1. **Introduction**

We are a senior capstone team at Northern Arizona University, consisting of team members Julian Bell and Remy Brandriff, with team mentor Jun Rao. Our project is entitled ‘Environmental Laboratory Informatics and Management System.’

The Civil and Environmental Engineering program at NAU operates an Environmental Engineering (ENE) laboratory which facilitates and hosts scientific research in the field; this research may be performed by students, as part of a course or as independent research, faculty, or industry professionals using university resources. The ENE lab supports research into, among other topics, climate change, water safety, and sustainability, which contribute to global efforts of preserving the environment. When there are failures or issues with research or data, which can occur in any number of ways, that endangers the validity of the results, and endangers the field of environmental engineering. Consequently, this may affect life on Earth as a whole.

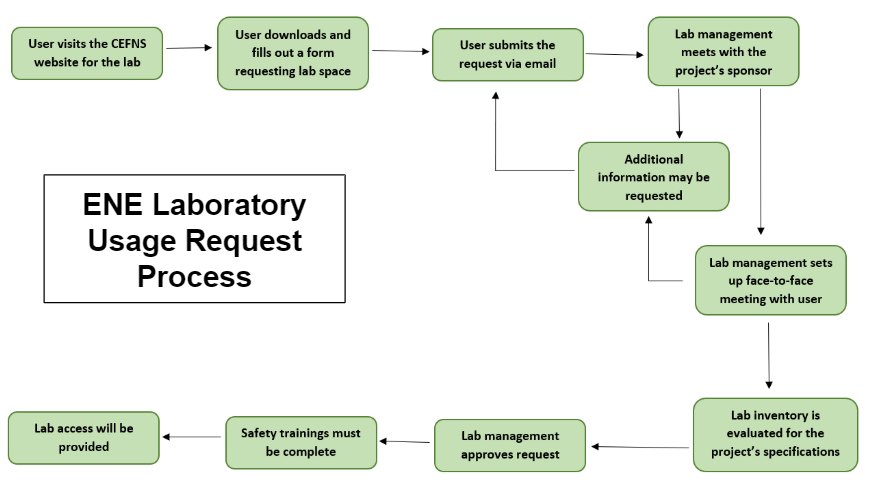
Our sponsors are Dr. Terry Baxter, a professor in the Environmental Engineering program at NAU, and Michael Kelly, the Environmental Health and Safety staff associated with the ENE lab. Dr. Baxter oversees the ENE lab along with the lab manager, Gary Slim, E.I.T., as well as supervising student research. His and students’ research performed in the lab contribute to global EE efforts as a whole, so ensuring that the lab operates efficiently and safely is critical for avoiding or mitigating, for example, failed experiments, faulty data, dangerous chemical interactions, and potential health risks.

**II. Problem Statement**

Efficient and capable management is required for any work space to operate optimally, and the ENE lab is no exception. The research performed in the ENE lab contributes to global efforts to preserve the environment and ensure life on Earth, but the current workflow for lab usage and management has failed to keep up with lab growth. In order to use the lab, a party must submit a request form. For lab management, they’re responsible for managing and evaluating these requests, tracking the chemical inventory and lab materials, and supervising the physical space.

Our client has expressed dissatisfaction with the current methods and processes that exist for lab operation, including how parties request lab usage and how lab management supervises the space. The current system runs almost entirely through paper and email, and this lack of centralization has proven to be detrimental to safe, efficient lab operation.

Currently, the processes for parties requesting use of the ENE lab is primarily through email and is a long and circular process.



*ENE Laboratory Usage Request Process*

1. Parties who would like to request usage of the ENE lab visits the lab website hosted by the College of Engineering, Forestry, and Natural Sciences, and download the request form—a document the party will fill out with the relevant information
2. The completed form is then emailed to the lab management
3. Lab management reviews the request
   1. They meet with the project sponsor, which for students may be a faculty supervisor, course instructor, etc.
   2. This process may go back and forth repeatedly until lab management has determined they have all the information they need
4. Lab management arranges a face-to-face meeting with the requesting party
   1. This meeting may result in the requesting party having to submit more information
5. Availability of chemical inventory and project materials will be evaluated to determine the feasibility of the project
6. Lab management may approve or deny the request for lab usage, or they may require the party develop their project more before usage may be granted
7. All lab safety trainings must be completed
8. Lab usage may be granted

Alongside this paper- and email-based request system, there are two systems associated with the ENE lab: A.C.I.D., the chemical inventory, and BioRaft, the Environmental Health and Safety trainings database.

The online automated chemical inventory database (A.C.I.D.) is the chemical inventory system used by the ENE labs to track the different chemicals used in the labs. Users can look up which chemicals are available in the lab and some of the health and safety information associated with those materials, and lab management can update the system according to the lab space. This software was created and is maintained by NAU’s Information Technology Services (ITS) department, and is primarily used by the Environmental Health and Safety staff to ensure proper and legal use of chemicals in the university facilities. It has been expressed to us that the A.C.I.D. system may be improved upon, but this is not the primary goal of the project.

BioRAFT is a third-party software that facilitates and tracks the health and safety trainings of NAU personnel. This system is used to ensure that everyone in the lab has completed the required trainings to help avoid chemical interactions, health and safety violations, and overall poor behavior in the lab space.

**III. Solution Vision**

To solve our client’s problem, we propose a web application which will facilitate easy management of lab spaces and simplify the existing request process for both requesting parties and lab administration.

This is largely composed of two major components, the web interface and the database system which serve as the “front-end” and “back-end” respectively. The web interface itself, what the user sees, will allow for account registration, submission of lab use requests, and a way to view the status of said requests and the chemical inventory. The latter element forms the “back-end”—our web application will be run off of a robust database system that stores lab chemical and safety information, requests, and other necessary information as needed. Our application will consist of the following:

* A web application to streamline and automate the various aspects of the ENE lab
* Front-end: web interface
  + Register for user account
  + Submit requests for use of lab space
    - Check on said requests and see added information from faculty
  + Look at chemical inventories with proper safety and informational sheets as needed
  + Roles for administrators to add to inventories, approve requests, correspondence, and so forth
* Back-end: robust database system
  + Chemical inventory
  + Inventory of requests
  + Integration with A.C.I.D. (pre-existing, ITS-developed inventory system for campus labs)
  + Integration with BioRaft (safety and training module)

The data that this web application uses will be provided from an administrative source; those with an aforementioned admin role will be able to feed chemical data for inventory purposes to the application, and handle the requests sent by standard users, which are the two major points of the application itself.

The system will automate the existing processes, therefore reducing the overhead efforts in coordinating lab space and increasing lab productivity overall, and will streamline lab management into one easy location. Our intent is that the sponsor will find themselves with a much better system to handle this complex, multi-faceted process that is lacking in its current state.

We know that there may be aspects lacking in the way of boundless features (as in a desktop/software application) or immediate availability (in the case of a mobile application, for example), but the use of a web application is what we consider a proper middle ground for achieving each of the requirements set out and planned between the team and the client. The client has mentioned the potential for use of this system beyond NAU, and expressed the desire to commercialize it to support the lab’s further operation. It is our belief that our solution will be ideal in facilitating these possibilities.

**IV. Project Requirements**

The requirements for this project have been determined through discussion and collaboration with our clients and the team associated with the project over the course of this semester so we can ensure we develop a solution that fulfills their needs and solves the problem outlined earlier in this document. We have also used documents that outline the current process and policies in our analysis and requirements acquisition to ensure they are accurately implemented in our system.

In this section, we outline a sensible and comprehensive set of requirements we have created based on the past semester’s work on this project. The purpose of these requirements are to guide the project’s development for the rest of the capstone, and to serve as the rubric by which the project will be evaluated by ourselves throughout development, and by our client as the measure by which we have satisfied their needs.

Through these requirements, we have detailed a robust web application that facilitates lab usage requests and inventory management, in cooperation with the existing systems, that is built in mind for future ITS support and the potential for expansion should the client choose. There are two aspects of the project requirements: Domain requirements, and system requirements. Furthermore, the system requirements consist of three types: Functional, non-functional, and environmental. We will outline all of these in this section.

**A. Domain Requirements**

Domain requirements are the overall user requirements, which guides how this system will operate at the domain level and lay out the features the user needs. There are the requirements that explain what the user wants to do with the system in order to satisfy their needs, and the criteria by which our system’s performance at the user level is evaluated.

|  |  |
| --- | --- |
| **#** | **Requirement** |
| DR 1 | The system will be simple to use |
| DR 2 | The system will be easy to extend |
| DR 3 | The system will be accessible |
| DR 4 | The system will simplify the existing lab usage request process |
| DR 5 | The system will simplify lab management for administration |
| DR 6 | The system will enable the Standard Operating Procedure of the ENE lab |
| DR 7 | Administrators and users can register accounts |
| DR 8 | Users can manage their accounts, edit their settings, make requests for lab usage, manage their requests, view the chemical inventory, request new inventory or report a problem, and contact lab administrators |
| DR 9 | Administrators will be able to manage user requests, manage the lab space in the system, and contact users. |

These requirements will be carried out by the client-side operations in the system. Developing this system with the requirements that it is easy to use and extend, with high usability, will ensure the system will be able to optimally serve the lab in the long run. Usability is part of a broader term “user experience,” and describes the ease of access and/or use of the system, as determined by its features. A system with low usability may ultimately frustrate both users and lab administration, and lead to a system that is actually *less* usable than the current one.

As such, we will ensure a simple system that is easy to use, such that:

* It is easy for the user to start using the system, and easy for the user to become familiar with it quickly
* It is easy for the user to achieve their goal, whether that is managing the lab space or requesting use of the lab, using the system
* It is easy for the user to come back to the system and start using it again

With this in mind, we will maximize responsive design in develop, in order to fully realize an easy to use, simple system with high usability. We will also develop the system to be highly accessible so we can maximize the system’s usability and make it accessible to all users. We may use the Web Content Accessibility Guidelines (WCAG), along with other resources, to ensure an accessible system by maximizing perceptibility, operability, understandability, and robustness. This may include:

* Making functionality accessible by keyboard and other assistive technologies
* Designing with color blindness and visual impairment-friendly palette
* Designing a highly readable application
* Providing alternatives for content and media
* Using transcriptions of audio where necessary
* Providing methods for navigation
* Maximizing compatibility with current and future assistive technologies
* Making the web app appear and operate predictably
* Providing support for correction and avoidance of user mistakes
* Ensuring that, should an element not be accessible, the rest of the relevant content is

Much of our work to fulfill these domain requirements will involve providing ITS, future capstone groups, and anyone else who will be supporting the system a solid framework to continue building on, which will not compromise accessibility and usability.

It is also necessary that we develop the system within the requirements that it simplify both lab usage requests and lab management, given that these are the problems this system will exist to solve. These domain requirements will be fulfilled and explained more in-depth in the System Requirements section.

This system must also follow and facilitate the Standard Operating Procedure for the ENE lab, which are the guidelines by which the lab operates within the Civil and Environmental Engineering program. They detail the process for requesting lab space, and the policies and procedure for lab access and usage, and as such, define the domain in which our system will operate.

**B. System requirements**

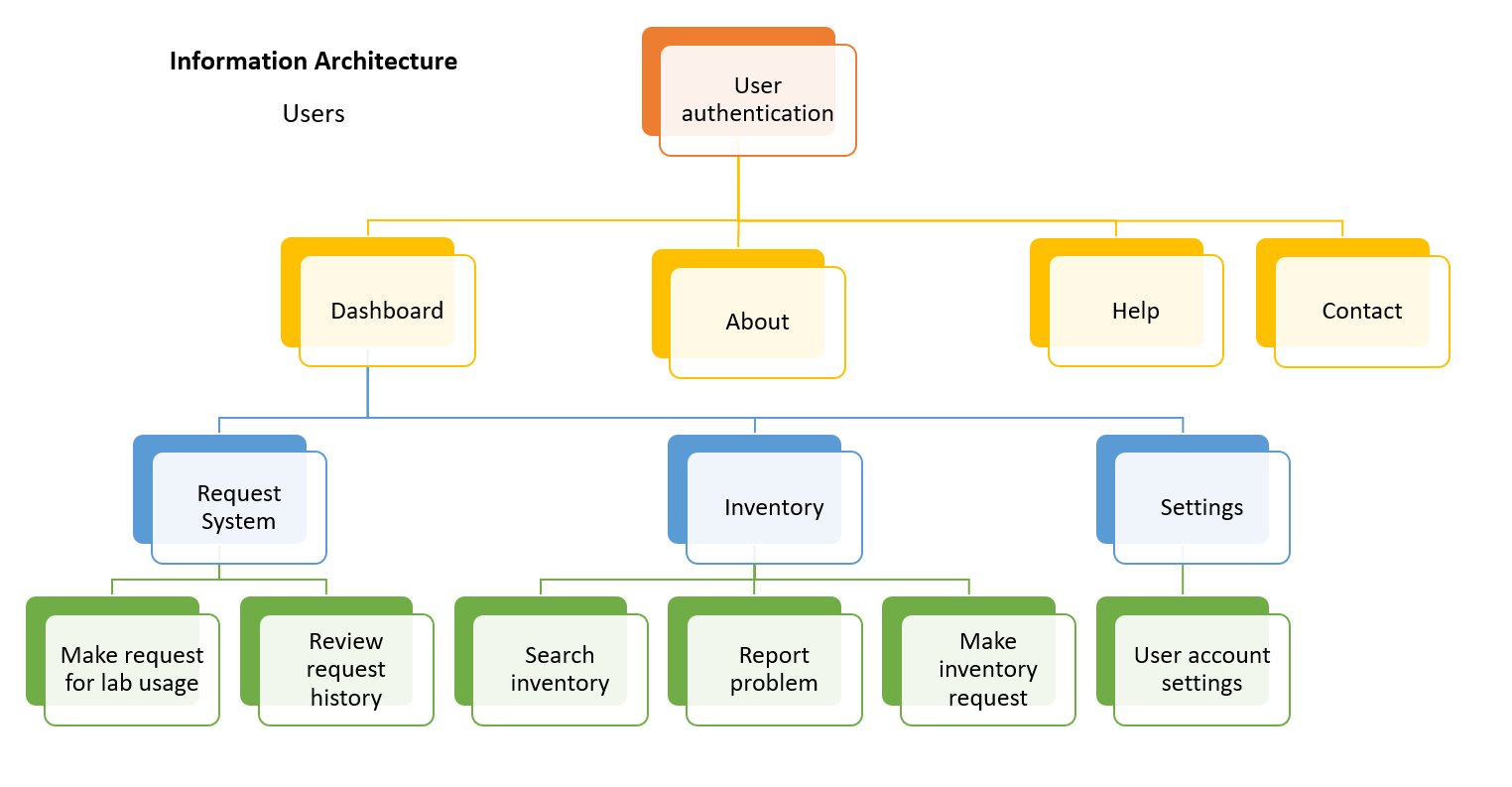
In order to run smoothly and operate efficiently, the system must fulfill system requirements as well as user requirements. These system requirements define the capabilities of the system and how it functions to fulfill the user requirements detailed in the previous section, and they fall into three categories: Functional, non-functional, and environmental.

Each of these sections contribute to the design and development of a cohesive system that satisfies our client’s needs and ensures safe and efficient operation of the ENE lab.

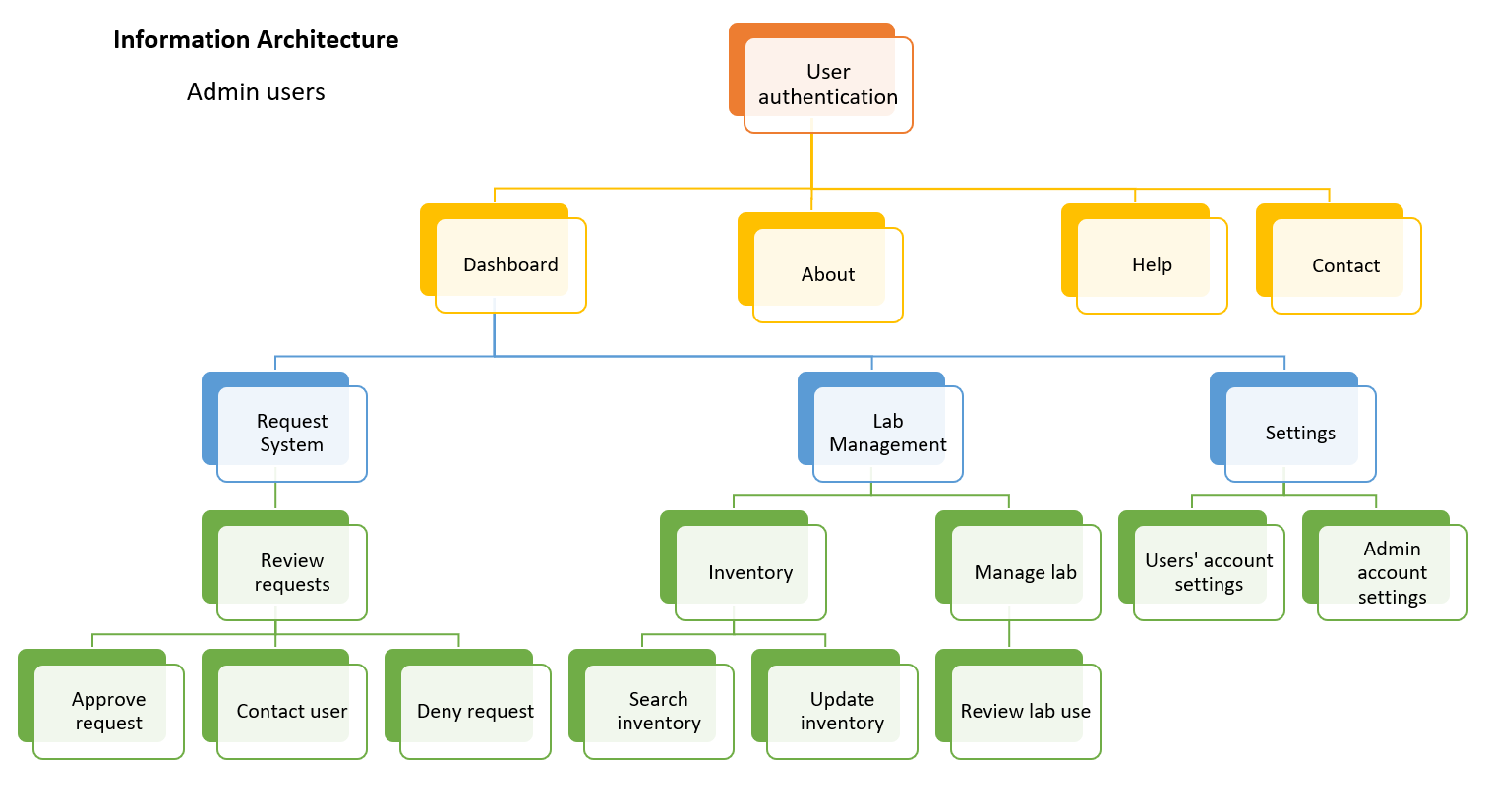
**1. Functional Requirements**

Functional requirements define the system capabilities and functionality. They describe a behavior or function the system must be able to perform—specifically, function requirements determine *what* the system should do. These requirements do not define *how* these functions are to be implemented, merely detailing the criteria by which the system’s behavior will be evaluated.

We have detailed requirements for the system as a whole, as well as specifically for the different user types.



*A detailed tree of the user’s information architecture in the system*

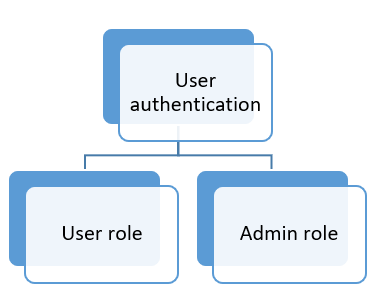
**

*A detailed tree of the administrative user’s information architecture in the system*

|  |  |
| --- | --- |
| **#** | **Requirement** |
| FR 1 | The system will process user authentication |
| FR 2 | The system will store and retrieve data with a database backend |
| FR 3 | The system will integrate with the existing BioRAFT system for health and safety trainings |
| FR 4 | The system will integrate with the existing A.C.I.D. system for inventory interaction |
| FR 5 | The system will support two types of users: Standard and Admin |
| FR 6 | All users can create accounts |
| FR 7 | All users can manage their account |
| FR 8 | All users can request new inventory |
| FR 9 | Standard users can submit requests for lab usage |
| FR 10 | Standard users can manage their existing requests |
| FR 11 | Standard users can view their request history |
| FR 12 | Standard users can search the lab chemical inventory |
| FR 13 | Standard users can contact administrative users |
| FR 14 | Standard users can report problems with the chemical inventory |
| FR 15 | Standard users can schedule face-to-face meetings with admin users |
| FR 16 | Admin users can manage standard user accounts |
| FR 17 | Admin users can manage the lab inventory |
| FR 18 | Admin users can contact standard users |
| FR 19 | Admin users can schedule face-to-face meetings with standard users |
| FR 20 | Admin users can manage how the lab space is represented in the system |
| FR 21 | Admin users will be able to access and manage the database backends of the system |
| FR 22 | The system will utilize a web form to receive and process the lab usage requests from standard users |
| FR 23 | Admin users can approve or deny lab requests |

We detail the functional requirements defined above here. These functional requirements will be carried out by client-side operations for the user, and powered by server-side operations in the software and database.

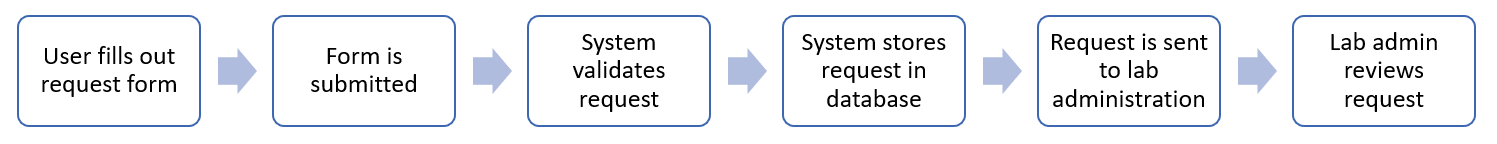
The system will process user authentication of registered NAU users requesting usage of the ENE lab, and allow them to register accounts with the system. User registration is necessary to use the system, and will facilitate account management, such as adjusting settings and changing details. All users will be able to do this, whether they are standard users or administrative. User information will be stored securely in a database backend, which will support two types of users: Standard and Admin. This allows the system to operate in different modes for the different users. It also allows administrative users to edit user information.



*The system will have two roles: User and Admin*

The database will also store lab usage request information submitted by users, and will facilitate lab administors to manage how the lab is treated by the system. Administrative users will be able to access and manage the database backends of the system, which will facilitate many of the other functional requirements associated with administrative users. This will include managing lab usage requests, user accounts, and how the lab space is represented in the system, which may include editing lab availability and indicating which spaces are off-limits. This will also help standard users in creating their lab usage requests, as it will help give an up-to-date and accurate idea of what the lab looks like at any moment.

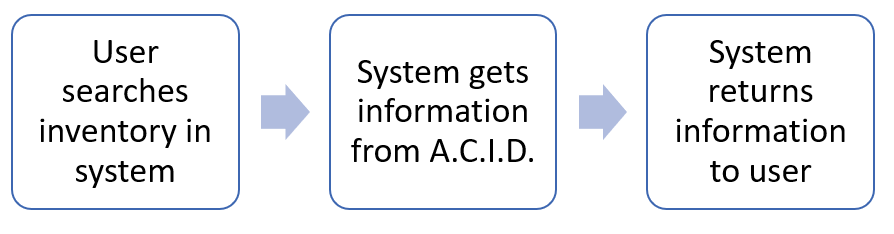
The system will utilize a web form in order to receive lab requests from users, facilitating the process by providing all of the fields, indicating which fields are required, and sending the form to lab management. Users will be able to submit requests for lab usage using the form, which sends the information to the system’s database; the system authenticates and validates this request, and sends on the request to the lab administration for review. Users will also be able to view their request history, including open and closed requests, and edit open ones for resubmission.



*The process for a lab usage request submission from user to lab administration*

Administrative users can approve or deny lab requests submitted by standard users, which indicates whether the requesting party can utilize the ENE lab space for their project or not.

The system must integrate with the existing BioRAFT system NAU uses to manage health and safety trainings in order to ensure a safe and legal lab environment, and it must integrate with the existing A.C.I.D. system for inventory interaction, which will allow the system to fulfill other functional requirements. These requirements are crucial for administrators’ to fully process lab usage requests, and manage the lab spaces. This also facilitates users being able to search the A.C.I.D. system for the lab inventory, such as checking to see if a specific chemical is in stock in the correct amount, and will let lab administration monitor the inventory. Users will also be able to request new inventory to be added to the A.C.I.D. system, which can only be done by lab management.



*The process for the system’s integration with the A.C.I.D. program*

Lab management will be able to manage the lab inventory, which may include adding or deleting inventory from the A.C.I.D. system, or editing the information associated with an entry.

Users will also be able to report problems with the inventory, such as A.C.I.D. reporting the presence of chemicals not physically in the lab. Lab management will be responsible for updating the system, as they are now, but it will give users greater power in the process, create a record of problems with the inventory from the users’ point of view, and make it easier for lab management to know what needs to be fixed.

Standard users will be able to contact lab administration, and vice versa, lab administration needs to be able to contact standard users. As part of the lab usage request process, it’s necessary for users to meet with lab management in a face-to-face meeting, and the system will help facilitate this. An actual scheduling system would be the work of a future update, not initial development, but we’d like to set the stage for the possibility.

**2. Performance (non-functional) Requirements**

Non-functional requirements, also called ‘performance requirements,’ define the system’s performance. They describe how the system should behave, and how the system works. Because this web application will be dynamic, constantly changing, and always responding to the needs of lab users and administrators of the system, we have a few essential pre-defined requirements of this regard that we aim to achieve. In these requirements, we need to set out the criteria by which our system may be evaluated for the actual operations it performs rather than its specific behaviors.

|  |  |
| --- | --- |
| **#** | **Requirement** |
| PR 1 | The app should load quickly |
| PR 2 | The app should always work reliably |
| PR 3 | The app should respond comfortably to user input |
| PR 4 | The app should be easy to maintain |
| PR 5 | The app should support data integrity |

* **App should load quickly**. It should be fast and reliable to create a more pleasant user experience overall
* **App should always work reliably**. This is a critical requirement, as the inventory and request systems should always be open and functional for safety and efficiency purposes
* **App should respond comfortably to user input**. The application should be intuitive to students and faculty with all levels of technological experience
* **App should be easy to maintain**. ITS plans to support this system in future semesters after the academic year is over
* **App should support data integrity**. The data we receive from already existing sources like A.C.I.D. and BioRAFT should be handled and processed in an accurate fashion

In addition, keeping these requirements in mind, we intend for our application to excel in five major ways as extensions of the original five non-functional requirements.

|  |  |
| --- | --- |
| **#** | **Requirement** |
| PR 6 | Speed |
| PR 7 | Usability |
| PR 8 | Extensibility |
| PR 9 | Scalability |
| PR 10 | Accessibility |

* **Speed**: The lab tool needs to work quickly and rapidly respond to input given by the user. Accessing chemical data, request data, and so forth in a timely manner, especially in lab, is a crucial aspect to making sure our application meets client requirements satisfactorily. We will utilize appropriate tools during development and testing to measure and evaluate the application’s bandwidth usage and speed, with a focus on the server’s capabilities.
* **Usability**: The system has to have a wide range of features while also being able to accommodate the needs of users with any experience level of technology use. This is largely how we will ensure our domain requirements are implemented correctly.
* **Extensibility**: The system is required to be built in mind with future ITS support. We need to make sure that we are building a system that can be extended and developed further upon completion of our capstone sequence. This will be expanded on in our Environmental Requirements.
* **Scalability**: Our application has to account for the fact that the system could grow far beyond both our expectations and what we anticipate. The inventory system, for example, could grow increasingly large and become more complex, and we need to account for this at the base level when building our first prototypes.
* **Accessibility**: In a similar manner to usability as mentioned above, we simply need to ensure that the application can be accessed by all required parties (students, faculty, administrators, ITS, and so forth) in a manner that is intuitive, straightforward, and clear from all perspectives and roles.

**3. Environmental Requirements**

Environmental requirements define the context the system exists in. At NAU, there is an extensive software ecosystem in which this system must operate, and we must take that, which is a requirement in and of itself, into account when creating our system. Following our completion of our capstone and our system is deployed, it will be managed and supported by NAU’s ITS department. NAU ITS operates and supports software that consists largely of Microsoft technology, and is moving farther away from other technologies in order to foster a united software ecosystem. Our system must be easily integrated *into* this ecosystem in order to be functional at NAU and facilitate any further development which may occur after our project has completed, and will be hosted by ITS on their servers.

This knowledge provides us with several environmental requirements that will need to be factored into the development of the system.

|  |  |
| --- | --- |
| **#** | **Requirement** |
| ER 1 | The system must be compatible with the existing NAU software ecosystem |
| ER 2 | The system must be supportable by NAU ITS |
| ER 3 | The system must have cross-browser compatibility |
| ER 4 | The system must be able to meet our other requirements on multiple platforms |
| ER 5 | The system must follow and enable the Standard Operating Procedure for the ENE lab |

Compatibility with the NAU software ecosystem introduces some constraints on our development, such as the necessity we implement this system using tools and languages approved by ITS. We hope to ensure this by working closely with ITS throughout development, so that we can catch and mitigate problems before they interfere with system functionality and effectiveness.

The system needs to be compatible across browsers, so that it doesn’t matter if a user is using Safari, Chrome, or another browser to access the system, or if the system is being used on a desktop, tablet, or mobile phone. We will need to ensure graceful degradation of the system across platforms, meaning it will adapt to smaller sized screens or less capable browsers. Part of this will be designing the system’s front-end so that it adapts to different views, and another part will be ensuring the back-end can handle the difference in platforms.

This system must also follow and enable the Standard Operating Procedure for the ENE lab, which dictate the lab usage request process, as well as the policies and procedures of lab operation. This is an environmental, as well as domain-level, requirement for the system.

Should we fail to keep these requirements in mind, we risk creating a system that is unsupportable and unusable, and we fail to solve our client’s problem. We will go further into this as a risk in the next section.

**V. Potential Risks**

There is inherently always a set of risks with large scale software projects such as these, and we have taken time to evaluate the issues that may become the most apparent as we work through our solution.

During the synthesis of our solution, we evaluated several risks associated with finding the ideal solution. We considered three different possible solutions: A mobile application, desktop software, and a web-based application. Each of these came with their own individual benefits, but also risks we needed to consider before settling on out solution.

|  |  |
| --- | --- |
| **Mobile app** | **Pros:** Portable, fast, continuous availability, easy for users to make requests  **Cons:** Not ideal for managing the different aspects of the lab, too many features could make the app too big, cross-platform compatibility |
| **Desktop software** | **Pros:** Makes it easy to have a lot of features, easy to have different modes for normal users and lab administration  **Cons:** Difficulties with cross-platform compatibility, difficult maintainability requiring software updates, would have to be installed on every device, limited availability only on desktops and laptops as opposed to mobile devices |
| **Web app** | **Pros:** Portable, platform independent, available on any device with a browser, easy to facilitate modes for normal users and lab administration, accessibility, zero installation for users, easily maintainable  **Cons:** Cross-browser compatibility, requires an Internet connection |

Ultimately, we chose to implement our solution as a web-based application. This carries its own set of risks, including web security and an inability to correctly implement cross-browser compatibility. We believe we can mitigate these risks if we take appropriate measures during development. There are still some risks that we face when we go into development.

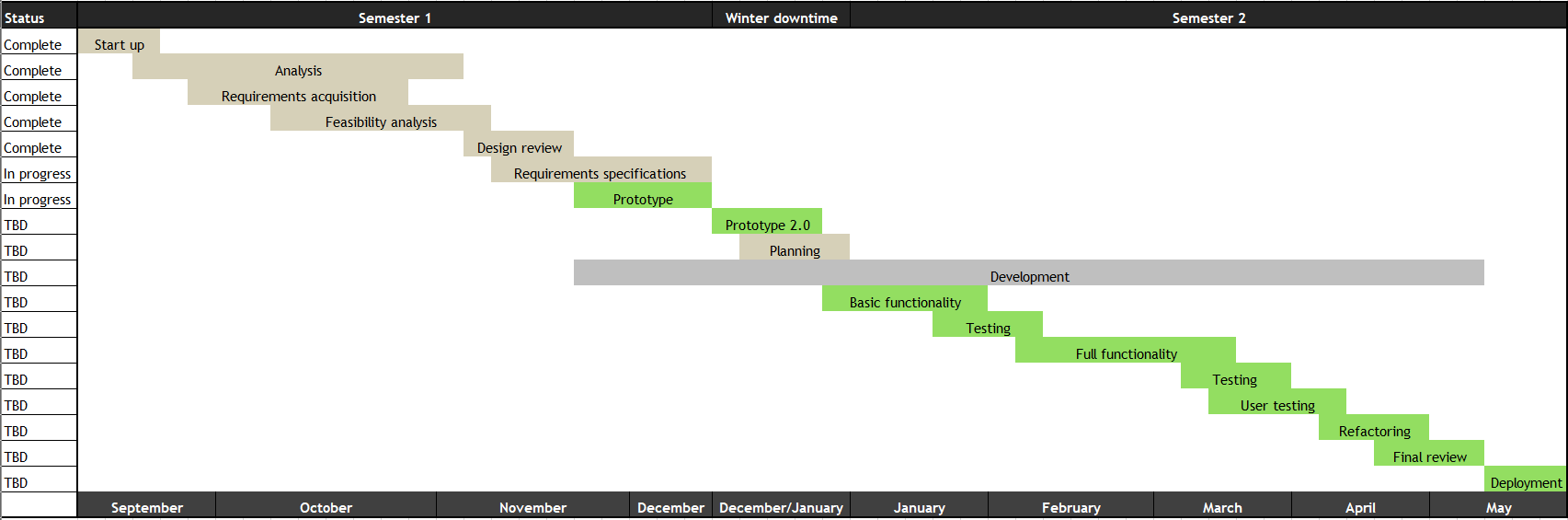
* Integration issues
  + A.C.I.D., BioRaft, ITS tools
* Registration problems
* Request form problems
* Non-working inventory
* Issues with learning ITS-sponsored tools we are required to use for the project
* Hazards and safety risks that could arise from any incorrect configurations above

Not all of these issues are as critical as others. We will ultimately determine through trial and error and working through the project which issues are the most apparent to us, but right now, it seems that the biggest points of contention will be the aforementioned integration issues and avoiding problems we may run into regarding the ITS toolset. There are potential problems involved with trying to integrate our system into the existing NAU software ecosystem, which we will address as they come up.

If we fail to correctly integrate these systems, or our system cannot fit into the NAU software ecosystem, then our system may not be able to be deployed and we will fail to create a system that appropriately addresses our client’s problems. Furthermore, if the system doesn’t work with A.C.I.D. or BioRAFT, then we will also fail to fully meet our requirements. This failure to meet our requirements is also an issue if we should have problems registering users, or if the request form does not correctly work.

**VI. Project Plan**

The project and software development will progress along the timeline set by NAU’s CS capstone program across two semesters, as well as a timeline we have created. As we feel it is unrealistic to set specific dates, our project plan is based on a series of project milestones with rough timelines to give us some flexibility, which we outline here in a Gantt chart.



*Planned progression of the project*

We explain each of these milestones below, detailing what they mean for the project and its progression.

* **Project start-up**: Project assignments and team website development
* **Analysis**: We began meeting with our clients and determining what the client’s needs are
  + **Requirements acquisition:** Determining what problem the project will solve, how it will solve those problems, and what kind of technologies we may employ to do so
* **Feasibility analysis**: Analysis of how we can solve these problems and create this software realistically using technologies, what challenges we may face, and how we’ll solve them, and ultimately, what technologies we will use to create this system
* **Design review:** A review of the project and how we intend to create it
* **Requirements specifications:** A determination of what requirements this project will have, and how we intend to address them, culminating in this document which will serve as our guide going forward in this project
* **Prototype:** The initial prototype for a tech demo, which serves as a proof of concept, and demonstrates the practical feasibility of the software. We intend the prototype to be the request system, as a demonstration that this project is feasible, that we will continue to build on throughout the project
* **Prototype 2.0:** A redo of the prototype, ironing out any issues, so that we may progress with further development (may possibly be combined with the original prototype if we find it unnecessary)
* **Planning:** We will complete planning for the full development and implementation of the software, so we have a blueprint to use for the semester
* **Basic functionality:** An implementation of the framework of the system and the bare essentials so the system exists at its lowest level, which we can flesh out in the next stage for a complete
* **Full functionality:** We fully flesh out the system so that it fulfills all requirements and is ready for final user testing and eventual deployment
* **User testing:** The system will be evaluated both by the client and by regular users, such as students, who will be using it
* **Refactoring:** We will make any changes necessary before deployment (this will overlap with the client’s final review)
* **Final review by client:** The client will do their final evaluation of the system in conjunction with our final refactoring
* **Deployment of software:** The system will go live at the end of the semester, at the client’s discretion

We are pursuing a test-driven development process, and as such, testing will occur throughout the development process, so it doesn’t have its own milestones in this project plan.

We will perform testing with different types of users, including the client and lab management, as well as targeted user testing with students in the Environmental Engineering program who would use this system in practical application to request usage of the ENE lab for their own research projects and classes; it’s important that the users of our system evaluate it before its deployment, so we don’t release a product that’s actually difficult, or even impossible, for real use.

**VII. Conclusion**

Our project will address the current difficulties and inefficiencies of the Environmental Engineering (ENE) lab management and usage processes, as explained to us by our client, Dr. Terry Baxter. The ENE lab supports research into, among other topics, climate change, water safety, and sustainability, which contribute to global efforts of preserving the environment. When there are failures or issues with research or data, which can occur in any number of ways, that endangers the validity of the results, and as a whole, endangers the field of environmental engineering. Consequently, this may affect life on Earth as a whole. The ENE lab critically needs a system that centralizes the management and lab usage processes in order to operate efficiently.

We propose a robust web-based application that will combine lab usage requests and lab management into one system, allowing users to make requests for lab usage and check the lab’s inventory, as well as facilitating lab management’s maintenance of the space, all in one convenient place. This app will be portable, platform independent, and easy to maintain, and will ensure the safe and efficient operation of the ENE lab.

In our efforts to solve this problem and make our solution a reality, we have completed our analysis of the requirements and feasibility of different technologies, and determined what requirements we must fulfill to fully implement this system. We are currently developing a prototype of the request form aspect of the system for our tech demo as a proof of concept, and we are confident that we will be able to fully implement a web-based system that addresses and solves our client’s problems by the end of this capstone project.

**VIII. Glossary**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **A.C.I.D.** | The online *a*utomated *c*hemical *i*nventory *d*atabase is the chemical inventory database system used in the Environmental Engineering laboratories for chemical information, health and safety information, and inventory management |
| **Accessibility** | The design of a system to be used by anyone, including persons with disabilities, such as the ability to use a screen reader or speech-to-text software with a system |
| **Back end** | The portion of the system that runs and operates the front end, and is not directly accessed by the user, such as any programs and databases; the functioning parts of the system that do all the work |
| **BioRAFT** | The third party software used by NAU’s Environmental Health and Safety department to track the necessary health and safety trainings required and completed by personnel for the labs at NAU |
| **Client side** | Operations that are performed by the client in a client-server relationship; operations performed by a computer application such as a web browser giving the user access to the web app |
| **DBMS** | The *d*ata*b*ase *m*anagement *s*ystem is the software used to manage and interact with a database |
| **Domain requirement** | How the system will satisfy the user’s needs at the user level; also called ‘user requirements’ |
| **ENE** | Environmental Engineering, used in reference to the academic program and the laboratory operated by the program |
| **Environmental requirement** | A requirement for the context the system exists and operates in |
| **Extensibility** | A measure of the ability to extend or grow the system in the future, such as adding new functionality or modifying current functionality. Extensibility often overlaps with scalability |
| **Front end** | The portion of a system directly accessed by the user, such as the website or a menu; the layout, visuals, and human-computer interaction of a system |
| **Functional requirement** | A function the system must be able to perform; what the system should do |
| **Non-functional requirement** | A behavior the system must be able to perform; how the system performs; also called ‘performance requirements’ |
| **Performance requirement** | Please see ‘Non-Functional Requirement’ |
| **Prototype** | An early model of a product built to test a concept or process, or as proof that a system is possible |
| **Scalability** | A system’s capability to handle a growing amount of work. Scalability often overlaps with extensibility |
| **Server side** | Operations performed by the server in a client-server relationship; operations performed by the web server running the web app and database |
| **Software ecosystem** | A collection of software projects which are created and operate in the same environment |
| **SOP** | The rules and guidelines for the operation of the Environmental Engineering laboratories, which detail the policies and processes for operating the lab, as well as the process for requesting lab usage |
| **Usability** | Describes how easy a system is to use and how easy or intuitive it is to learn; the extent to which a system can be used by specified users to achieve specific goals |
| **User requirement** | Please see ‘Domain Requirement’ |
| **W3C** | The World Wide Web Consortium; the main international standards organization for the World Wide Web |
| **WAI-ARIA** | ‘Web Accessibility Initiative -- Accessible Rich Internet Applications’; a technical specification published by the W3C that specifies how to increase the accessibility of web pages; consists of a suite of several documents detailing specifications, problems, and best practices for developing highly accessible web applications |
| **WCAG** | The ‘Web Content Accessibility Guidelines’ are a W3C checklist to evaluate a website or application’s accessibility to disabled users |
| **Web application** | A program that runs in and can be used through a web browser |

**IX. Appendices**

1. Standards Operating Procedure
2. Lab Usage Request Form