<Company Name>

<Project Name>

UNF SoC Honors in Computing Hours Tracking
CAPA Tech

Vision

Version <1.0>

[Note: The following template is provided for use with the Rational Unified Process. Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]

[To customize automatic fields in Microsoft Word (which display a gray background when selected), select File>Properties and replace the Title, Subject and Company fields with the appropriate information for this document. After closing the dialog, automatic fields may be updated throughout the document by selecting Edit>Select All (or Ctrl-A) and pressing F9, or simply click on the field and press F9. This must be done separately for Headers and Footers. Alt-F9 will toggle between displaying the field names and the field contents. See Word help for more information on working with fields.]

Revision History

Date	Version	Description	Author
<14/009/22>	<1.0>	Sections 1.1, 1.2, 1.3, 1.4, 1.5, 3.1, 3.4, 3.8	Chris Clark
14/009/22	1.0	Sections 4.1, 4.2, 4.3, 4.4, 4.5	Alexander Perez
14/009/22	1.0	Sections 2.1, 2.2, 2.3	Sophia Abuzeni
14/009/22	1.0	Sections 3.2, 3.3, 3.5, 3.6, 3.7	Chance Abenes

Table of Contents

Vision

1. Introduction

[The purpose of this document is to collect, analyze, and define high-level needs and features of the <<System Name>>*. It focuses on the capabilities needed by the stakeholders and the target users, and* ***why*** *these needs exist. The details of how the* <<System Name>> *fulfills these needs are detailed in the use-case and supplementary specifications.]*

[The introduction of the ***Vision*** *document provides an overview of the entire document. It includes the purpose, scope, definitions, acronyms, abbreviations, references, and overview of this* ***Vision*** *document.]*

This document's goal is to collect, analyze, and define high-level requirements and features for the UNF SoC Honors in Computing Hours Tracking system. (1) The system will automate the process of honors students reporting their activities in order to earn leadership hours. (2) Permit faculty to act on student-submitted activity reports, and (3) allow school administration to review student records and generate end-of-semester reports.

1.1 Purpose

The purpose of this document is to define the high-level feature requirements of the tracking system regarding the students who will use the tracking system each semester.

1.2 Scope

[A brief description of the scope of this ***Vision*** *document; what Project(s) it is associated with and anything else that is affected or influenced by this document.]*

This Vision Document applies to the UNF SoC Honors in Computing Hours Tracking system. This system will be developed by CAPA Tech. This system will: (1) automate the process of honors students reporting their activities in order to earn leadership hours; (2) Allow faculty to act on activity reports submitted by students. (3) Permit school administration to review student records and generate end-of-semester reports.

1.3 Definitions, Acronyms, and Abbreviations

[This subsection provides the definitions of all terms, acronyms, and abbreviations required to properly interpret the ***Vision*** *document. This information may be provided by reference to the project's Glossary.]*

Currently None. (Will change/update if necessary)

1.4 References

None.

1.5 Overview

[This subsection describes what the rest of the ***Vision*** *document contains and explains how the document is organized.]*

The remainder of this Vision document will define the problem we are attempting to solve. The user experience, Stakeholder summary/Market Demographic, and product features.

2. Positioning

2.1 Business Opportunity

This software application will keep track of leadership hours for honor students.
[Briefly describe the business opportunity being met by this project.]

2.2 Problem Statement

[Provide a statement summarizing the problem being solved by this project. The following format may be used:]

The problem of	*UNF school of Computing does not have a way to keep track of all leadership hours completed by students.*
affects	*Students, faculty, and administrators.*
the impact of which is	*Lack of a leadership hour logs.*
a successful solution would be	*Having a form where students can insert the number of hours they have achieved.*

2.3 Product Position Statement

[Provide an overall statement summarizing, at the highest level, the unique position the product intends to fill in the marketplace. The following format may be used:]

For	*Students, faculty, and administrators*
Who	*The students who log in their hours*
The (product name)	*is a software application to keep track of leadership hours completed by honor students.*
That	*This software app could be used for other departments or other schools. [statement of key benefit; that is, the compelling reason to buy]*
Unlike	*Onbase*
Our product	*Our product differs from onbase due to the fact that*

[A product position statement communicates the intent of the application and the importance of the project to all concerned personnel.]

3. Stakeholder and User Descriptions

[To effectively provide products and services that meet your stakeholders' and users' real needs, it is necessary to identify and involve all of the stakeholders as part of the Requirements Modeling process. You must also identify the users of the system and ensure that the stakeholder community adequately represents them. This section provides a profile of the stakeholders and users involved in the project, and the key problems that they perceive to be addressed by the proposed solution. It does not describe their specific requests or requirements as these are captured in a separate stakeholder requests artifact. Instead, it provides the background and justification for why the requirements are needed.]

3.1 Market Demographics

[Summarize the key market demographics that motivate your product decisions. Describe and position target market segments. Estimate the market's size and growth by using the number of potential users or the amount of money your customers spend trying to meet needs that your product or enhancement would fulfill. Review major industry trends and technologies. Answer these strategic questions:

- *What is your organization's reputation in these markets?*
- *What would you like it to be?*
- *How does this product or service support your goals?]*

The Tracking System's intended audience is UNF honors students, who are required to submit their leadership activity hours.

3.2 Stakeholder Summary

[There are a number of stakeholders with an interest in the development and not all of them are end users. Present a summary list of these non-user stakeholders. (The users are summarized in section 3.3.)]

Name	Description	Responsibilities
CAPA TECH	*Student Developers*	• *Developing a system to manage a leadership hour program.* • *Ensures the system will meet all end-users' goals.*
Johnathan Ohlrich	*Team Mentor*	• *Provide coaching and support for development teams.*

3.3 User Summary

[Present a summary list of all identified users.]

Name	Description	Responsibilities	Stakeholder
Faculty	*Administrators*	• *Host/Sponsor leadership activities* • *Approve/deny student leadership form requests* • *Send leadership forms back to students for correction* • *Send leadership forms to administrators for final approval*	*Administrator*
Administrators	*True Administrators*	• *Approve leadership forms sent by faculty for documentation* • *Generating reports for hours accrued at the end of semester* • *Sole System Administrator oversees account creation approval*	*Administrator*

Student	*User*	- *Utilize the system to apply for their leadership hours to be documented.*	*Students*

3.4 User Environment – chris doing

[Detail the working environment of the target user. Here are some suggestions:

The working environment is four senior college students called CAPA Tech working on a tracking system for UNF honor students. Each task cycle lasts between 2 weeks to one month. Task cycles are subject to change and can change in the future if needed. Mobile responsive (feature) and desktop ready application for UNF SoC Honors in Computing, using express.js. This Tracking system is a new platform and does have plans to expand to other honor programs in the future if necessary. As far as the application does not need anything to integrate with, this is a totally self-contained application.

3.5 Stakeholder Profiles

[Describe each stakeholder in the system here by filling in the following table for each stakeholder. Remember that stakeholder types can be as divergent as users, departments, and technical developers. A thorough profile would cover the following topics for each type of stakeholder.]

3.5.1 CAPA TECH

Representative	*Alexander Perez, Sophia Abuzeni, Chance Abenes, Christopher Clark, Johnathan Ohlrich*
Description	*[A brief description of the stakeholder type.]*
Type	*Experts*
Responsibilities	*Develop leadership hour management system*
Success Criteria	*A functional leadership hour workflow system that allows for student record documentation*
Involvement	*Address Dr.Sherif Elfayoumy's needs for the leadership hour program*
Deliverables	*3 Deliverables including documentation: project vision, product backlog, user stories, product releases, activity diagram, analysis diagram, sequence diagram, entity-relationship diagram, and user interface mock-ups.*
Comments / Issues	*Follow and adhere to Dr.Sherif Elfayoumy's concerns regarding the system and implement all requirements that would most benefit our client.*

3.5.2 UNF School of Computing

Representative	*Dr. Sherif Elfayoumy, Administrators, Faculty, and Students who will use this system.*

Description	*The UNF School of Computing*
Type	*Typically, avid computer users*
Responsibilities	*Provide CAPA TECH development team with all requirements. Give consent and documents needed to authorize the development team's work on the Leadership Program.*
Success Criteria	*Success will be graded upon the efficiency and functionality of the develop team's work system. Success will be achieved given that this program is in excellent order and meets all requirements the system will be implemented to use within the UNF School of Computing.*
Involvement	*Provides requirements and feedback for each iteration of the system.*
Deliverables	*The system will need updates in the future following the development cycle.*
Comments / Issues	*N/A*

3.6 User Profiles

[Describe each unique user of the system here by filling in the following table for each user type. Remember user types can be as divergent as gurus and novices. For example, a guru might need a sophisticated, flexible tool with cross-platform support, while a novice might need a tool that is easy to use and user-friendly. A thorough profile needs to cover the following topics for each type of user.]

3.6.1 <Administration>

Representative	*Administrators*
Description	*Assigned administrators of the School of Computing*
Type	*Typically, avid computer users*
Responsibilities	*Manage student record documentation, approve final requests from faculty, manage account creation*
Success Criteria	*Success is graded upon the ease of use of the system.*
Involvement	*Reviews requirements and features provided by development team.*
Deliverables	*Any future updates made to the system post-final product release. Any documentation that is to be recorded by the administrative team.*
Comments / Issues	*Take note of who will be directly managing this system (I.e., System Admin) in the future and who will be responsible for updates being made to the system.*

3.6.2 <Faculty>

Representative	*Administrators*
Description	*Responsible for creating events*

Type	*Typically, avid computer users*
Responsibilities	*Host events that would allow students to achieve leadership hours, approve requests from students*
Success Criteria	Success is graded upon the ease of use of the system by faculty
Involvement	*Manage leadership hour requests from students.*
Deliverables	*Reevaluating student reports*
Comments / Issues	*N/A*

3.6.3 <Students>

Representative	*School of Computing Students*
Description	*Students who wish to submit a request for leadership hours*
Type	*Primary user of the system*
Responsibilities	*Submit requests pertaining to leadership hour accrual events*
Success Criteria	*Students can submit requests for documentation of their leadership hours in a simple manner.*
Involvement	*Primary user of system*
Deliverables	*Students will create an account to register with the system providing any relevant information*
Comments / Issues	*N/A*

3.7 Key Stakeholder or User Needs

Need	Priority	Concerns	Current Solution	Proposed Solutions
Securely store student record information	HIGH	None	N/A	SQL/NOSQL Database that will store student account information, leadership hour data, n-numbers. This will need to be secure.
Ability to track student reports	HIGH	None	N/A	Workflow history will be implemented, or a search criteria function to find specific instances of students or events.
Be able to print end of semester reports	HIGH	Issues with formatting	N/A	.xlsx documents format for reports

Abide by UNF Accessibility Requirements	HIGH	None	N/A	Follow Accessibility Requirements
Able to register for events and gain approval for leadership hours	HIGH	None	N/A	Ideally faculty create events. Students may find the events to register for and later request approval of leadership hours. This is then sent to faculty who will review and approve or send the report back to the student. Assuming approval of the report is sent to administration where it will undergo a final approval before being documented for the program. Data regarding hours collected will be available to view on request of students and faculty alike.

3.8 Alternatives and Competition – doing

[Identify alternatives the stakeholder perceives as available. These can include buying a competitor's product, building a homegrown solution or simply maintaining the status quo. List any known competitive choices that exist or may become available. Include the major strengths and weaknesses of each competitor as perceived by the stakeholder or end user.]

3.8.1 <Onbase>

A system that will be used in place if the tracking system that is being created by CAPA Tech does not work properly. (Only known competitor with similar workflow in place).

4. Product Overview

[This section provides a high level view of the product capabilities, interfaces to other applications, and system configurations. This section usually consists of three subsections, as follows:

- *Product perspective*
- *Product functions*
- *Assumptions and dependencies]*

4.1 Product Perspective

[This subsection of the ***Vision*** *document puts the product in perspective to other related products and the user's environment. If the product is independent and totally self-contained, state it here. If the product is a component of a larger system, then this subsection needs to relate how these systems interact and needs to identify the relevant interfaces between the systems. One easy way to display the major components of the larger system, interconnections, and external interfaces is with a block diagram.]*

The product will need to be independent of existing UNF systems, and completely self-contained. While the product is a component of the student, faculty, and administrator experience, it will not utilize existing UNF architecture or cloud infrastructure. Anything the product is capable of will need to be created from scratch.

4.2 Summary of Capabilities

*[Summarize the major benefits and features the product will provide. For example, a **Vision** document for a customer support system may use this part to address problem documentation, routing, and status reporting without mentioning the amount of detail each of these functions requires.*

Organize the functions so the list is understandable to the customer or to anyone else reading the document for the first time. A simple table listing the key benefits and their supporting features might suffice. For example:]

Table 4-1 Customer Support System

Customer Benefit	Supporting Features
Leadership hour submission will be easy and straightforward.	Faculty will be linked to their students, and can either create events for students to submit hours on, or students can create new events which they provide hours on.
Leadership hour approval will be swift and clean.	If opted in to email notifications, users will be notified when they have actionable items.
Administration can pull reports on anyone at any time.	All student accounts and hours accrued will be available for administrators to review. Filters will exist to only pull data for certain individuals or between certain dates.
Users can remotely submit or approve hours at any time from any device.	The product will be compatible with desktop and mobile devices, and hour submission is not time-gated. It may be done at any time.
The software will be entirely self-service, eliminating the need for support staff.	Product documentation will be made available on the website, with tooltips linking related articles where appropriate. Additionally, account creation and management will be entirely self-service.

4.3 Assumptions and Dependencies

*[List each of the factors that affect the features stated in the **Vision** document. List assumptions that, if changed, will alter the **Vision** document. For example, an assumption may state that a specific operating system will be available for the hardware designated for the software product. If the operating system is not available, the **Vision** document will need to change.]*

While we understand fundamentally what data needs to be tracked, there is always the possibility that more data will be necessary as the project progresses. While on the topic of data, student data is very sensitive, and is subject to FERPA. Working with university registrars may prove to be exceptionally challenging, as identifiable data will need to be secured in accord with university regulations. If we are unable to secure the data to UNF standards, we will be unable to use this data.

In regards to where the program will be hosted, we have two options. We can either host it on-premises or in the cloud. Both have their advantages and disadvantages that need to be weighed very carefully. The vision could change significantly depending on where we implement the product.

The workflow between faculty and students isn't crystal clear. We have been given a broad outline of what is important at the end of the day, but how faculty and students interact within the system is not clear.

Finally, while account creation will be self-service, there will need to be a system administrator role. There needs to be a fallback for issues relating to the system administrator account.

4.4 Cost and Pricing

[For products sold to external customers and for many in-house applications, cost and pricing issues can directly impact the application's definition and implementation. In this section, record any cost and pricing constraints that are relevant. For example, distribution costs, (# of diskettes, # of CD-ROMs, CD mastering) or other cost of goods sold constraints (manuals, packaging) may be material to the projects success, or irrelevant, depending on the nature of the application.]

The only cost that should be considered for this project is the cost of hosting the product. If we choose a cloud provider, we will specifically need to select a free one. If we use an on-premises solution, electricity cost and support staff costs will need to be considered. (Support staff is not really a consideration, as the product would be hosted on a server managed by ITS, a service provided to the college for free)

4.5 Licensing and Installation

[Licensing and installation issues can also directly impact the development effort. For example, the need to support serializing, password security or network licensing will create additional requirements of the system that must be considered in the development effort.

Installation requirements may also affect coding or create the need for separate installation software.]

As we intend to use open-source software for the entirety of this product, there are no licensing limitations at this time.

5. Product Features

[List and briefly describe the product features. Features are the high-level capabilities of the system that are necessary to deliver benefits to the users. Each feature is an externally desired service that typically requires a series of inputs to achieve the desired result. For example, a feature of a problem tracking system might be the ability to provide trending reports. As the use-case model takes shape, update the description to refer to the use cases.

*Because the **Vision** document is reviewed by a wide variety of involved personnel, the level of detail needs to be general enough for everyone to understand. However, enough detail must be available to provide the team with the information they need to create a use-case model.*

To effectively manage application complexity, we recommend for any new system, or an increment to an existing system, capabilities are abstracted to a high enough level so 25-99 features result. These features provide the fundamental basis for product definition, scope management, and project management. Each feature will be expanded in greater detail in the use-case model.

Throughout this section, each feature will be externally perceivable by users, operators or other external systems. These features need to include a description of functionality and any relevant usability issues that must be addressed. The following guidelines apply:

- *Avoid design. Keep feature descriptions at a general level. Focus on capabilities needed and why (not how) they should be implemented.*
- *If you are using the Rational RequisitePro toolkit, all need to be selected as requirements of type for easy reference and tracking.]*

5.1 <aFeature>

5.2 <anotherFeature>

6. Constraints

[Note any design constraints, external constraints or other dependencies.]

7. Quality Ranges

[Define the quality ranges for performance, robustness, fault tolerance, usability, and similar characteristics that are not captured in the Feature Set.]

8. Precedence and Priority

[Define the priority of the different system features.]

9. Other Product Requirements

[At a high level, list applicable standards, hardware or platform requirements, performance requirements, and environmental requirements.]

9.1 Applicable Standards

[List all standards with which the product must comply. These can include legal and regulatory (FDA, UCC) communications standards (TCP/IP, ISDN), platform compliance standards (Windows, UNIX, and so on), and quality and safety standards (UL, ISO, CMM).]

9.2 System Requirements

[Define any system requirements necessary to support the application. These can include the supported host operating systems and network platforms, configurations, memory, peripherals, and companion software.]

9.3 Performance Requirements

[Use this section to detail performance requirements. Performance issues can include such items as user load factors, bandwidth or communication capacity, throughput, accuracy, and reliability or response times under a variety of loading conditions.]

9.4 Environmental Requirements

[Detail environmental requirements as needed. For hardware- based systems, environmental issues can include temperature, shock, humidity, radiation, and so forth. For software applications, environmental factors can include usage conditions, user environment, resource availability, maintenance issues, and error handling and recovery.]

10. Documentation Requirements

[This section describes the documentation that must be developed to support successful application deployment.]

10.1 User Manual

[Describe the purpose and contents of the User Manual. Discuss desired length, level of detail, need for index, glossary of terms, tutorial versus reference manual strategy, and so on. Formatting and printing constraints must also be identified.]

10.2 Online Help

[Many applications provide an online help system to assist the user. The nature of these systems is unique to application development as they combine aspects of programming (hyperlinks, and so forth) with aspects of technical writing, such as organization and presentation. Many have found the development of an online help system is a project within a project that benefits from up-front scope management and planning activity.]

10.3 Installation Guides, Configuration, and Read Me File

[A document that includes installation instructions and configuration guidelines is important to a full solution offering. Also, a Read Me file is typically included as a standard component. The Read Me file can include a "What's New With This Release" section, and a discussion of compatibility issues with earlier releases. Most users also appreciate documentation defining any known bugs and workarounds in the Read Me file.]

10.4 Labeling and Packaging

[Today's state-of-the-art applications provide a consistent look and feel that begins with product packaging and manifests through installation menus, splash screens, help systems, GUI dialogs, and so on. This section defines the needs and types of labeling to be incorporated into the code. Examples include copyright and patent notices, corporate logos, standardized icons and other graphic elements, and so forth.]

A Feature Attributes

[Features are given attributes that can be used to evaluate, track, prioritize, and manage the product items proposed for implementation. All requirement types and attributes need to be outlined in the Requirements Management Plan, however, you may wish to list and briefly describe the attributes for features that have been chosen. The following subsections represent a set of suggested feature attributes.]

A.1 Status

[Set after negotiation and review by the project management team. Tracks progress during definition of the project baseline.]

Proposed	*[Used to describe features that are under discussion but have not yet been reviewed and accepted by the "official channel," such as a working group consisting of representatives from the project team, product management, and user or customer community.]*
Approved	*[Capabilities that are deemed useful and feasible, and have been approved for implementation by the official channel.]*
Incorporated	*[Features incorporated into the product baseline at a specific point in time.]*

A.2 Benefit

[Set by Marketing, the product manager or the business analyst. All requirements are not created equal. Ranking requirements by their relative benefit to the end user opens a dialog with customers, analysts, and members of the development team. Used in managing scope and determining development priority.]

Critical	*[Essential features. Failure to implement means the system will not meet customer needs. All critical features must be implemented in the release or the schedule will slip.]*
Important	*[Features important to the effectiveness and efficiency of the system for most applications. The functionality cannot be easily provided in some other way. Lack of inclusion of an important feature may affect customer or user satisfaction, or even revenue, but release will not be delayed due to lack of any important feature.]*
Useful	*[Features that are useful in less typical applications will be used less frequently or for which reasonably efficient workarounds can be achieved. No significant revenue or customer satisfaction impact can be expected if such an item is not included in a release.]*

A.3 Effort

[Set by the development team. Because some features require more time and resources than others, estimating the number of team or person-weeks, lines of code required or function points, for example, is the best way to gauge complexity and set expectations of what can and cannot be accomplished in a given time frame. Used in managing scope and determining development priority.]

A.4 Risk

[Set by development team based on the probability the project will experience undesirable events, such as cost overruns, schedule delays or even cancellation. Most project managers find categorizing risks, as high, medium, and low, is sufficient, although finer gradations are possible. Risk can often be indirectly assessed by measuring the uncertainty (range) of the projects team's schedule estimate.]

A.5 Stability

[Set by the analyst and development team, this is based on the probability that features will change or the team's understanding of the feature will change. Used to help establish development priorities and determine those items for which additional elicitation is the appropriate next action.]

A.6 Target Release

[Records the intended product version in which the feature will first appear. This field can be used to allocate features from a ***Vision*** *document into a particular baseline release. When combined with the status field, your team can propose, record, and discuss various features of the release without committing them to development. Only features whose Status is set to Incorporated and whose Target Release is defined will be implemented. When scope management occurs, the Target Release Version Number can be increased so the item will remain in the* ***Vision*** *document but will be scheduled for a later release.]*

A.7 Assigned To

[In many projects, features will be assigned to "feature teams" responsible for further elicitation, writing the software requirements, and implementation. This simple pull-down list will help everyone on the project team to understand responsibilities better.]

A.8 Reason

[This text field is used to track the source of the requested feature. Requirements exist for specific reasons. This field records an explanation or a reference to an explanation. For example, the reference might be to a page and line number of a product requirement specification or to a minute marker on a video of an important customer review.]