Requirements

November 12, 2024

Lumberjack Balancing

Project Sponsor: Dr. Scot Raab

Project Mentor: Paul Deasy

Team Members: Riley Burke, Cristian Marrufo, Sergio Rabadan, Braden Wendt

Acceptance Statement

Accepted as baseline requirements for the project

_____	_____
Team	Client

Table of Contents

Introduction

“Lumberjack Balancing” operates within the higher education sector, specifically focusing on faculty workload management in universities. The education industry continuously evolves, seeking to optimize administrative processes that support academic excellence. At large institutions like Northern Arizona University (NAU), workload distribution for faculty members is crucial. It ensures compliance with institutional policies and provides equitable workload allocation, directly impacting faculty performance and institutional efficiency.

The sponsor of this project, Dr. Scot Raab, is an associate dean at NAU. He, along with other associate deans, is responsible for assigning workload percentages for faculty members based on various criteria, including class assignments and credit hours. This process, currently handled manually, is labor-intensive and prone to errors due to the volume and complexity of data. Each semester, hundreds of faculty members and their course assignments must be accurately represented to support NAU's operational and academic goals. The existing process demands significant time and meticulous attention to detail, impacting the administrative efficiency of the university and the faculty’s ability to focus on critical tasks.

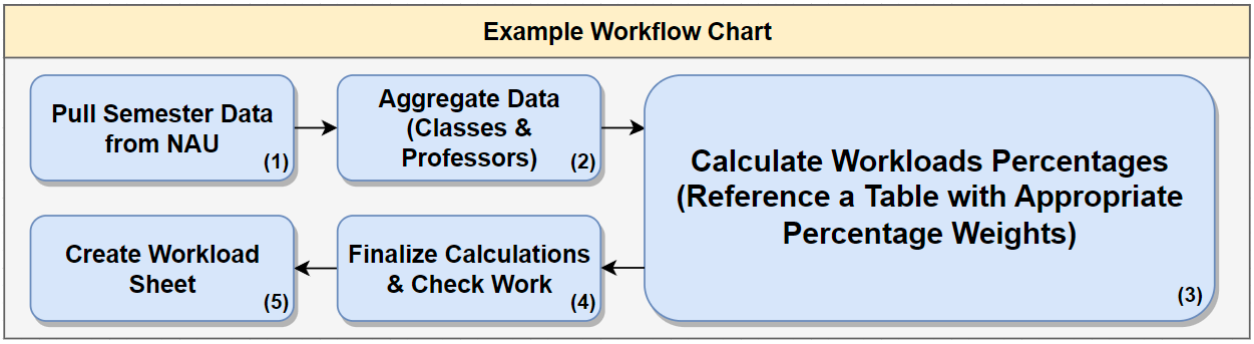
To address these challenges, our team proposes developing an application that will automate the faculty workload calculation process for NAU. The system will streamline workload assessments by automatically parsing data from Excel files, condensing it, and generating a detailed report. By reducing manual calculations, the application will enhance both accuracy and time efficiency. Additionally, this project will deliver a flexible solution where conditions and weights for workload calculations are easily configurable by the associate deans through a customizable Excel sheet. This flexibility will allow the application to adapt to any future changes in NAU’s workload policies without requiring extensive code updates.

Through an intuitive graphical user interface (GUI), the application will support essential functionalities such as data uploads, calculation initiation, and report generation, designed for ease of use by non-technical staff. This approach not only minimizes training requirements but also ensures that associate deans and their teams can efficiently utilize the tool in their daily workload management tasks.

In essence, the "Lumberjack Balancing" project will provide a scalable, user-friendly, and accurate solution for NAU’s faculty workload management, alleviating the burden of manual calculations while fostering a balanced academic environment. By leveraging data automation and user-friendly interfaces, the system will allow administrators to allocate their time more effectively, fostering improved productivity across NAU’s academic departments.

Problem Statement

The current faculty workload calculation process at Northern Arizona University (NAU) is an essential but challenging task that requires significant manual effort from the associate deans, such as Dr. Scot Raab. This process begins each semester with the extraction of data from NAU's systems, capturing essential details about each professor's assigned courses **(1)**. These details include the types of classes (e.g., lectures, labs), credit hours, and the specific faculty roles associated with each course. The data is then aggregated to compile each professor's teaching assignments into a comprehensive workload profile **(2)**. Once the data is aggregated, the associate deans must apply predefined workload percentages based on institutional policies. This calculation requires referencing an external table containing percentage weights that vary by course type and faculty role **(3)**. The deans apply these weights manually to each faculty member's aggregated workload data to ensure policy-compliant distribution. After the initial calculations, the deans carefully review the results to confirm accuracy and alignment with NAU's standards **(4)**. Finally, a workload sheet is created, summarizing each professor's assigned workload for transparency and administrative record-keeping **(5)**.



Identified Problems in the Current Workflow

While this workflow is structured, the manual nature of the process introduces several critical inefficiencies and risks:

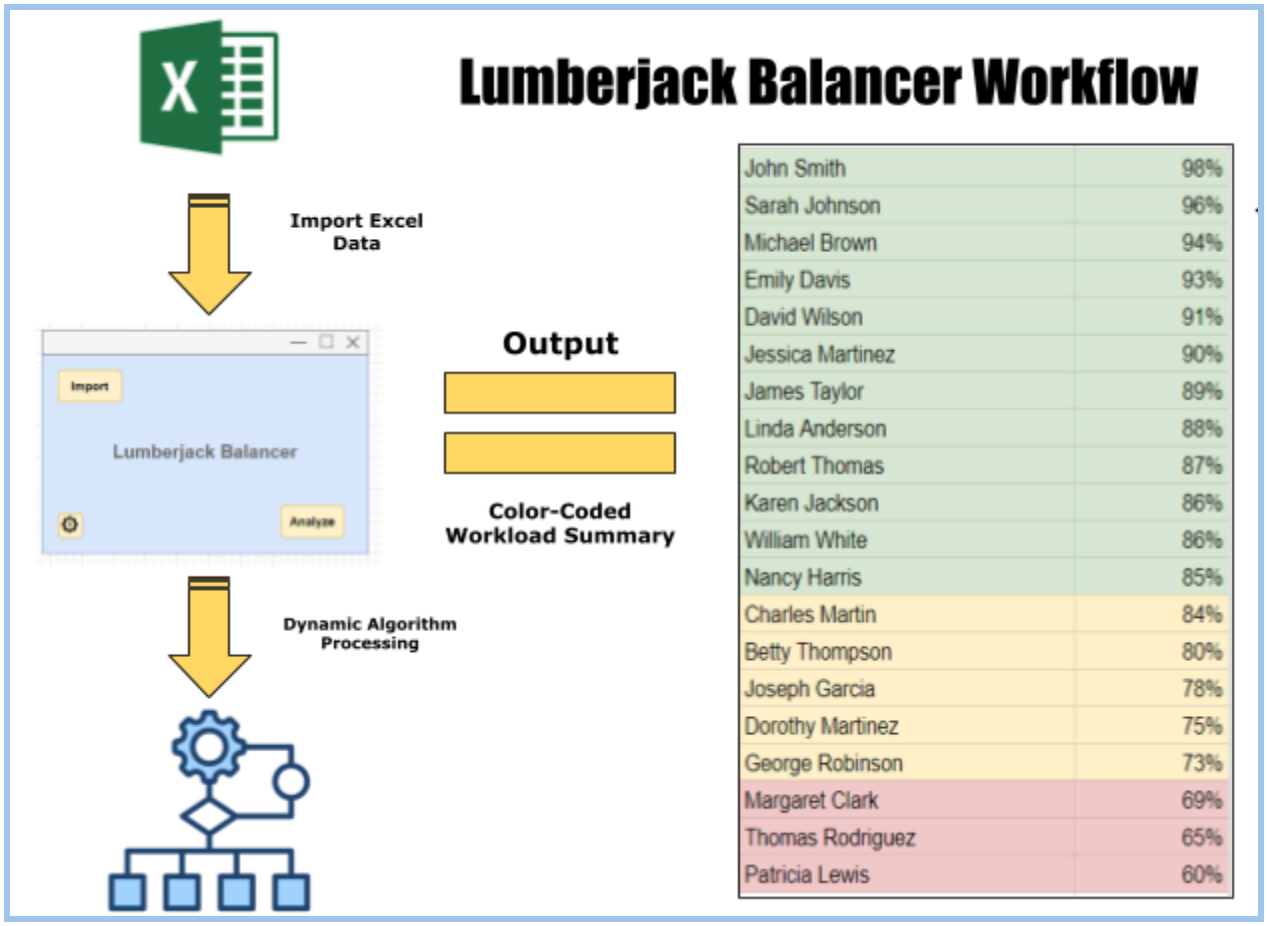
- **Time-Consuming Process**: The manual steps required for data aggregation, calculation, and validation consume considerable time, limiting the associate deans' availability for other responsibilities.
- **Error-Prone Calculations**: With large volumes of data and intricate workload rules, the process is susceptible to human error in both data entry and calculation. Even minor inaccuracies can lead to policy misalignment and workload imbalances.
- **Lack of Real-Time Error Checking**: Errors in data input or policy application are typically identified only at the final review stage, leading to multiple rounds of corrections and rechecks, which prolongs the overall process.
- **Entirely Manual Reporting and Documentation**: Generating and sharing workload reports with stakeholders involves additional manual steps, which introduces delays and adds to the administrative burden.

Solution Vision

As a response to the various issues presented by the current faculty workload assignment workflow, we have proposed the development of a Python based desktop application that will automate the workload calculation process and present the university with the necessary features that will ideally alleviate the problems inherent with a fully manual system. To begin with, the application we have started to design will make use of a specially constructed algorithm which will automatically aggregate the necessary data points used in the workload percentage calculations from the imported faculty workload data for a respective semester period. The format of the data that is extracted from the university's database systems is expected to be in Microsoft Excel files, and should include relevant information pertaining to each faculty member such as academic responsibilities, class curriculum, and expected credit hours based on the type of class the given faculty member is required to teach. As one would assume, this process is incredibly time consuming and generally exhausting if performed manually, as such, our system aims to reduce the time and effort required to conduct these data analysis operations significantly. Furthermore, as our system will incorporate an4 algorithm that utilizes a computer’s operational capabilities to execute any given mathematical process, no direct human involvement is required in the mathematical calculations or individual data entry, resulting in an intrinsic decrease of errors associated with human interaction. What’s more, in order to further reduce errors that might arise during the workload calculations, validation of data inputs will be conducted in real time, preventing inaccurate workload allocations and additional reviews and corrections of the produced results. Lastly, a workload report will be generated in a similar Excel format as the data inputs, clearly displaying and summarizing pertinent information relevant to each staff member. The report will be generated automatically once the necessary operations have concluded, and should be available to export at any time as desired by the user.

Key Objectives:

- **Efficient Data Analysis**: Perform a swift and accurate analysis of the Excel data to deliver meaningful insights.
- **Adaptable Algorithm**: Design the algorithm to be flexible, enabling it to handle potential changes and scale smoothly.
- **User-Friendly Interface**: Create an accessible and intuitive application to ensure ease of use for all users.



Green: Faculty members working at or above 85% of their expected workload.
Yellow: Faculty members working between 70% and 85% of their expected workload.
Red: Faculty members working below 70% of their expected workload.

Project Requirements

Domain-Level Requirements

The Lumberjack Balancing application, whose primary functions consist of faculty workload allocation, has been designed to address the inefficiencies and inaccuracies of the manual workload assignment process that is currently employed at Northern Arizona University, as has previously been described in the document. Lumberjack Balancing is intended to automate the calculations that are required by the faculty workload institutional policies that have been established, ensure real-time validation of input data to minimize various errors that might arise, and generate comprehensive workload reports that adhere to the percentage allocation algorithm which is ultimately decided by a selection of different criteria. The subsequent tools that our application will provide are expected to aid in the long-term planning and resource allocation demands that directly inform future University investments of faculty efforts in

alignment with NAU's mission and strategic goals. The following domain-level requirements will guide the development of our system:

- **D1.1**: Automated Workload Calculation: Automatically assign teaching workloads to faculty based on faculty workload policies specific to their academic roles and teaching responsibilities. This eliminates the need for manual calculations.
- **D1.2**: Real-Time Data Validation: Validate input data in real-time in order to minimize errors and ensure consistency across all entries, while also ensuring that workload assignments adhere to the defined policies and constraints.
- **D1.3**: Centralized Data Management: Provide a centralized platform for importing, processing, and managing semester-specific data for faculty and their workload profiles. This will reduce the need to employ any other combination of superfluous platforms and tools that serve similar functions as our applications, though in a less specialized and consolidated manner.
- **D1.4**: Customizable Workload Policies: Support dynamic workload policies to accommodate changes in university regulations or department needs. The application will allow administrators to explicitly define, store, and update workload policies for different faculty types and academic roles, ensuring flexibility as institutional policies evolve.
- **D1.5**: Comprehensive Reporting: Generate clear, accurate workload summaries that can be exported in a suitable format that satisfies the university's needs. The reports created will directly highlight and indicate any discrepancies exhibited by the workload assignments, such as an overload or underload of credit hours, providing the administrator with actionable information for future changes and resource allocation.
- **D1.6**: Scalability for Long-Term Planning: Offer the university predictive analytics tools and data visualization features for identifying staffing demands and inform the university of future resource allocation obligations.
- **D1.7**: User-Friendly Interface: Ensure the system is intuitive and accessible for users with varying technical knowledge, such as department administrators and faculty members.
- **D1.8**: Security and Reliability: Protect sensitive faculty data by implementing secure data handling practices and ensuring compliance with applicable educational data regulation standards.

Functional Requirements

This section outlines the functions our system must provide, beginning with high-level capabilities and expanding into more detailed low-level requirements.

High-Level Functional Requirements

- **H1.1**: Data import and Management: Import workload data from the standard file format used by NAU, such as Excel, containing faculty assignments, course details, and relevant semester data. Allow administrators to manage faculty profiles, including career

status (e.g., tenured, career-track), teaching requirements and thresholds, and workload history. Provide a dashboard summarizing workload data for the current and previous semesters.

- **H1.2**: Policy Configuration: Enable administrators to define workload policies, such as the maximum and minimum credit hours assigned to faculty based on their role. Support for different policy templates associated with standardized configurations for different departments. Maintain a workload policy version history in order to allow for comparisons over time.
- **H1.3**: Workload Calculation: Automatically calculate workloads for all faculty based on input data and workload policies. Highlight discrepancies such as overloads or underloads, which occurs when faculty is assigned a workload below or above the established threshold.
- **H1.4**: Error Validation: Validate input data for completeness in the absence of faculty names or course assignments. Identify and emphasize data inconsistencies, such as duplicate entries or contradictions between assigned and calculated workloads. Provide actionable error identification methods that allow the user to easily find a resolution without necessitating any other external tools.
- **H1.5**: Report and Exporting: Generate workload summary reports divided into faculty, department, or semester, accompanied by any adequate visual representation. Export reports in user-friendly formats, such as Excel, for ease of distribution and archival purposes.

Low-Level Functional Requirements

- **L1.1**: Data Integration and Compatibility: Ensure compatibility with existing university databases and data formats to allow seamless integration with other pre-established systems. Support importing data in the standardized file formats used by the university, like Excel.
- **L1.2**: User Interface Features: Include intuitive menus, field options, and validation prompts to streamline data entry.

Performance Requirements

- **P1.1**: Response times: The system must load and process Excel Files containing this workload data in a short amount of time. Larger files shall not exceed a minute of processing time to ensure a smooth user experience.
- **P1.2**: Calculation accuracy: The workload calculations must maintain an error rate of less than 0.1% to ensure that NAU policies are not violated.
- **P1.3**: Scalability: The system will be able to handle 1000-2000 user profiles per semester to ensure a reliable performance even as the number of faculty members change.
- **P1.4**: User interface: The system interface will be easy to use allowing for new users to complete basic functions with minimal to no guidance.

Environmental Requirements

- **E1.1**: Platform capability: The system will be compatible with Windows and macOS operating systems to accommodate users. Cross platform ensures a flexible deployment.
- **E1.2**: Data Format: The application will be able to import and process Excel Files and export Excel Files.
- **E1.3**: Hardware Requirements: The application will be able to run smoothly with a minimum of 8 GB of ram and 2.0 GHz processor.
- **E1.4**: Maintenance: The system will be able to be maintained easily and changed as needed.
- **E1.5**: Data Privacy: Sensitive faculty data will be protected. The application will not save any of the data inputted through it.

Potential Risks and Impact Analysis

The development of an automated faculty workload calculation system for Northern Arizona University (NAU) aims to reduce errors, streamline processes, and enhance efficiency. However, this project also introduces potential risks that could impact the system's reliability, user satisfaction, and overall success. Here, we outline the most relevant risks, analyzing their likelihood, potential consequences, and the broader implications for NAU and its administrative processes.

Calculation Errors in Workload Percentages

- **Risk Description**: Errors in the system's calculations could lead to inaccurate workload assignments for faculty members. These errors could arise from bugs in the algorithm, incorrect interpretation of policy rules, or issues with data input from Excel sheets.
- **Impact**: Incorrect workload calculations could lead to unfair or disproportionate workloads for faculty, resulting in dissatisfaction, reduced morale, and potential disputes. In a worst-case scenario, significant miscalculations could violate institutional policies, creating compliance risks for NAU and possibly impacting faculty contracts and compensation. If left unresolved, this risk could also damage the credibility of the system, discouraging administrators from using it.
- **Likelihood**: Moderate – While rigorous testing can reduce this risk, any software involving complex calculations has a risk of occasional errors, particularly when handling large datasets with variable conditions.

Data Integrity and Input Errors

- **Risk Description**: Since the system will rely on data input from multiple Excel sheets, there is a risk of data entry errors or inconsistencies in formatting that could lead to incorrect calculations or processing failures.

- **Impact**: Errors in data input could propagate through the system, resulting in incorrect workload assessments or system malfunctions. This could require administrators to spend additional time troubleshooting and re-entering data, negating the time-saving benefits of the system. Frequent data errors could also reduce user trust in the system and lead to delays in generating workload reports.
- **Likelihood**: High – Manual data entry and formatting issues are common, especially when data is sourced from multiple files or departments. Consistent training and data validation tools could help mitigate this risk.

Inadequate User Training and Adoption

- **Risk Description**: If users do not receive adequate training on how to use the system effectively, there is a risk that they may misuse it, leading to errors, inefficiencies, or a reluctance to adopt the new tool fully.
- **Impact**: Insufficient training could result in incorrect data entry, misuse of system functionalities, or errors in interpreting generated reports. If users feel the system is difficult to use or are unclear on its functions, they may revert to the manual process, reducing the return on investment for NAU.
- **Likelihood**: Moderate to High – User adoption and training are common challenges with new systems, especially for non-technical staff. A comprehensive guide and user-friendly design will be essential to mitigate this risk.

Policy Changes and System Flexibility

- **Risk Description**: Changes in NAU's workload distribution policies or data requirements could affect the system's relevance, particularly if these changes require extensive reconfiguration of the calculation rules or interface.
- **Impact**: If the system cannot adapt to policy changes efficiently, it may become outdated or require frequent updates from developers, leading to increased maintenance costs and delays. In the worst case, an inflexible system could be rendered unusable if it cannot meet new policy requirements, forcing NAU back to a manual approach or requiring a costly overhaul.
- **Likelihood**: Moderate – University policies change periodically, but ensuring flexibility in the system's architecture and algorithm by use of the excel sheet for algorithmic calculations can help it adapt to new requirements without extensive redevelopment.

Data Privacy and Security Risks

- **Risk Description**: As the system will handle sensitive data about faculty workloads, roles, and responsibilities, there is a risk of data breaches or unauthorized access if robust security measures are not in place.
- **Impact**: A data breach could compromise the privacy of faculty information and harm the university's reputation. Additionally, any unauthorized access or tampering with workload data could lead to erroneous calculations, misreporting, or administrative conflicts.

- **Likelihood**: Low – With the totally client-side nature of the application, and the lack of streaming data to outside tools, the security concern is little to none. Any security breach of the host system would be far more dangerous.

Emergence of Competing Solutions

- **Risk Description**: Competitors may develop similar workload management systems with more advanced features or lower costs, potentially impacting the perceived value of this project.
- **Impact**: If a more robust or cost-effective solution becomes available, NAU may reconsider the need for an in-house system, potentially leading to project abandonment or a shift in investment to the competitor's product.
- **Likelihood**: Low – While competition in software development is constant, the specialized nature of this system, tailored specifically to NAU's needs, reduces the likelihood of an exact competitive match.

In summary, these risks highlight the importance of rigorous testing, user training, data validation, and a flexible algorithm to support the successful adoption and use of the automated faculty workload calculation system, Lumberjack Balancing. Addressing these risks proactively will be essential to delivering a reliable and valuable solution that meets NAU's workload management needs.

Project Plan

The project execution plan is organized to track progress across multiple main phases: completed milestones, in-progress tasks, and upcoming work. Each milestone addresses specific functional requirements or groups of requirements, contributing to the system's final functionality. Here is a list of the project as it stands:

Completed Milestones (Upper Section of Gantt Chart):

- **Team Startup (8/30 - 9/12):** The foundation was formed, establishing roles, responsibilities and project goals.
- **Initial Website (9/13 - 9/20):** A basic website structure was created displaying an initial message to inform users Lumberjack balancing will shortly begin be planned.
- **Mini Intro Presentation (9/20 - 9/27):** An introductory presentation was delivered, allowing our team to collect early ideas and refine our approach based on the insight given at the end of the presentation.
- **Team Standards & Inventory (9/27 - 10/4):** Established standards, best practices, and team workflow guidelines to establish more efficient collaboration.
- **Tech Feasibility (10/11 - 11/1):** Conducted in-depth analysis of technological approaches, addressing feasibility, potential benefits & risks, and confirming alignment with the project's functional requirements.

- **Poster Approval (11/1 - 11/8):** Created a visual project summary through a poster for an upcoming presentation.

Work In Progress (Middle Section of Gantt Chart):

- **Requirements Document (11/8 - 11/15):** Initial document that outlines all functional requirements of the project, serving as a roadmap for development. It mainly focuses on defining the functionality of different components of the system to ensure alignment between the technical specifications and the project goals.

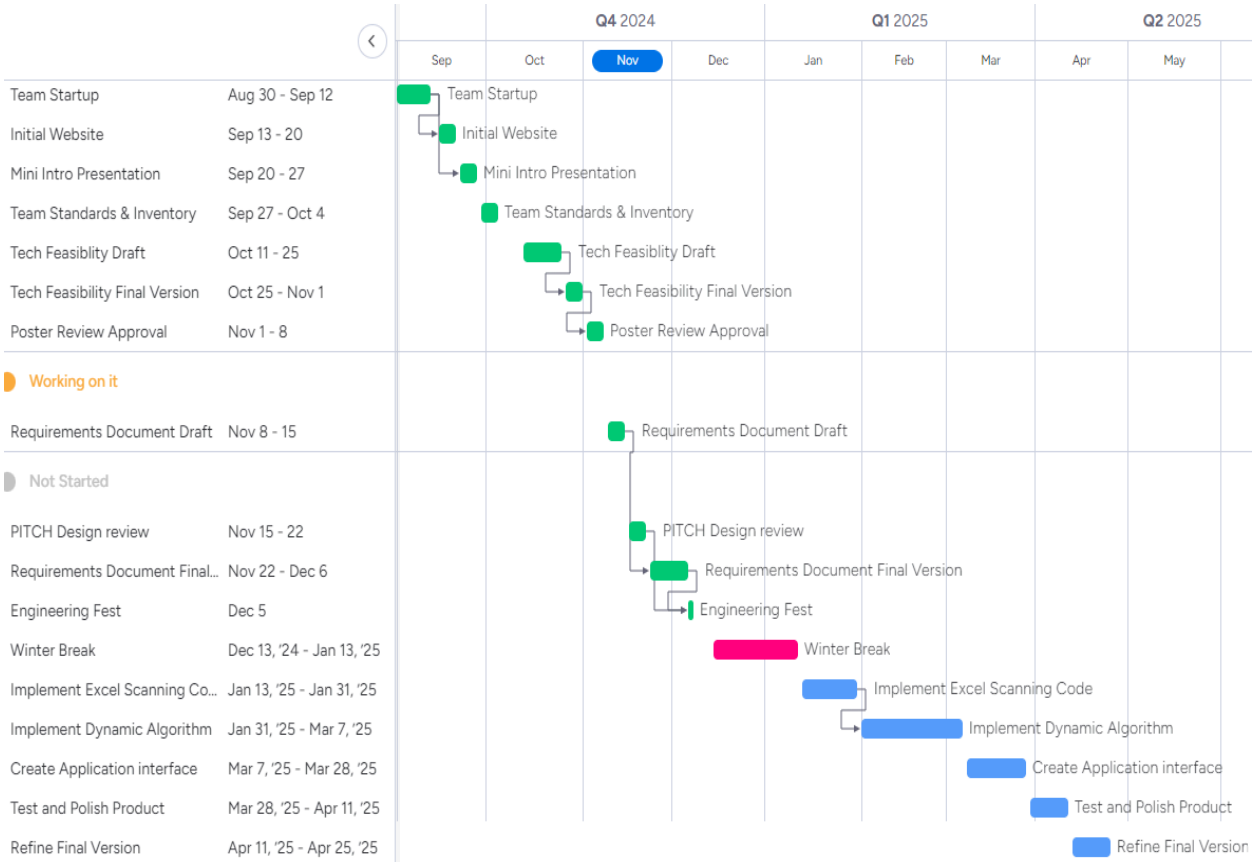
Future work (Lower Section of Gantt Chart):

Short-term:

- **PITCH Design Review (11/15 - 11/22):** A technical review session will be held, where design choices will be evaluated. This review will ensure that design meets the functional requirements.
- **Engineering Fest Preparation (Nov 23 - Dec 4):** During this period, the team will consolidate and synthesize all project information to prepare a comprehensive presentation. This preparation aims to highlight the key features, technologies, and overall project structure, effectively communicating our goals and objectives. By focusing on these elements, we ensure that we are well-prepared to present our work confidently and clearly during the Engineering Fest.
- **Engineering Fest (Dec 5):** The Big Day. This event is an opportunity to showcase our project to a wider audience, including industry professionals and peers. It serves as a platform to gather valuable feedback, validate our progress, and demonstrate the practical impact of our work.

Long-term (Subject to change):

- The Gantt chart outlines additional long-term milestones that must be completed to achieve a functional final product. These include implementing key features, refining the application interface, testing, and final polishing. While tentative dates have been assigned, the timing of each milestone may adjust as the project progresses. This flexibility allows us to adapt to unforeseen challenges or expedite tasks that require less time, ultimately ensuring quality and functionality in the finished product.


Q4 2024
Q1 2025
Q2 2025
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Apr
May
Team Startup
Aug 30 - Sep 12
Initial Website
Sep 13 - 20
Mini Intro Presentation
Sep 20 - 27
Team Standards & Inventory
Sep 27 - Oct 4
Tech Feasiblity Draft
Oct 11 - 25
Tech Feasibility Final Version
Oct 25 - Nov 1
Poster Review Approval
Nov 1 - 8
Working on it
Requirements Document Draft
Nov 8 - 15
Not Started
PITCH Design review
Nov 15 - 22
Requirements Document Final...
Nov 22 - Dec 6
Engineering Fest
Dec 5
Winter Break
Dec 13, '24 - Jan 13, '25
Implement Excel Scanning Co...
Jan 13, '25 - Jan 31, '25
Implement Dynamic Algorithm
Jan 31, '25 - Mar 7, '25
Create Application interface
Mar 7, '25 - Mar 28, '25
Test and Polish Product
Mar 28, '25 - Apr 11, '25
Refine Final Version
Apr 11, '25 - Apr 25, '25

Conclusion

The "Lumberjack Balancing" project addresses a critical need within Northern Arizona University's (NAU) faculty workload management system. With the university's current process being labor-intensive and error-prone, the proposed automated solution aims to streamline workload calculations, enhance data accuracy, and reduce administrative burdens. This project is essential for ensuring fair and policy-compliant workload distribution, which directly influences faculty satisfaction and NAU's operational efficiency.

In this document, we outlined the current manual workflow used by NAU's associate deans, identifying key pain points such as time inefficiency, error susceptibility, and lack of flexibility. Our solution leverages an adaptable algorithm, real-time data validation, and a user-friendly interface to automate this process. Designed specifically to handle NAU's workload policies and adaptable to future changes, this system will automate data aggregation, perform workload calculations, and generate detailed reports, thereby alleviating the workload of NAU's administrative staff.

Through this requirements document, we have defined the project's goals, outlined the anticipated functionality, analyzed technological feasibility, and assessed potential risks. These elements provide a solid foundation for the ongoing development and refinement of the system, ensuring that it meets NAU's needs both now and in the future.

The insights gathered here underscore our commitment to delivering a robust, flexible, and accurate solution for NAU's faculty workload management. With thorough planning, risk mitigation strategies, and a focus on user-friendly design, we are confident that this project will achieve its intended outcomes, ultimately improving efficiency and supporting NAU's commitment to academic excellence.