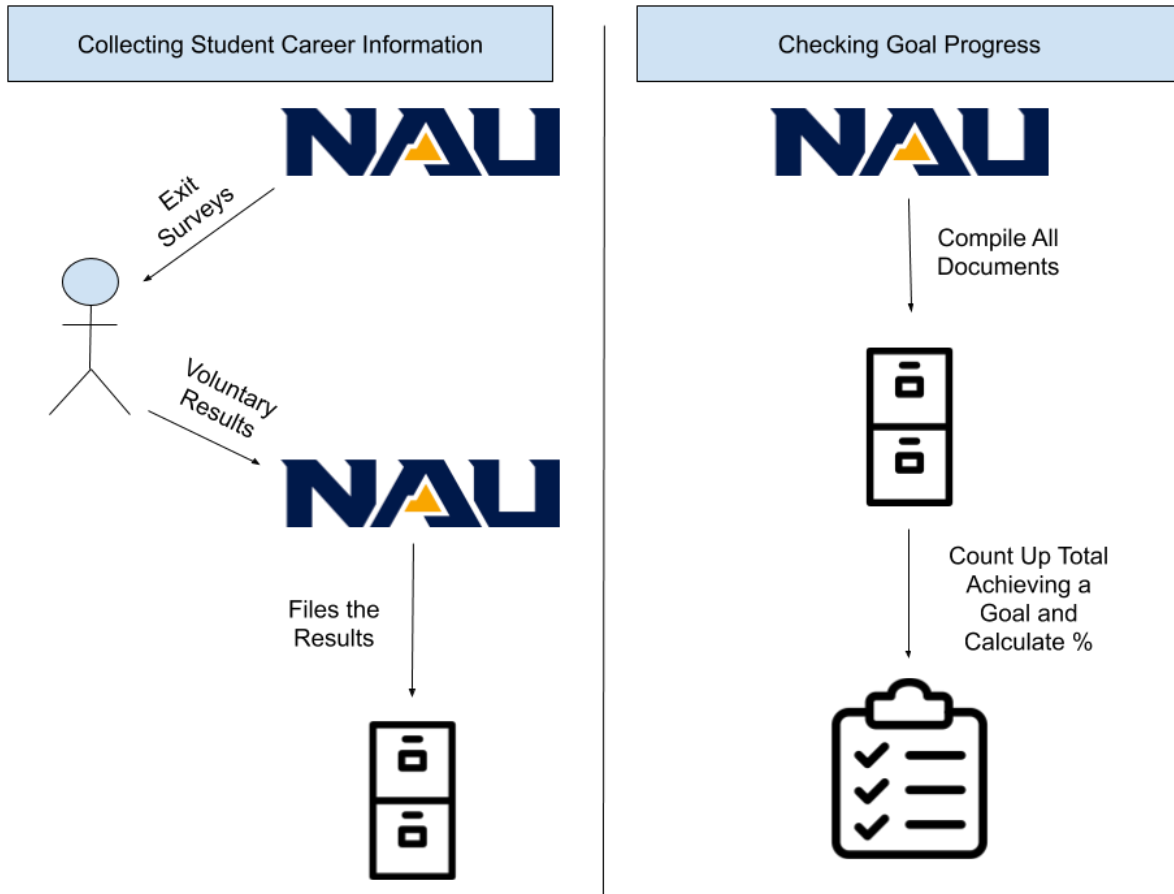
Currently, Northern Arizona University tracks student success through exit surveys before the student graduates. While this is somewhat effective in getting feedback from those about to graduate, this information is obtained much too late to actually help graduating students. Even with programs in place such as job fairs, Handshake, etc. they are all optional for the student to participate in. Our solution aims to change this and actually promote student success by tracking a student's progression at the beginning of their college career rather than at the end through the use of a web application that will display their career milestones and progress highlights. With this, faculty and students will be able to see how the student is doing at any point in their college career and be able to assist them in reaching their set goals.

Our sponsor for this project is Dr. Andy Wang, professor and dean of the College of Engineering, Informatics, and Applied Sciences here at Northern Arizona University. With Dr. Wang being the dean of the college, he looks over thousands of students. Understanding how well the college and its students are performing is not easy to do with only exit surveys. Our product will be able to paint an accurate representation of this so that Dr. Wang can work to help students achieve goals and be more successful by the time graduation comes. On the other hand, students will be able to use this tool to track their progress towards their career goals.

2. Problem Statement

As stated earlier, the student career success is currently measured through exit surveys around the time of graduation. Near the time of graduation the College of Engineering, Informatics, and Applied Sciences sends these exit surveys to those set to graduate. The survey is either filled out by the student or ignored (another problem with this method),

and that information is collected by faculty and stored. This method does work well for two reasons. First, it is not accurate nor complete, as students may skip the survey or students are still looking for a job at the time of survey. Second, it is not the most efficient way to track student career success because it is a manual process and cannot scale up easily. This problem is outlined in the left workflow diagram below.



The overall problems our project aims to address are the time at which success is being measured, the way success is being measured, and how that measurement is being calculated to aid students in pursuit of their goals. Currently, the university uses exit surveys near the time of graduation to give the university an idea of where a student stands only at the time of graduation. Looking closer at the overall problems with the process presents why they have become problematic:

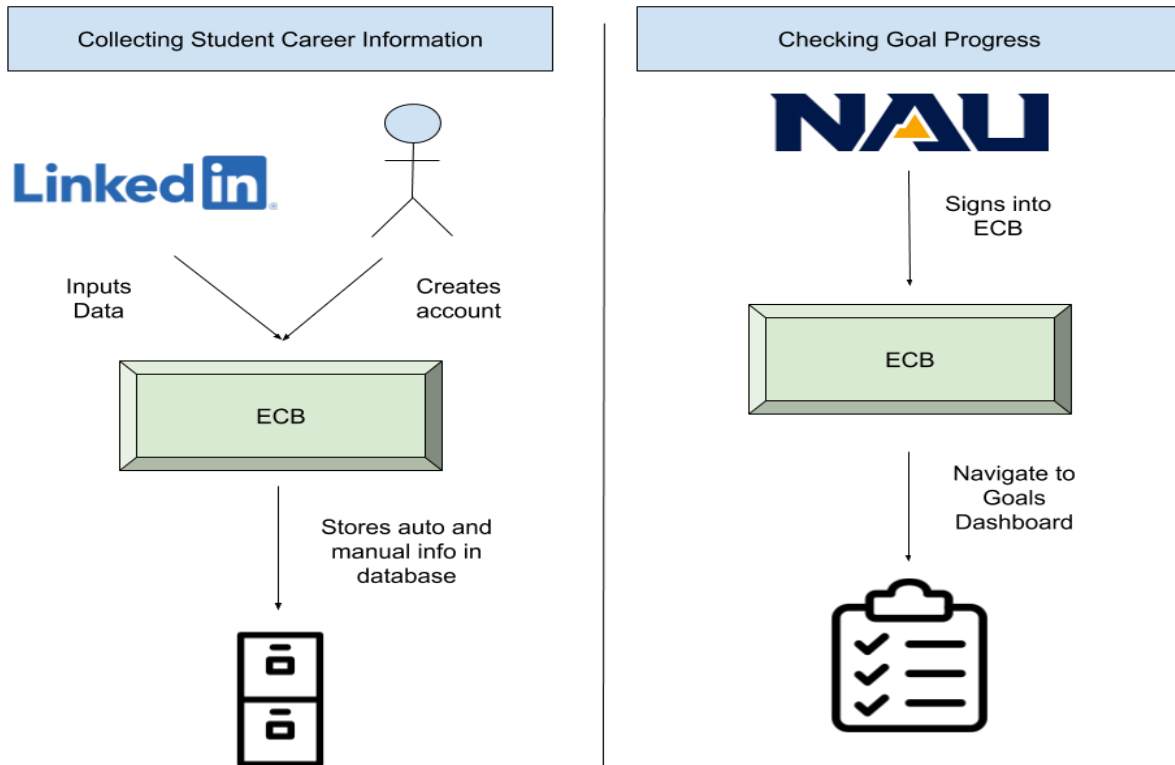
- Measuring student success with exit surveys
 - Optional to the student
 - Does not paint an actual representation
- Timing of measurement

- Does not measure success for previous years, nor a short period of time, e.g., six months, after graduation, which is a standard job placement period.
- Too late to help the student
- Calculation of measurement
 - Takes manual gathering of documents and manual counting
 - Simple, repetitive processes

After analyzing the current approach and process it is apparent that it is rather useless to both the student and the school and could be made considerably more efficient. Once again the essential problems occurring are the time at which student career success is being measured, the way student career success is being measured, and how that measurement is being calculated to display to both students and faculty. These big picture dilemmas run in parallel with the inefficiency observed, and our solution will solve them.

3. Solution Vision

The solution that we plan on implementing, currently referred to as the Engineering Career Builder (ECB), should cure the ailments of the current system. Our plan is to implement a web application that creates a digital portfolio for a student at the beginning of their college career, which has the ability to gather the student's experiential learning data and read data from the student's LinkedIn accounts and transform/add that information to his/her portfolio. It will display students' progress in their career path including their job offers, internships, externships, co-op, research experience, accomplished professional goals, etc. This will resolve the timing of when students are being recorded, be able to provide much more information about a student and how they are progressing in their programs, and will be able to use that information to positively service the student and program they are in. This can be seen in the following workflow that we envision using our solution.



Diving deeper there are specific features that we plan on including to allow this web application to do this including:

- Adopting an API to gather information from LinkedIn
- Easy Access
- Calculating student totals based on a given goal
- Displaying different charts of student progress based on program and department

Our system will be using data collected from LinkedIn profiles and from student input as they update their profile. With that information it will generate a portfolio that can be viewed by the school, alumni, and the student as a means to measure progression and assist in the acquisition of many things including but not limited to internships, jobs, references, goals, etc. As previously stated, the main computation that will be occurring is the collection of data from LinkedIn accounts, student input, and the transformation of that data into the students' portfolio on our web application. The initial setup of this system will be more overhead than the current system in place, but in the end there will be less work for faculty as they will not have to send out exit surveys, collect, and store them manually. The only overhead our system provides is to the student, as they will need to update their profile occasionally to make full use of the application.

We settled on our current solution because for one it will fix all the complications of the current system but also because it will encompass more features to model the students'

success accurately. This solution will further provide the application with all the necessary elements such as an API, etc. to make it possible to do everything we require it to. Once this system is functional it has the potential and should be implemented at other universities, perhaps even high schools, to really motivate and drive student success at an early stage in whatever setting and career they are in.

4. Project Requirements

In this section, we will detail every requirement we have acquired in order to fulfill our solution vision. We have gone about acquiring these requirements through multiple meetings with our mentor, along with internal meetings to discuss what would still be necessary and what still needed to be broken up. All of these meetings, as well as our own individual brainstorming, has led to the domain level requirements we need in order to create the solution.

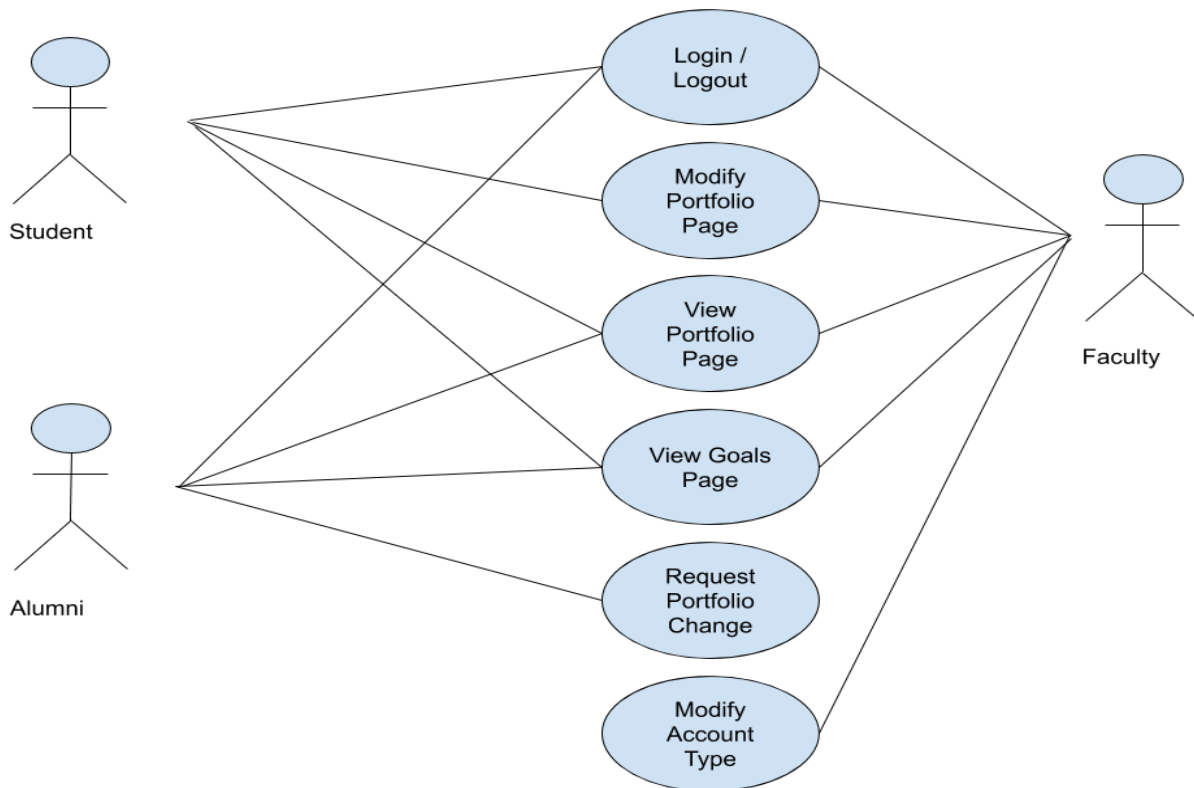
These domain level requirements we have envisioned are as follows:

1. The solution must provide a user access system with role-based permissions.
2. The solution must provide a digital portfolio for every student account.
3. The solution must provide a dashboard that displays progress toward specific goals, for individual students as well as for department and college.
4. The solution must provide information pulled from LinkedIn.
5. (Stretch) The solution must provide a job posting page.

Achieving these four domain level requirements (and possibly one stretch) will ensure all necessary functionalities are included in the solution system. In order to help achieve these domain level goals, we will break each one up into its basic functions in this next section.

4.1 Functional Requirements

In order to plan the implementation of the previously mentioned domain level requirements, we have broken them down to multiple levels of functional requirements in this section. Some of the basic user functions are outlined in the following diagram, where the basic user-facing functions are seen.



In the use case diagram here, it is clear to see that each user type does not have too much complexity in their uses. The three primary actors also have similar functions between each other. There are only two functions unique to one actor, and this is because faculty will have modification, and alumni will have request capabilities. In terms of functionality, the faculty will have the most freedom, students will have the second most, and alumni will have the least.

Domain Requirement 1: User System

Our first domain level requirement involves implementing a user system that will allow for user accounts with role-based permissions. There will be three account types, with these types being Student, Faculty, and Alumni. Each of these types will have specific basic functional requirements.

FR1.1: A student can create a student account with student permissions.

1.1.1: A student can log in and log out.

1.1.2: A student can view and modify their own portfolio page.

1.1.3: A student can decide what is visible on their page.

- 1.1.4:** A student can view public information on other student pages.
- 1.1.5:** A student can view the overall progress dashboard.

This first functional requirement is broken down into five smaller functionalities. It is a fairly simple requirement, mainly just requiring permission to be made for a student account. A student's use in this application will involve logging in and out, viewing portfolio pages, modifying their own page contents and visibility, and seeing the overall progress. This will be implemented through a header bar of some design.

FR1.2: A faculty member can create a faculty account with faculty permissions.

- 1.2.1:** A faculty member can log in and log out.
- 1.2.2:** A faculty member can view and modify any student portfolio page.
- 1.2.3:** A faculty member can view the overall progress dashboard for individual students, as well as a summative progress dashboard for a program, department, and college.
- 1.2.4:** A faculty member can change a student account into an alumni account.

The second functional requirement is broken down into four smaller functionalities, and it is also a simple requirement. This is just like the first one, with the exception that it is for the permissions of a faculty member. The biggest difference here is that the faculty member is able to modify any student page, but is not able to select what pieces are public. Also, the dashboard will show faculty individual student progress toward her/his goals, as well as a summative progress for a program, a department, and a college. Finally, the faculty will also be able to modify a student account's status, allowing student accounts to become alumni accounts through a faculty member.

FR1.3: A student account will become an alumni account when the student has graduated.

- 1.3.1:** An alumni can log in and log out.
- 1.3.2:** An alumni can view their own portfolio page.
- 1.3.3:** An alumni can decide what is visible (public) on their page.
- 1.3.4:** An alumni can request to have their portfolio page updated.

The third functional requirement is broken down into four smaller functionalities, with this user type having the least functionality. Ultimately, this account is similar to a student account, with the ability to request page updates rather than being able to directly modify the page. This is so the information being added can be verified by an administrator.

Domain Requirement 2: Digital Portfolio

Our second domain level requirement involves implementing a digital portfolio page that will be built for each student account created. This page will be linked to a student account, and it will include information about their name, contact information, and career milestones.

FR2.1: The portfolio can be modified by the user.

2.1.1: A user can add a career milestone to the page.

2.1.2: A user can remove a career milestone from the page.

2.1.3: A user can report a data problem to a system administrator.

The first functional requirement is broken down into three smaller functionalities. These are mainly the different ways a user is able to modify a portfolio page. These functions are primarily focused around the milestones on the portfolio, as these pieces will be the part of the profile that are constantly being updated, especially with them being the primary piece of information being displayed. The user will be able to add and delete milestones, and an extra button will allow the user to report a message to an administrator if there is some piece of information that needs to be fixed outside of the milestone updates.

FR2.2: The portfolio will display student career information.

2.2.1: The student name will be displayed.

2.2.2: The student career milestones will be displayed.

The second functional requirement is broken down into two smaller functionalities. These both involve information being displayed on the page, and no user input outside of the request for the page initially. Still, the page will need to be able to build using the user information when they choose to request their page. This will involve displaying the student name, as well as the milestones connected to the student. These are the two types of information being displayed.

Domain Requirement 3: Goal Dashboard

Our third domain level requirement involves implementing a digital dashboard capable of displaying the current progress of the student population in achieving university-specified goals. For now, there are two different goals this dashboard will be capable of tracking: the total percentage of students with an internship (including externship, Co-op, research project with experiential learning outcomes), and the total percentage of students with a job offer.

FR3.1: The dashboard must calculate the percentage of students with an internship (including externship, Co-op, research project with experiential learning outcomes) for a unit (program, department, or college).

The first functional requirement focused on the dashboard is simple enough to not need any further breaking down. Ultimately, this will be a function that is done semi-automatically, with the dashboard module needing to be able to calculate how many students have earned an internship, externship, co-op, or research/experiential learning opportunities.. As this will be displayed in multiple possible ways, the dashboard will achieve this by keeping a total count of students and consistently updating the number who have had this internship. This way, the two values it needs are always available to display the results as a fraction or as a percentage. Also, the internship will be checked using a flag attached to the student, to allow for quicker calculation when running through all of the students.

FR3.2: The dashboard must calculate the percentage of students with a job offer for a unit (program, department, or college)

The second functional requirement focused on the dashboard is also not needing to be broken down further. This will be extremely similar to the previous challenge, with the only difference here involving what goal is being checked. In this case, the module will already be storing the total number of student accounts, and it will now store the number of students who have received a job offer. Then, it works like the previous requirement in how it will check and calculate the results.

FR3.3: The dashboard must display each percentage in multiple chart forms

3.3.1: The percentage can be displayed as a fraction.

3.3.2: The percentage can be displayed as a pie chart.

3.3.3: The percentage can be displayed as a progress bar.

The third functional requirement focused on the dashboard is the only one needing to be broken down further, mainly to lay out the different formats that will be possible on the dashboard. As it stands, we have decided the dashboard will be capable of displaying the goal results in three different ways, with it being displayed as a text fraction, a pie chart, or a progress bar. The user will have the option to swap between these three different charts with a simple menu of choices. We also expect that both challenges will be displayed by the dashboard in a side by side type of way.

Domain Requirement 4: Pulling from LinkedIn

Our fourth and last domain level requirement involves implementing a module capable of reading data from LinkedIn and analyzing what data fits into which category. This data will then be used to populate the student portfolio pages. This module must be able to pull information from LinkedIn, analyze and classify the data, and store the data in its correct place.

FR4.1: The LinkedIn module can pull data from LinkedIn servers

4.1.1: A request can be sent to the LinkedIn servers using its API

4.1.2: The resulting package is checked to verify it is what was requested.

4.1.3: The package can be unpacked into temporary memory.

The first functional requirement focused on the LinkedIn module is broken up into three separate functionalities, with each being necessary to achieve the greater requirement of pulling data from LinkedIn. In order to do so, the module must be able to send a request to the LinkedIn server, verify the result that is sent, and unpack the result into temporary memory. These stored results will then be used for the other functional requirement, and this step will typically be done automatically, with the choice to scan for new data manually.

FR4.2: The LinkedIn module can classify the data into categories.

4.2.1: Data can be classified as an identifying piece (ie. email)

4.2.2: Data can be classified as a career milestone

The second functional requirement focused on the LinkedIn module is broken up into just two smaller functionalities, with the main purpose being to classify the information pulled from LinkedIn into categories. These categories as we see it will include two important ones to take note of, with the first being any identifying information (LinkedIn API uses email), and the second being a career milestone. LinkedIn will likely not make it obvious what each career milestone we are pulling entails, so this module will check and place the data in the right spot once it is pulled.

Domain Requirement 5 (Stretch Goal): Job Posting Page

One domain level requirement not mentioned above is the Job Posting Page. This is because it is our main stretch goal, and we plan to work on implementation if we are able to implement the rest of the domain requirements with time to spare. Through this, the job posting page is not urgent to the functionality of the rest.

This page will be made to display job postings that are on Handshake, and this will be done using a tool created for the exact purpose. This mainly leads to only one functional requirement, and that is displaying data.

FR5.1: The Job posting page can automatically display a list of job postings from Handshake, allowing links to quick access to the Handshake posting from their site.

This single functional requirement is all that will be required of the job posting page. It will be made as a gateway of sorts to specific pages on the Handshake servers. This will allow students to have a viewable list of job postings that they can easily access and apply for through the given route on Handshake.

Overall, these are all of the functional requirements we have acquired in order to ensure our solution. We do expect to handle each piece as a separate module, and bring them together to form the solution as a whole in one web application.

4.2 Nonfunctional Requirements

In this section following up the functional requirements, we will now lay out the nonfunctional requirements relating to the functional ones laid out above. These are related to the performance of the functionalities, rather than implementing the specific function. Our first goal is to implement our functional requirements, and we will follow it up with performance improvements. If a nonfunctional requirement is crucial for avoiding some malfunction risk, like data accuracy, it will be explicitly stated as such.

NR1 (Usability): A user can create an account within 4 minutes of first arriving at the homepage.

The first nonfunctional requirement is related to the first domain level functional requirement as a whole. Each smaller piece of FR1 involves creating a type of user account, and the goal here is to have a user who is new to the application have an account created within 4 minutes of first coming to the homepage. This is not urgent, and can be achieved through tweaking the user interface and login process once it is functional and working.

NR2 (Consistency): User portfolio information and settings are backed up twice a day.

The second non-functional requirement will be focused on the student portfolio pages. These pages will constantly have information being updated, meaning the data is constantly being used and modified. This requirement is to ensure that the data being written is backed up using our server implementation, ensuring data loss is never substantial in the case of a server failure. Twice a day also seems sufficient enough, as the pieces of information on the portfolio pages are not going to be needed in any critical capacity, and it helps lighten the load on the backend.

NR3 (Performance): A portfolio can hold up to 12 career milestones.

The third nonfunctional requirement is related to the portfolio pages as well, with this being related to how a user modifies their page. For functional purposes, our FR2 is based on being able to add and remove milestones. However, in order to save on space and have an idea of how much storage is needed, there must be a limit on the number of milestones. This limit can later be increased as the storage space is increased, but for now, 12 milestones seem to be a fair amount for students who have not been working in the field for too long.

NR4 (Security): The user is asked for permission to scan their LinkedIn information on account creation.

The fourth nonfunctional requirement is related to the privacy of the user, and is somewhat brought on due to environmental effects. This will mainly be related to FR4, as our LinkedIn module will be taking in information about users from an external source. With this being the case, we would like to give the user a heads up in the form of a checkbox, showing that they accept the data scanning. While this may not be necessarily required, as the LinkedIn data is publicly accessible, it is nice to give the user notice of what is being stored related to them. This is a critical non-functional requirement, as privacy-related problems can lose the userbase's trust.

NR5 (Performance): The portfolio and dashboard pages will load up within 2 seconds of requesting the page.

The fifth and final nonfunctional requirement being outlined involves the load times of the application. When a user is navigating the application, they will have mainly two different pages that they can view: the goals dashboard(FR3) and their portfolio page (FR2). While our primary goal is to get these pages functional, we do want to make sure the load times of the pages are responsive. Waiting for constant loading can lead to a horrible user experience, and with these pages being the central viewing materials of the site, improving these page load times through code optimization and reworking could help improve the overall experience the user has.

These five nonfunctional requirements are the primary performance-related goals we plan on implementing when the functional requirements are implemented. Some of these will be easier to implement than others, and some are more important than others. Most of them have to do with improving the user experience in some way, whether it be through speed, privacy, or an easy-to-use user interface. With these requirements laid out, we will now go on to discuss the potential risks we see arising when the solution is implemented.

5. Potential Risks

In looking ahead at our solution vision, we also have to envision the possible risks we will face. These risks are mainly focused on user-related dangers, as they are the primary ones who will be affected by any kind of risk to come up. We have laid out a table of the main risks we envision and go into detail on each one and how we plan to mitigate or avoid them if possible.

Risk Overview Table

	Likelihood	Severity
User Information Leakage	Low	High
Database Accuracy Error	High	Low
Frontend Application to Backend Database Connection Failure	Low	High
Network Security / Availability	High	Low

5.1 User Information Leakage

Since our project needs to store user data, it is our ethical responsibility to protect the information our users give us. This data contains not only what a student experiences during their college education, but also emails, passwords, and other account information. The information of the students’ experience is one of the key pieces of data in our project. Emails and passwords are much more important as they give some form of access to another person’s data, especially since passwords are commonly reused. We have decided to use the LinkedIn email as a way of linking accounts to ensure the correct information is being linked to the correct account. Moreover, the module will ask the user for permission to use the data when the account is created. If the users’

information is already leaked, we will send the email to inform the users to change their password as soon as possible in order to prevent further disclosure of information. Sending the email as a reminder to users in a timely manner is also a preventive measure.

5.2 Database Accuracy Error

Since the front-end application will constantly be accessing and using the database, we need to make sure it remains up and running as much as possible. The inability to use the database will result in users not able to retain progress made in modules, as well as reading data that may be false. This is a minor inconvenience, however, as the data is not critical. Also, we will work on implementing backups through our database system to help alleviate this. The biggest issue with the backups is ensuring the data remains correct and accurate, and having a backup will ensure any data corruption can be reverted.

5.3 Frontend Application to Backend Database Connection Failure

Since the dashboard will generate charts to show how well the students' goals are being achieved, we need to ensure that the dashboard has a firm connection with our database. Moreover, some data like the percentage of students who have had a job offer and the students who achieve the goals will consistently be updated rather than calculated at the time of the request. The risk is much more critical than the others, as the dashboard is one of the central features of the program. If the dashboard goes down, then most of the program itself loses its purpose as the dashboard displays the goals to reach.

5.4 Network Security / Availability

As with any website hosted on the world wide web, there is a risk of hacking attempts being made on it. A hacking attempt on our website can result in frustrated users, data loss, or worse. One type of attack could crash the application, preventing users from accessing the different pages. Other types of attacks may arise in the future as well. Luckily, the information encompassed in our software is not personal or sensitive outside of passwords to our system and basic contact information. We will help to alleviate this risk by using a simple encryption system when storing any private account information. We surely do not plan on storing passwords as plaintext, as anyone who gains access to the database, even an administrator, could use that information for their own gain.

PROJECT PLAN

Team DigiFolio
Logan Burbank

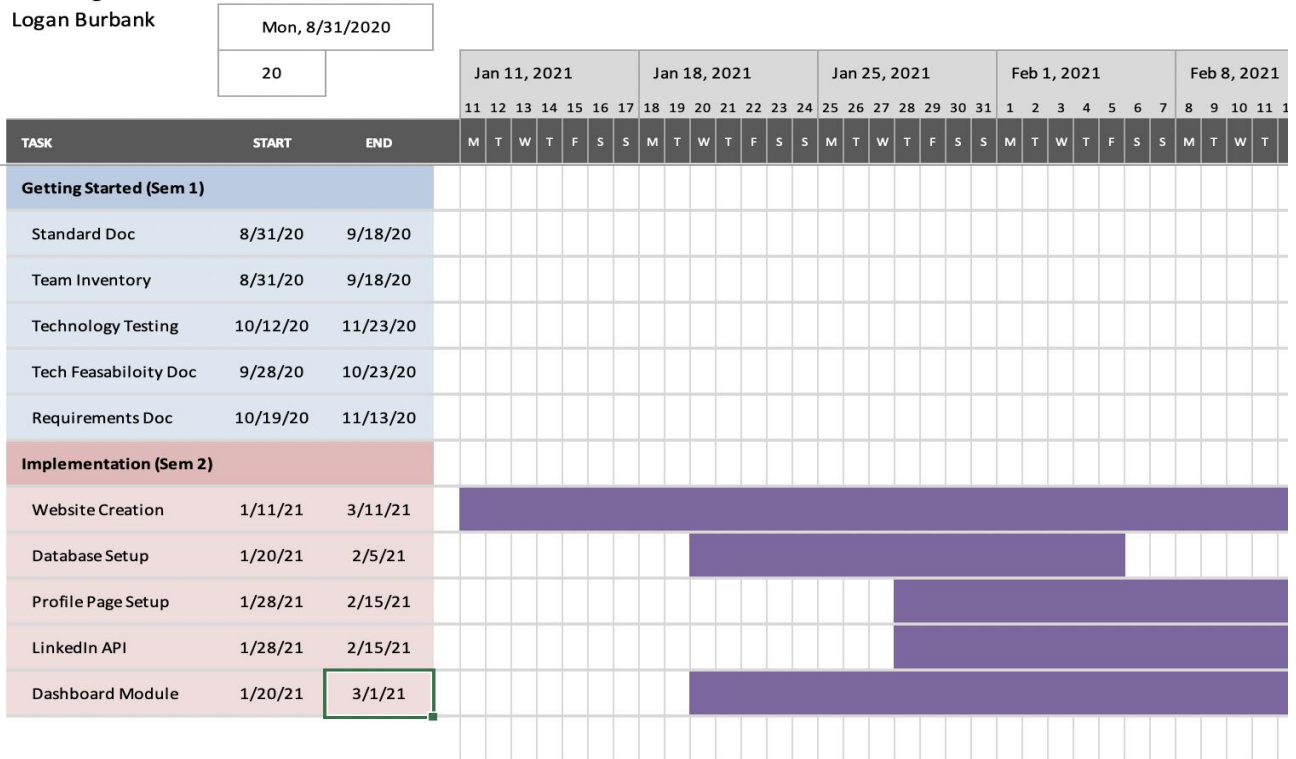


Figure 6.2

With these 5 milestones, there are sure to be smaller milestones around each of these overall milestones. Our team is currently still testing our technologies to ensure that we have the best technologies that are well equipped for the website we intend to create for NAU, as well as creating the proper documentation for both our client and ourselves to ensure the understanding of what our team is creating. As a group, we plan to tackle each of these technologies separately so we can ensure we are making the best decisions with our limited time. When we finally come together to implement each of these technologies, as a group, we will then integrate them into the final website.

These tasks will be completed in our next semester to come starting on January 11th, 2021. As we go into the next semester, our goals are to get our website designed with enough time to be able to edit the website to our client's liking. Starting with the basic design of the website, we then want to incorporate our database so that we can begin to store data. While the database is being created and set up, we also want to set up our dashboard where we will be able to calculate students' success and display it in a

graphical way where faculty can see the progress of our client's goal of 100% of students having an internship by 2025. After we are able to store data we want to create our user pages, so the data can be seen on each individual students' page. While we are designing and creating the user pages, we hope to be able to mine data from LinkedIn so that our data becomes full and accurate based on each students' milestones and achievements.

Through each of these milestones, we have planned them out to be done in line with each other so that if one task needs to be pushed back, then all the other tasks won't fall behind in their own schedule. As we finish this first semester, we have created a design review video which talks about what the problem and solution of the project are as well as functional and non-functional requirements of how the project will run. In addition to our design review video we have also created a tech demo which will show proof of concept for what we intend to create, which will give both us, as a team, and our client a better understanding of where we are headed in this project. With this schedule, we hope to stay on task and have a fully functioning, working product by April of 2021.

7. Conclusion

The main problem that we face is being able to account for the highlights of students' careers. When a student gets an internship or a job, NAU wants to highlight that fact and encourage every student to pursue anything that will help them get a job in the future. Our client, Dr. Wang, wants to set a goal for NAU Engineering students that one hundred percent of students will have had at least one internship or experience in the degree they're pursuing by 2025.

Currently, NAU has no way of tracking students' progress, we don't know who has had internships, or if any student is even benefitting from their learning. Our team aims to give students a place where they can store the milestones of their time here at NAU. NAU currently has exit surveys in place that ask students if they have gotten a job offer since graduation. There are a couple of problems with this, one being is that students might not have had a job offer one to two months after graduating. Another problem with this is that students might not always fill out the survey.

With our website, we hope to create a place where students can constantly be putting in milestones onto their profile page where NAU can see and track each student's progress and even help them get job offers. Our team hopes to create a digital portfolio where students can set goals for themselves, and display milestones of their achievements. Our website will also display the goals that our client will put in place to show the progress of all students. With that in mind, this document has broken down each functional

requirement to help us achieve our own milestones in creating this website. We have set up a project plan which lays out when each of the milestones should be completed. With our key requirements being laid out, we have given ourselves a huge boost in our entire project progress.

We all are excited to be creating this digital portfolio for NAU, where we ourselves can even display our milestones from being a student here at NAU. After laying out our entire project plan, we can not wait to get started on implementing all our requirements and technologies. Before we know it, NAU will have its own website dedicated to helping students keep track of their achievements, set goals for students, and help students get a job for when they graduate.