UNA Advising System

Version 1.3

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 07/Mar/20 | 1.0 | Initial document creation | Bryon Miller |
| 10/Mar/20 | 1.1 | Resources section | Sai Namilakonda |
| 10/Mar/20 | 1.2 | Deployment Planning section | Andry Ndongosieme |
| 10/Mar/20 | 1.3 | Added Training section and made last second corrections to other sections for final submission | Bryon Miller |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms, and Abbreviations 4

1.4 Overview 4

2. References 5

3. Deployment Planning 5

3.1 Responsibilities 5

3.2 Schedule 5

4. Resources 5

4.1 Facilities 6

4.2 Hardware 6

4.3 The Deployment Unit 6

4.3.1 Support Software 7

4.3.2 Support Documentation 7

4.3.3 Support Personnel 8

5. Training 8

Deployment Plan

# Introduction

This document details the purpose, scope, and deployment plan for the UNA Advising System as developed by Team 5 for CIS 615.

## Purpose

The purpose of this software is to create a system for advising and registration that may be utilized by the students, advisors, registrars, and administrators at the University of North Alabama.

## Scope

The scope of this project covers four users including Advisee, Advisor, Registrar, and Administrator. The scope is also limited to the online MBA program at the University of North Alabama.

Advisee: This project will all Advisees to view their degree plan, view their schedule, view the institutional course schedule, add/drop classes, and view their Advisor’s contact information. Other Advisee tasks, such as paying for courses, will be outside the scope of the project.

Advisor: Advisors will be able to view Advisee’s degree plans and schedules, add/drop classes for Advisees, view the institutional course schedule, and request for courses to be added to the schedule or that prerequisites be overridden. Other Advisor tasks, such as changing an Advisee’s major, will be outside the scope of this project.

Registrar: Registrars will be able to view and edit degree plan templates, input Advisor and course/instructor information (including the institutional course schedule for the semester), and manage Advisee records and information. Other Registrar tasks, such as updating an Advisee’s transcript each semester, will be outside the scope of this project.

Administrator: Administrators will be superusers that have access and authority to do any of the tasks that the other users can do. Additionally, they will be able to run various queries and generate reports from the system. Other administrative tasks will be outside the scope of this project.

## Definitions, Acronyms, and Abbreviations

Advisee: A student that is assigned to and Advisor. May add/drop classes to their own Schedule and view this Schedule at any time.

Advisor: A UNA employee that is responsible for advising Advisees and registering them for classes.

Institutional Course Schedule: A list of courses available during the given semester.

Registrar: A UNA employee that is responsible for updated degree plans, transcripts, and Institutional Course Schedules each semester.

Schedule (for Advisee): A list of courses that been added to an Advisee’s schedule and have not been dropped. This list is empty initially and may remain empty or become empty again.

Student: A person that has been, is, or will be enrolled in courses at UNA.

Team 5: Any specific member of our company, “UNA Team 5”, or to refer to the company collectively.

UNA: The University of North Alabama.

## Overview

This document begins with a list of all documents referenced elsewhere in the Deployment Plan. Then, it proceeds to detail the deployment plan including responsibilities of UNA and Team 5 members and schedule/timeline for the project. Next, it discussed the various resources required for the project ranging from facilities to hardware, and so on. In this section, the plan also references all the software, support, and documentation that will be delivered to UNA upon completion. Lastly, a discussion on the training that will be provided and a rough timeline for training is provided.

# References

Deployment Schedule: This document is located in Appendix A (PDF file) and details maps out the proposed schedule for project completion.

# Deployment Planning

In this section we will point out the deployment planning which includes the responsibilities of the development team and the customer.

## Responsibilities

* + 1. *Development team responsibilities*

The development team will have the following responsibilities in order to deploy the software:

* Make sure that the software has been well frequently tested
* Make sure that the software is secure and function properly
* Make sure that the code is readable, well documented and well presented
* Make sure that there is a manual of how-to use the system
* Make sure that everything respond
* Make sure that customer’s requirements have been fulfilled
* Make sure that the features of the system are clarified
* Make sure that the deployment is on schedule and as planned
  + 1. *Customer responsibilities*

The customer in his turn also has responsibilities such as:

* Focus on the training for a better understanding of the system
* Follow the instructions related to the security of the system
* Report issues to the development team
* Request enhancement, update or upgrade of the system
* Allocate the space and resources for the deployment
* Schedule training for the use of the software

## Schedule

The proposed deployment schedule for this project may be found in Appendix A which is a separate (PDF) file.

# Resources

The **resources** are required to carry out the project tasks. They can be people, equipment, facilities, funding, or anything else capable of definition (usually other than labor) required for the completion of a project activity. The lack of a resource will therefore be a constraint on the completion of the project activity. Resources may be storable or non-storable. Storable resources remain available unless depleted by usage and may be replenished by project tasks which produce them. Non-storable resources must be renewed for each time period, even if not used in previous time periods.

Resource scheduling, availability and optimization are considered key to successful project management

Allocation of limited resources is based on the priority given to each of the project activities. Their priority is calculated using the Critical path method and heuristic analysis. For a case with a constraint on the number of resources, the objective is to create the most efficient schedule possible - minimizing project duration and maximizing the use of the resources available, the below mentioned list has resources and their sources required to carry out the planned deployment activites.

* Traditional Project Management. - Toggl
* Software Requirements Management. - aNimble
* Software Development Kits and Software Quality. Java Runtime environments (JRE)
* Rich Interface Application – HTML5
* Unified Modeling Language. - Visual paradigm
* Database Development. - Postgres Database
* Java Software Development.-Node.js

## Facilities

Disciplined software implementation principles, planning, and resources for systems build-up provide effective testing to be conducted in a development facility for a software and system integration environment. Software released under configuration management control is described in a defined and documented Configuration Management Plan (CMP) to provide the necessary requirements for software implementation inside integration facilities.

In the early stages of software design and development this project, a Secured Development Facility (DF) with good network connectivity is established for software development activities. This facility is used for preparation of software prior to delivery, the facility also has an effective way to test traffic loads on software products.

## Hardware

1. **OS:** Windows 7 with SP1; Recommended*:* Windows 10
2. **CPU:** Intel or AMD processor with 64-bit support; Recommended*:* 2.8 GHz or faster processor
3. **GPU:** NVidia GeForce GTX 1050 or equivalent; Recommended*:* NVidia GeForce GTX 1660 or Quadro T1000
4. **Disk Storage:** 20 GB of free disk space
5. **Monitor Resolution:** 1280x800; Recommended*:* 1920x1080
6. **Internet:** Internet connection required for software activation
7. Internet Explorer 11 or later
8. Google Chrome 44 or later

## The Deployment Unit

The detailed statement of Deployment Unit work categorized into seven (7) major implementation tasks. A summary of each task is provided below.

1. Task 1 - Project Initiation and Planning: it is regarding the project kick-off and management.

2. Task 2 – System, Interface, and Data flow Design: Regarding the developing and detailing of the plans for designing the System to meet the needs of Student Advising system software. This includes the design of the interfaces and data flow.

3. Task 3 - System Development / Configuration: it is regarding the development and/or configuration of the System to meet Student Advising system software’s needs through execution of the designs created in Task 2. This includes the development of the interfaces and data flow.

4. Task 4 – System Testing: it is regarding the testing of the System developed/configured in Task 3 to ensure that it meets the needs of ASR.

5. Task 5 – Project Training: it is regarding the training of ASR staff in using the new System.

6. Task 6 – Deployment: it is regarding the deploying of the new System into production.

7. Task 7 – Implementation Closeout: : it is the expectations regarding the process of concluding implementation.

### Support Software

Software tools are used to accomplish and investigate the business processes, document the development process of the software and optimize all the processes, by using these tools in the software development process, the outcome of the projects will be more productive. Using the development tools, a developer can easily maintain the workflow of the project.

**GitHub** is a powerful collaboration tool and development platform for code review and code management. With this GitHub, the users can build applications and software, manage the projects, host the code, review the code

**Node.js** is an open source, cross-platform and JavaScript run-time environment that is built to design a variety of web applications and to create web servers and networking tools.

**Microsoft Azure** is a cloud computing service that is used for designing, deploying, testing and managing web applications or hybrid cloud applications through Microsoft’s global network of data centers.

**SoapUI** is an open-source web service testing application for service-oriented architectures and representational state transfers. Its functionality covers web service inspection, invoking, development, simulation and mocking, functional testing, load and compliance testing

**PostgreSQL**, also known as Postgres, is a free and open-source relational database management system emphasizing extensibility and technical standards compliance. It is designed to handle a range of workloads, from single machines to data warehouses or Web services with many concurrent users

### Support Documentation

The Functional Design Document, including at a minimum:

1. Rules definition’s

2. Details on the requirements supported by the Student advising System software.

3. Reporting capabilities and prebuilt reports

4. User profiles and security role permissions

5. System functionality traceable back to the Requirements Traceability Matrix

6. System overview diagrams illustrating which system components provide what functionality, linking back to the workflows, use cases, and functional requirements

7. A maintainable list of workflows mapped to business processes and use cases mapped to System requirements

8. User interface screens for the system

9. A comprehensive list of functional specifications to implement the functionality required

.

The Testing Strategy deliverable, which shall, at a minimum, include:

1. The test methodology to be employed for overall system testing

3. Identified strategies for each type/level of Project testing: a. Unit and integration testing b. System testing c. End-to-end testing d. User acceptance testing e. Performance and load testing f. System regression testing g. Security testing h. Test Scripts

Backup and recovery procedures will be included to ensure that the Student advising system software can continue to operate in the event of an unexpected destruction of hardware, software, or communications through System failure, disruption of connectivity or natural disasters (these procedures and operations may differ depending on the proposed system delivery model, but shall still be addressed) o Arrangements for backup hardware or processing sites; off-site data storage; schedule for creation of backup media; and detailed recovery procedures for all anticipated types of disasters o Document proposed escalation plans that specify the necessary points of contact and decision-making authority at ASR o Restoration sequencing of the System implemented as a result of this Project/program

### Support Personnel

The Plan shall include the scope of support services documented in the Student Advising System software,

1. Level 1 – This is the initial support level responsible for basic student issues. The goal of Level 1 support is to gather the Student’s information and to determine the Student’s issue by analyzing the symptoms and figuring out the underlying problem. Level 1 will typically handle straightforward and simple problems while using a knowledge management tool. The goal for Level 1 is to generally handle 80% of the user problems before finding it necessary to escalate the issue to a higher level

2. Level 2 – This is a more in-depth technical support level than Level 1, and the staff are more experienced and knowledgeable with the use of the system. Support is often provided for bug fixes, custom reports, etc., which require configuration and/or technical expertise. If new problems are encountered and resolved that have not previously been documented in a knowledge management tool, Level 2 support resources are often responsible to develop and post instructions in the knowledge management tool 3.

Level 3 – This level represents an escalation to Proposer personnel responsible for the support of the software (or hardware, if applicable). Level 3 support resolves complex issues related to configuration and/or technical issues with the software. As is the case with Level 2 support, new problems that are encountered/resolved and have not previously been documented in a knowledge management tool are the responsibility of Level 3 support resources to develop and post instructions in the knowledge management tool The Proposer will use a help desk issue management software suite to collect and track all issues submitted to the Proposer for production support.

# Training

Training for this project will be multifaceted. It will include at minimum: a) written training documentation that may be utilized for future projects, b) training videos for Advisees, c) group training sessions for Advisors and Registrars, and d) individual training sessions with Administrators. Additionally, this section offers a rough schedule (e) for how training will be structured and offered.

* 1. Written documentation will be divided into subsections based on the user group and will be released as each user group is trained.
     1. Administrators will begin training first so that they can assist other user groups as they progress with system deployment. Their documentation will consist of details on how to perform core functions such as add/drop classes, view schedules, edit various fields, etc. Additionally, they will receive documentation on other Administrator-specific tasks such as ad hoc queries, system reports, and managing user permissions.
     2. Advisors will begin training next along with Registrars (see below) and they will therefore receive documentation next. Their documentation will focus on the core functions discussed in the Administrator section, but only the ones related to their role as Advisors. This means that they will be limited to documentation on how to add/drop classes, view degree plans and schedules, view their Advisee’s, etc. It will be written from their perspective with examples from the Advisee perspective as well.
     3. Registrars will begin training alongside Advisors and will receive their documentation at the same time. Their documentation will also focus on the core functions discussed previously but will be limited to those related to their roles. This will include editing degree plans, how to add/remove classes from the institutional course schedule, etc. It will be written from the perspective of a Registrar.
     4. Advisees will be trained last since they will be the last to gain access to the system. Their documentation will be in the form of FAQs, written instructions on how to perform basic tasks in the system, and related information. It will be written at a high school reading level and will be in an easy to navigate format accessible online.
  2. Training Videos
     1. Advisees will also have access to training videos that will walk them through the most important system tasks as defined by UNA employees. They will be from the perspective of an Advisee.
     2. These videos will also be accessible by other user groups, but their primary usage will be to train and support Advisees.
     3. Group training sessions (see below) will also be recorded for future use by other users.
  3. Group Training Sessions
     1. These sessions will be conducted with groups of Advisors or Registrars. Some large groups will be offered on-site initially but will move to primarily web-based training sessions after the first 2-4 weeks of training.
     2. Sessions with both groups will go over the documentation provided, the basics of the system and its usage, and role-specific instructions. Lastly, time will be allotted to answer questions and provide individual assistance as possible.
  4. Individual Administrator Training
     1. Administrators training sessions will all be individual since this is a small group and their needs are wide-ranging. It is expected that there will be several individual training sessions (10-15) lasting around 1-2 hours each. These sessions may be facilitated on- or off-site. They will cover the basics of the systems, role-specific instructions, and other areas as requested.
     2. A small portion of this training time will also be allotted to teach the Administrator how to assist other users in the future. Administrators may also choose to attend other user group training sessions.
  5. Schedule
     1. Administrators will begin training four weeks prior to the “Go Live” date
        1. Basic tasks and system overview – 4 weeks before Go Live
        2. Role-specific tasks – 2-3 weeks before Go Live
        3. Requested training – 1-2 weeks before Go Live through 4 weeks after
     2. Advisors and Registrars will begin training next at two weeks prior to Go Live
        1. Basic tasks and system overview – 2 weeks before Go Live
        2. Role-specific tasks – 2 weeks before Go Live through 4 weeks after
     3. Advisees will receive their documentation and training videos one week prior to the Go Live date and the video training will be available by the Go Live date