5-2018

CSLC Tutoring Portal

Brian Hodges
bhodges@unomaha.edu

Group Members:

Chengcheng Zhang
James Hopkins
Brian Hodges
Douglas Stahlnecker

Project Manager/Sponsor:
Dr. Paul van Vliet

Computer Science Learning Center

University of Nebraska Omaha
Peter Kiewit Institute

Contents

Milestone 5

January 10 – February 11, 2017

Milestone Manager:

Brian Hodges

Milestone 5 Contents

Control Documents

Roles and Responsibilities Matrix

Department: The Peter Kiewit Institute

Product/Process: Database of Students

Document Owner: Alpha Group Beta

Project or Organization Role: Design and Implement Database

Version	Date	Author	Change Description
1.0	2/1/16	Brian Hodges	Initial Version

Title	Role	Responsibilities
Dr. Paul Van Vliet Project Manager	The Project Manager is responsible for judging the quality of the project documents. The Project Manager ensures that the project is delivered on time, within budget, and to the required quality standards.	• Manage and lead the project team. • Manage the coordination of the partners and the working groups. • Develop and maintain a detailed project plan.
Brian Hodges Milestone Manager	The Milestone Manager is responsible for coordinating team efforts and maintaining the running document including tables of contents.	• Change Log • Roles & Responsibility Matrix • Running Document • Table of Contents • Relational Database Design
James Hopkins Project Coordinator	The Project Coordinator is responsible for maintaining and tracking the milestones and requirements for the project.	• Tracking Gantt Chart • Assist with the Entity Relationship Diagram
Chengcheng Zhang Communications Manager	The Communications manager is responsible for maintaining records of meetings and communication.	• Meeting Minutes • Email Log • Assist with the Entity Relationship Diagram
Douglas Stahlnecker Data Manager	The Data Manager is in charge of creating and updating the data requirements for the project.	• Entity Relationship Diagram • Client Documents

Milestone Change Log

Date	Rev. No.	Change	Reference Section
1/28/2017	1.0	Initial release	Roles and Responsibilities Matrix (RRM)
1/28/2017	1.0	Initial release	Entity Relationship Diagram (ERD)
2/1/2017	1.0	Initial release	Client Documents (CD)
2/4/2017	1.0	Initial release	Relational Database Design (RDD)
2/4/2017	1.0	Initial release	Tracking Gantt Chart (TGC)
2/7/2017	2.0	Made Dr. van Vliet's recommended changes	ERD
2/7/2017	1.1	Made Dr. van Vliet's recommended changes	CD
2/7/2017	1.1	Made Dr. van Vliet's recommended changes	RRM
2/7/2017	1.1	Made Dr. van Vliet's recommended changes	TGC
2/7/2017	2.0	Made Dr. van Vliet's recommended changes	RRD
2/8/2017	1.2	Made Dr. van Vliet's recommended changes	TGC
2/8/2017	1.3	Made Dr. van Vliet's recommended changes	TGC
2/8/2017	3.0	Made Dr. van Vliet's recommended changes	ERD
2/9/2017	1.2	Made Dr. van Vliet's recommended changes	CD
2/9/2017	1.2	Made Dr. van Vliet's recommended changes	RRM
2/9/2017	1.4	Made Dr. van Vliet's recommended changes	TGC
2/9/2017	4.0	Made Dr. van Vliet's recommended changes	ERD

Client Documents

Opening Statement

Milestone 5 of the Computer Science Learning Center (CSLC) system has been completed. The portal and database system project continues to be on time and on budget.

Executive Summary

The purpose of Milestone 5 is to create the Tracking Gantt chart, the Entity-Relationship Diagram, and the Full Relational Database Design. The Milestone Manager for Milestone 5 is Brian Hodges.

Milestone 5 documents include:

- **Tracking Gantt Chart (TGC)**

  The **TGC** tracks the project progress and provides a graphical look at the schedule for the remainder of the project. The chart has a baseline which is then compared to the actual start and finish times of tasks as they are completed to identify variances from the planned schedule.

  The **TGC** establishes activity duration tracking and will be updated as each milestone is completed.

  Further information can be found in Appendix C.

- **Entity-Relationship Diagram (ERD)**

  The **ERD** is a diagram that demonstrates the fundamentals of the database design. For the CSLC database, each student will create a ticket to request tutoring services. The purpose of the **ERD** is to determine the data that needs to be stored in the database and what form that content may take. The structure of how the data will be stored is also determined and diagramed.

  The **ERD** is the design of the database and is comparable to the blueprint of a building.

  Further information can be found in Appendix D.

- **Relational Database Design (RDD)**

  The **RDD** is the implementation of how the **ERD** will be built. The **ERD** was carefully reviewed, and the benefits of the RDD is that data can be entered, stored reliably, and retrieved so that a variety of reports can be created from the data.

  Further information can be found in Appendix E.

❖ Implications for Client

Milestone 5 has no implications for the CSLC.

❖ Items for Approval

None

Tracking Gantt Chart

Please see Appendix C

Entity Relationship Diagram

Please see Appendix D

Relational Database Design

Please see Appendix E

Milestone 6

February 12 – March 4, 2017

Milestone Manager:
James Hopkins

Milestone 6 Contents

Control Documents

Roles and Responsibilities Matrix

Department: The Peter Kiewit Institute

Product/Process: Database of Students

Document Owner: Alpha Group Beta

Project or Organization Role: Design and Implement Database

Version	Date	Author	Change Description
1.0	2/23/16	James Hopkins	Initial Version

Title	Role	Responsibilities
Dr. Paul Van Vliet Project Manager	The Project Manager is responsible for judging the quality of the project documents. The Project Manager ensures that the project is delivered on time, within budget, and to the required quality standards.	• Manage and lead the project team. • Manage the coordination of the partners and the working groups. • Develop and maintain a detailed project plan.
James Hopkins Milestone Manager	The Milestone Manager is responsible for coordinating team efforts, maintaining the running document, maintains tracking documentation, and assists the other Managers with their documentation.	• Change Log • Roles & Responsibility Matrix • Running Document • Table of Contents • Meeting Minutes • Email Log • Tracking Gantt Chart
Brian Hodges Design Manager	The Design Manager is responsible for the design of the computer interface and hardware choices.	• Input/Output and Interface Design • Computer Architecture Design – actual hardware, hardware procurement
Chengcheng Zhang Proposal Manager	The Proposal manager is responsible for the RFP process.	• Request for Proposal

Title	Role	Responsibilities
Douglas Stahlnecker Operations Manager	The Operations Manager is in charge of client and planning documentation.	• Comprehensive Systems Controls Plan • Client Documents

Milestone Change Log

Date	Rev. No.	Change	Reference Section
2/24/2017	1.0	Initial release	Roles and Responsibilities Matrix (RRM)
2/25/2017	1.0	Initial release	Client Documents (CD)
2/25/2017	1.0	Initial release	Comprehensive Systems Control Plan(CSCP)
2/28/2017	1.0	Initial release	Request For Proposal (RFP)
3/1/2017	1.0	Initial release	Actual Architectural Design (AAD)
3/1/2017	1.1	Made Dr. van Vliet's recommended changes	Client Documents (CD)
3/1/2017	1.0	Initial release	Relational Database Design (RDD)
3/1/2017	1.0	Initial Release	Tracking Gantt Chart (TGC)
3/2/2017	2.0	Made Dr. van Vliet's recommended changes	Request For Proposal (RFP)
3/2/2017	1.2	Made Dr. van Vliet's recommended changes	Request For Proposal (RFP)
3/2/2017	3.0	Made Dr. van Vliet's recommended changes	Relational Database Design (RDD)
3/2/17	1.0	Initial release	Input/Output and Interface Design (UI)
3/4/2017	1.1	Final release	TGC

Client Documents

Opening Statement

Milestone 6 of the Alpha Group Beta database system has been completed. The systems development project continues to remain on time and on budget. The Milestone Manager for Milestone 6 is James Hopkins.

Executive Summary

Milestone 6 consists of the design of the system's architecture, controls, inputs and outputs, interface design, and updating the Tracking Gantt Chart.

Milestone 6 documents include:

- **Updated Tracking Gantt Chart (TGC)**

  The **TGC** tracks the project progress and provides a graphical look at the schedule for the remainder of the project. The chart has a baseline which is then compared to the actual start and finish times of tasks as they are completed to identify variances from the planned schedule.

  The **TGC** establishes activity duration tracking and has been updated and will continue to be updated as each milestone is completed.

  Further information can be found in Appendix C.

- **Actual Architecture Design (AAD)**

  The **AAD** describes the hardware that is needed for the system as well as the Computer Science Learning Center's procurement procedures for obtaining any new hardware or any new software.

- **Request for Proposal (RFP)**

  The **RFP** is a formal request detailing what hardware is needed for the web portal and database application system that will be built and what vendors will need to submit if they wish to supply the hardware for the system. The **RFP** details our need for hardware and the Computer Science Learning Center's existing hardware that the new hardware will need to be compatible with.

  The purpose of the **RFP** is to make vendors aware of our hardware needs so that they may submit bids to provide the hardware to us. After the **RFP**'s stated open window dates for submitting bids, all of the bids that have been submitted by vendors are carefully evaluated and weighted against the criteria set forth in the **RFP** as well as against all other bids and then one vendor is chosen to supply the hardware for the system based upon this evaluation of their bid.

  Further information can be found in Appendix F.

- **Comprehensive Systems Control Plan (CSCP)**

  The purpose of the **CSCP** is to ensure business continuity. This is accomplished by evaluating what threats exist for the system and create safeguards and security measures against those threats in order to prevent them from impacting the system.

  The **CSCP** identifies possible threats to the system. The threats range from hardware malfunctions to malicious hackers. For each listed threat, an associated control is identified. A disaster recovery plan is also described. In the event of a system failure, the disaster recovery plan is implemented; the disaster recovery plan consists of the steps that will be taken if system failure/outage occurs in order to bring the system back up. This plan also includes what will be done during the failure/outage in order to keep track of data accumulating during that time, and what will be done once the system is restored to bring it back to its pre-failure state and minimize the loss of any data.

- **Input/Output and Interface Design (IOID)**

  The **IOID** shows preliminary designs of the input and output screens. The format for data inputs is also described. In addition, sample output reports are shown and the usage of these reports is described.

  These designs are included in order to demonstrate what the screens for data input, the screens that show output, and what the reports will look like in order to begin the process of making improvements to these screens until they are intuitively usable as possible and also sufficiently attractive.

  Further information can be found in Appendix G.

❖ Implications for the Client

None

❖ Items for Approval

None

❖ Items for Client Feedback

- Input/Output and Interface Design

Updated Tracking Gantt Chart

Please see Appendix C

Computer Architecture Design

Actual Architecture Design

The Computer Science Learning Center currently possesses 3 Dell Precision M6800 laptops as clients for the portal's web interface. These computers run Windows 10 with Google Chrome as the web client.

The Computer Science Learning Center will also procure a wall mounted monitor with a raspberry pi to display status information in the room. To procure new hardware an acquisition request is sent to Mike Grove. If the request is approved, Mike Grove will then procure the hardware using the Computer Science Learning Center grant budget.

Request for Proposal

Please see Appendix F

Comprehensive Systems Controls Plan

Introduction

The purpose of the **Comprehensive Systems Control Plan (CSCP)** is to keep the web portal and database system operating continuously so that the Computer Science Learning Center business processes can operate without interruption. The **CSCP** identifies possible threats to the Computer Science Learning Center web portal and database. These threats range from hardware malfunctions to malicious individuals seeking to harm the system and/or extract privileged information. For each listed threat, an associated control is identified that will mitigate the effects of the threat or eliminate it completely. A disaster recovery plan is also described. The disaster recovery plan consists of a contention plan, a contingency plan, and a recovery plan.

Data Entry Controls

- **Threat: Invalid Data**
  Invalid data is entered into the system rendering the record useless, or worse, causing errors within the system. Invalid data is data that does not fall in one of the valid options for input in the field. These fields are ticket_status, section_id, and problem_type_id. Data that is absent for a required field (i.e. a blank input) would also be invalid data (this includes the above fields as well as student_email). A string input into a field requiring an integer or a non-date entered into a date field would also be invalid data.
- ✓ **Associated Control**
  Dropdown menus will only allow inputs to be selected from a list of valid inputs for fields that require that input comes from a predetermined list. The form will also not allow required fields to be left blank. Fields that expect an email will be validated to ensure that the input matches the something@something.something pattern. For fields expecting a date, the form will only accept dates and will properly format them. For fields expecting an integer, non-integers values and strings will not be accepted by the form.

- **Threat: Buffer Overflow Attack**
  Input exceeding the length of the expected input is maliciously entered by the user in a very precise manner in order to insert their own code to be run on the system.
- ✓ **Associated Control**
  The length of all input will be validated by the form and/or by the backend code so that no inputs that are longer than expected will be able to create a buffer overflow situation.

- **Threat: Cross Scripting Attacks**
  A user maliciously includes HTML tags in their input causing the browser to respond in ways not intended by the system. For example, an attacker might enclose an input in <script> tags in order to cause a browser that displays the information to be redirected to a website that downloads a virus.

- ✓ **Associated Control**
  All user input is sanitized by the backend code to remove any HTML tags from the input before it is saved to the database or before it is output to the screen outside of the initial form entry.

Output Controls

- **Threat: Privileged Information Displayed for Unauthorized User**
  Information that is for administrators only is displayed to the screen of a tutor or student. Information that is for tutors only is displayed to the screen of a student.
- ✓ **Associated Control**
  Tutors and administrators must be logged in to view privileged information and any information that is not supposed to be seen without proper credentials will not be displayed to anyone not logged in at the appropriate level.

- **Threat: Administrative Report Viewed by an Unauthorized Party**
  An administrative report is saved to a USB drive or is printed out on paper, and the report is viewed by or otherwise falls into the hands of an individual not authorized to view the report.
- ✓ **Associated Control**
  Administrators must be properly logged in in order to view, save, or print any administrative reports. Any administrative reports that are printed out on paper or saved to any digital media outside of the system become the responsibility of the administrator performing the activity and it is that administrator's duty to keep these reports confidential.

Database Controls

- **Threat: SQL Injection**
  A query is sent to the database that has been manipulated by a user to perform a malicious function, such as deleting records or tables, or displaying privileged information.
- ✓ **Associated Control**
  All user received information will be sanitized to remove characters that might alter an SQL query maliciously. Characters to be removed include quote marks, semicolons, equal signs, less than signs, greater than signs, and the word "true".

- **Threat: Corruption or Loss of Data**
  A system or security failure causes the data to be lost or incorrectly/maliciously manipulated, added, or otherwise changed.
- ✓ **Associated Control**
  The database is backed up nightly to minimize the loss from corruption or loss of data to an acceptable amount. This backup is stored on a separate part of the same server the system is stored on but, more importantly, a second copy is stored in the cloud in case the entire server is corrupted or otherwise fails.

Access Controls

- **Threat: Unauthorized Access to database**
  An unauthorized user is able to interact directly with the database. Information could then maliciously be viewed, deleted, added, or altered.
- ✓ **Associated Control**
  Access to the database is password protected and only administrators have the password. A password that is deemed secure by the head administrator is chosen, can only be chosen by him, and is changed quarterly. These password requirements are not hard coded and are at the discretion of the head administrator.

- **Threat: A Tutor Changes Another Tutor's Profile**
  A tutor changes the profile of another tutor in order to cause difficulty or confusion. For example, changing another tutor's password so that they cannot login, or changing another tutor's status to inactive as if they were no longer employed.
- ✓ **Associated Control**
  Tutors may only access their account by logging in with a username and password. Passwords are only stored as hash values to make them secure within the system.

- **Threat: Guessing Passwords or Brute Force Password Attacks**
  An unauthorized user tries to make a series of educated password guesses in order to log in as a tutor or administrator or uses technology to try a large volume of different passwords in hopes of gaining access.
- ✓ **Associated Control**
  In order to prevent easily guessed passwords, when setting up a password, the system only allows passwords that are at least 8 characters long and include a number and a symbol. In order to prevent brute force password attacks, after five unsuccessful login attempts, an account is locked and can only be unlocked by an administrator.

- **Threat: Unauthorized Access to Server**
  An unauthorized user accesses the server and deletes, adds, or alters system files. Or physical access to the server is gained by an unauthorized individual.
- ✓ **Associated Control**
  Access to the sever is also password protected. The server is kept in a securely locked room.

Software Controls

- **Threat: A Publicly Known Software Bug is Used to Exploit the System**
  A bug is discovered in a piece of software used by the system. An update to the software is released that fixes or patches the bug; this update may even state what the bug is that is being fixed. Now that the bug is publicly known, it is easy for a hacker to exploit it on any system using the software that has not applied the update.

- ✓ **Associated Control**
  An administrator will do a weekly check for software updates and apply any that are found. This can be accomplished with only a few simple inputs on the command line, or the process could be automated with a script that is run on a regular basis.

- **Threat: Viruses and Other Malware**
  A virus or other piece of malware makes its way onto the system and proceeds to do harm, possibly putting the entire system at risk.
- ✓ **Associated Control**
  The University of Nebraska Omaha employs anti-virus/anti-malware software will provide protection; however, this software will be outside of and not a part of our system.

Hardware Controls

- **Threat: Server Failure**
  The server that hosts the web application and database suffers complete failure and all access and all data is lost.
- ✓ **Associated Control**
  A backup of all software, code, and data is maintained and a new server can quickly be acquired and the system can be recreated using the backup.

- **Threat: Unauthorized Network Access**
  Unauthorized computers outside of the network connect with computers inside of the network and thereby interact with the network maliciously causing, among other things, denial of service.
- ✓ **Associated Control**
  The hardware firewall that protects the network keeps unauthorized computers outside of the network from accessing the network.

Disaster Recover

- **Contention Plan**
  During a disaster that renders the system inaccessible, the administrator on duty will assess the damage to the system and determine what needs to be done to make the system accessible again. During a fire in the server room, the existing overhead sprinkler system will spray the area with water in an attempt to put out the fire. In the event of a disaster requiring evacuation, no additional steps will be taken to restore the system and the Computer Science Learning Center will cease to offer tutoring sessions until it is deemed safe to return to the area.
- **Contingency Plan**
  Tutoring information will be logged on paper by tutors while the system is down and each paper, upon tutoring session completion, will be given to the administrator on duty to safely retain.
- **Recovery Plan**
  If restoration from a backup is required, then the system will be rebooted from the most recent backup. In the event of complete server failure, a new server

will be procured and the complete system will be recreated from software backups as well as data backups. This could potentially take a day or possibly even two. In concurrent events of system failure and required personnel evacuation, the system will be inoperable and unavailable until such time as an administrator is able to regain access to the property and begin the aforementioned process of bringing the server back up to a working state from back up or procure a new server and get the system running from backups. Once access to the system has been restored and the database has been recreated using the latest backup (if necessary), it can again be used as normal. The tutoring sessions that were logged on paper while the system was down will be input into the system by the administrator on duty or an appointed tutor as the administrator sees fit. Some data will be lost if it is necessary to recreate the database from the latest backup because backups are only made nightly. Any information entered from the start of the day up until the system went down will be lost. Because a disaster resulting in the need to recreate the database from backup will be a rare event, the amount of data lost will be nominal and this is an acceptable loss that can be tolerated.

Input/Output and Interface Design

Please see Appendix G

Milestone 7

March 5 – April 1, 2017

Milestone Manager:
Douglas Stahlnecker

Milestone 7 Contents

Control Documents

Roles and Responsibilities Matrix

Department: The Peter Kiewit Institute

Product/Process: Database of Students

Document Owner: Alpha Group Beta

Project or Organization Role: Design and Implement Database

Version	Date	Author	Change Description
1.0	3/7/16	Douglas Stahlnecker	Initial Version

Title	Role	Responsibilities
Dr. Paul Van Vliet Project Manager	The Project Manager is responsible for judging the quality of the project documents. The Project Manager ensures that the project is delivered on time, within budget, and to the required quality standards.	• Manage and lead the project team. • Manage the coordination of the partners and the working groups. • Develop and maintain a detailed project plan.
Douglas Stahlnecker Milestone Manager	The Milestone Manager is responsible for coordinating team efforts and maintaining the running document including tables of contents.	• Change Log • Roles & Responsibility Matrix • Running Document • Table of Contents • Email Log • Meeting Minutes • Structure Chart Collaboration • Run-sheets Collaboration

Title	Role	Responsibilities
James Hopkins Project Coordinator	The Project Coordinator is responsible for updating progress in the Project Management Chart and designating team members' roles in the software development as well as outlining development approaches.	• Updated Tracking Gantt Chart • Software Design Team Identification • Software Development Approach • Structure Chart Collaboration • Run-sheets Collaboration
Chengcheng Zhang Chart Specialist	The Chart Specialist is responsible for overseeing the creation of the Structure Chart and Run-sheets as well as creating the CRUD table.	• CRUD Table • Structure Chart Collaboration • Run-sheets Collaboration
Brian Hodges Software Specialist	The Software Specialist is responsible for identifying the OS and other software that will be included in the system as well as how software upgrades will be handled	• Commercial Software Component • Structure Chart Collaboration • Run-sheets Collaboration

Milestone Change Log

Date	Rev. No.	Change	Reference Section
3/7/2017	1.0	Initial release	Roles and Responsibilities Matrix (RRM)
3/9/2017	1.0	Initial release	Client Documents (CD)
3/11/2017	1.0	Initial release	Structure Chart (SC)
3/11/2017	1.0	Initial release	Run-sheets (RS)
3/11/2017	2.0	Incorporated Professor van Vliet's suggested changes	CD
3/12/2017	3.0	Incorporated Professor van Vliet's suggested changes	CD
3/14/2017	2.0	Incorporated Professor van Vliet's suggested changes	SC
3/15/2017	2.0	Incorporated Professor van Vliet's suggested changes	RS
3/15/2015	1.0	Updated for Milestone 7	Entity Relationship Diagram (ERD)
3/16/2017	3.0	Incorporated Professor van Vliet's suggested changes	SC
3/16/2017	3.0	Incorporated Professor van Vliet's suggested changes	RS
3/16/2017	1.0	Initial Release	CRUD Table (CT)
3/20/2017	1.0	Initial Update	Updated Tracking Gantt Chart (TGC)
3/21/2017	1.1	Additional Updates	
3/27/2017	1.0	Initial Release	Proprietary Software Component (PSC)
3/27/2017	1.0	Initial Release	Commercial Software Component (CSC)
3/28/2017	1.1	Incorporated Professor van Vliet's suggested changes	PSC
3/28/2017	1.2	Incorporated Professor van Vliet's suggested changes	PSC
3/28/2017	1.1	Incorporated Professor van Vliet's suggested changes	TGC
3/28/2017	1.1	Incorporated Professor van Vliet's suggested changes	CSC
3/29/2017	1.2	Incorporated Professor van Vliet's suggested changes	TGC
3/29/2017	1.2	Incorporated Professor van Vliet's suggested changes	CSC
3/29/2017	1.3	Incorporated Professor van Vliet's suggested changes	CSC

Client Documents

Opening Statement

Milestone 7 of the Computer Science Learning Center (CSLC) system has been completed. The portal and database system project continues to be on time and on budget. The Milestone Manager for Milestone 7 is Douglas Stahlnecker.

Executive Summary

Milestone 7 consists of the updated Tracking Gantt Chart, specifics regarding the commercial software that the system will use, the system's proprietary software design team and development approach, the Structure Chart of the system, the Run-sheets, and the CRUD Table.

Milestone 7 documents include:

- **Updated Tracking Gantt Chart (TGC)**

  The **TGC** tracks the project progress and provides a graphical look at the schedule for the remainder of the project. The chart has a baseline which is then compared to the actual start and finish times of tasks as they are completed to identify variances from the planned schedule.

  The **TGC** establishes activity duration tracking and has been updated and will continue to be updated as each milestone is completed.

  Further information can be found in Appendix C.

- **Commercial Software Component (CSC)**

  The **CSC** specifically describes the commercially available software that will be used for the system. The operating system is identified, including the manufacturer, name, and version number. Any commercial packages used are also identified, including their manufacturers, names, and version numbers.

  The way that commercial software upgrades will affect the system and how these upgrades are to be handled are also included in this document.

- **Proprietary Software Component (PSC)**

  The **PSC** details the design for the software that will be created for the system. The **PSC** details which tasks will be assigned to which team members. This includes who will perform what parts of the design, coding, testing, and documentation of the software.

  The **PSC** also details the overall development approach for this project. It specifies that the development will be a top-down approach, which means that development will begin with the full overview of the system and break that down

into smaller parts. Those parts are broken down into even smaller parts until each part performs just one specific function. A part of the system that performs just one function is called a module. The details of the module are designed, such as what the module does and how it will do it, and then the coding is performed at the modular level. For example, the New Professor module displays a form so that an administrator can enter the information for a new professor. When the submit button is clicked, the Submit Professor module is triggered, this module puts the information from the form into the proper syntax for saving it to the database and triggers the Save Professor module, which is a library call that actually saves the record to the database.

- **Structure Chart (SC)**

  The **SC** is a graphical representation of the software modules that make up the system. First and foremost, this shows the developers what modules they will need to design and build. In addition, the modules are logically arranged on the **SC** to show how they will be used in the system. Arranging all of the modules of the system graphically in this way allows the system developers to easily see which modules of the system interact directly with which other modules and what information or data is passed between those modules. For example, once an administrator has entered the information into the form in the New Professor module and clicked submit, the information that was entered into the form is passed to the Submit Professor module, which triggers the Save Professor module and passes it the information in the proper syntax to be saved to the database. If the save is not successful for some reason, an error message is passed back to the Submit Professor module.

  Further information can be found in Appendix H.

- **Run-sheets (RS)**

  The **RS** contains a run-sheet for four different modules of the system. The purpose of the **RS** is to provide documentation, in a specific format, of the function that each module of the system performs and some information about how that function is performed. This is absolutely necessary so that the developers can see what the module needs to do when they perform the coding for the module. For example, the run-sheet for the New Professor module explains that the module allows an administrator to enter information into a form, what fields the form has, any validations performed on the information entered into the form, and that when the submit button is pressed, the information from the form is passed to the Submit Professor module.

  Further information can be found in Appendix I.

- **CRUD Table (CT)**

  The **CT** is a table that shows the interactions between modules and the tables of the database. A database table is related data that is grouped together in a

structured format; for example, the Professors table contains professors' first names, last names, and a unique number that identifies them within the database. The **CT** defines what actions a module will perform upon the database and the privileges necessary for those actions.

The purpose of the **CT** is to allow the developers to easily see how modules will be interacting with the database so that only the necessary database privileges are granted to modules. For instance, the New Professor module does not directly interact with the database and therefore needs no privileges. The Save Professor module, however, needs to be able to create a new Professor record in the database, so it needs to be granted the privilege to do so. Granting unnecessary privileges could create security issues. For example, if the New Professor module was given privileges to modify records in tables that only administrators should be able to modify, a hacker that found a bug that enabled them to manipulate the module could easily manipulate records in tables that the module should have been excluded from manipulating in the first place.

The **CT** also provides an important maintenance benefit. If modifications are made to a table in the database, looking at the **CT** immediately makes it clear which modules interact with the table and will therefore need to be modified as well so that those modules can accommodate the changes to the table.

Further information can be found in Appendix J.

❖ Implications for the Client

None

❖ Items for Approval

None

❖ Items for Client Feedback

None

Updated Tracking Gantt Chart

Please see Appendix C

Commercial Software Component

Server Operating System:

- Any operating system with support for the version of Python being used

Server Commercial Packages:

- **Name:** Python
  - **Version:** 3.4
  - **Future Versions:** software can be upgraded to future versions that meet the following requirements
    - Must allow the same syntax as Python 3.4
    - Must include these modules without major changes:
      - __builtins__
      - argparse
      - csv
      - datetime
      - io
      - os
    - Must have support for the versions being used of the modules listed below
  - **Manufacturer:** Python Software Foundation

- **Name:** Flask
  - **Version:** 0.12
  - **Future Versions:** software can be upgraded to future versions that meet the following requirements
    - Must not remove any current API features
  - **Manufacturer:** Armin Ronacher

- **Name:** SQLAlchemy
  - **Version:** 1.1.5
  - **Future Versions:** software can be upgraded to future versions that meet the following requirements
    - Must not remove any features of the ORM or query APIs
  - **Manufacturer:** Michael Bayer

- **Name:** Flask_SQLAlchemy
  - **Version:** 2.1
  - **Future Versions:** software can be upgraded to future versions that meet the following requirements
    - Must include the SQLAlchemy API
    - Must not remove the _QueryProperty structure
    - Must support the versions of Flask and SQLAlchemy being used
  - **Manufacturer:** Immunda Mitsuhiko

Client Operating System:

- Any operating system with support for the web browser being used

Client Commercial Packages:

- **Name:** Web browser
    - **Version:** any browser can be used that meets the following requirements
        - Must support HTML5 and JavaScript
    - **Recommended Version:** Chrome 56
    - **Recommended Manufacturer:** Google

Browser support for HTML5 can be confirmed by checking for "datepicker" support on caniuse.com. JavaScript support can be confirmed by checking "File API" support on the same site.

Proprietary Software Component

Software Development Approach

Alpha Group Beta will be using a top-down software development approach. This approach is appropriate for our project because of our choice to use a relational database platform. In addition, the clear scope of our project has provided us with a thorough understanding of the requirements for the project. This has allowed us to take a big picture view of the project and break the software development down into components.

Software Design Team Identification

Alpha Group Beta team members have all participated in the software design process. A list of the team members who have/will lead stages of that process:

Design:	Brian Hodges
Coding:	Brian Hodges
Testing:	Douglas Stahlnecker
Documentation:	James Hopkins and Chengcheng Zhang

Structure Chart

Please see Appendix H

Run-sheets

Please see Appendix I

CRUD Table

Please see Appendix J

Milestone 8

April 2 – April 15, 2017

Milestone Manager:
Chengcheng Zhang

Milestone 8 Contents

Control Documents

Roles and Responsibilities Matrix

Department: The Peter Kiewit Institute

Product/Process: Database of Students

Document Owner: Alpha Group Beta

Project or Organization Role: Design and Implement Database

Version	Date	Author	Change Description
1.1	3/29/2017	Chengcheng Zhang	Initial Version

Title	Role	Responsibilities
Dr. Paul Van Vliet Project Manager	The Project Manager is responsible for judging the quality of the project documents. The Project Manager ensures that the project is delivered on time, within budget, and to the required quality standards.	• Manage and lead the project team. • Manage the coordination of the partners and the working groups. • Develop and maintain a detailed project plan.
Chengcheng Zhang Milestone Manager	The Milestone Manager is responsible for coordinating team efforts such as Software Testing Plan, and maintaining the running document including tables of contents.	• Change Log • Roles & Responsibility Matrix • Running Document • Table of Contents • E-mail Log • Meeting Minutes • Software Testing Plan Collaboration
James Hopkins Project Coordinator	The Project Coordinator is responsible for updating progress in the Project Management Chart and develop maintenance plan for team project based on the discussion with team members.	• Updated Tracking Gantt Chart • Maintenance Plan • Application Development Collaboration

Title	Role	Responsibilities
Douglas Stahlnecker Testing Specialist	The Communication Specialist is responsible for Client Documents and develop the Software Testing Plan based on the discussion with team members.	• Client Documents • Software Testing Plan
Brian Hodges Software Specialist	The Software Specialist is responsible for developing the Implementation Plan and complete Maintenance Plan based on the discussion with team members.	• Implementation Plan • Maintenance Plan Collaboration • Application Development

Milestone Change Log

Date	Rev. No.	Change	Reference Section
3/29/2017	1.0	Initial release	Roles and Responsibilities Matrix (RRM)
4/2/2017	1.1	Incorporated Professor van Vliet's suggested changes	Roles and Responsibilities Matrix (RRM)
4/4/2017	2.0	Updated for Milestone 8	Roles and Responsibilities Matrix (RRM)
4/6/2017	1.0	Initial release	Client Document (CD)
4/6/2017	1.0	Initial release	Software Testing Plan (STP)
4/6/2017	1.1	Incorporated Professor van Vliet's suggested changes	Client Document (CD)
4/6/2017	1.1	Incorporated Professor van Vliet's suggested changes	Software Testing Plan (STP)
4/8/2017	1.2	Incorporated Professor van Vliet's suggested changes	Software Testing Plan (STP)
4/8/2017	1.2	Incorporated Professor van Vliet's suggested changes	Client Document (CD)
4/10/2017	1.0	Initial release	Implementation Plan (IP)
4/11/2017	1.3	Incorporated Professor van Vliet's suggested changes	Client Document (CD)
4/11/2017	1.0	Initial release	Maintenance Plan (MP)
4/11/2017	1.1	Incorporated Professor van Vliet's suggested changes	Maintenance Plan (MP)
4/11/2017	5.1	Incorporated Professor van Vliet's suggested changes	Implementation Plan (IP)

Client Documents

Opening Statement

Milestone 8 of the Computer Science Learning Center (CSLC) system has been completed. The portal and database system project continues to be on time and on budget. The Milestone Manager for Milestone 8 is Chengcheng Zhang.

Executive Summary

Milestone 8 consists of the updated Tracking Gantt Chart,

Milestone 8 documents include:

- **Updated Tracking Gantt Chart (TGC)**

  The **TGC** tracks the project progress and provides a graphical look at the schedule for the remainder of the project. The chart has a baseline which is then compared to the actual start and finish times of tasks as they are completed to identify variances from the planned schedule.

  The **TGC** establishes activity duration tracking and has been updated and will continue to be updated as each milestone is completed.

  Further information can be found in Appendix C.

- **Software Testing Plan (STP)**

  The **STP** is a schedule of when the various phasing of testing will be performed and when any errors discovered in the various phases will be corrected. Software testing is a structured evaluation of the code to ensure that everything works and works correctly. Different phases of software testing test the various levels of the software, fixing errors at the lowest level they occur before advancing to testing the next hierarchical level of the system's software. Software testing saves costs because the sooner an error is found, the less it costs to fix that error; it costs much more to fix an error in a system that is in a production environment than it does to fix the error before the system is deployed and also does not require downtime. Testing also involves introducing the system to the client and giving them the chance to "test drive" the system before accepting it. Testing ensures to the client that the system works correctly and that it contains all of the necessary/requested functionalities, ultimately allowing the client to make the decision of whether or not to approve the system or to request changes before the system is implemented.

- **Implementation Plan (IP)**

  The **IP** describes how the system will be integrated into the client's environment and how it will replace the current system; it is how the system will be put into day-to-day operations. The **IP** identifies the tasks required to implement the

system, the order in which those tasks will be performed and estimates of the duration of those tasks. The tasks are described thoroughly and the plan for the post-implementation review is also included. The post-implementation review is a review of the whole system design process, and includes information from the users and managers and answers questions, such as does the system meet requirements and is it perceived as valuable, was the system on time and meet expected costs and benefits.

- **Maintenance Plan (MP)**

  The **MP** describes how the system will be maintained once it is put in place. The **MP** includes plans for 1) corrective maintenance, 2) adaptive maintenance, 3) perfective maintenance, and 4) preventive maintenance. Also included is the maintenance work order design.

  Corrective maintenance is simply correcting the system errors that were not detected during the testing process and occur once the system is in normal operations.

  Adaptive maintenance is maintenance that makes changes to the systems that are not intended to correct faults or errors in the system, but rather to allow the system to accommodate to new demands. Adaptive maintenance may be needed to deal with changes in the operating environment, or to accommodate new regulations. Adaptive maintenance may also be necessary to enable the system to perform functions that the system was originally not intended to perform, perhaps something such as automatically assigning a ticket to the first available tutor well versed in the particular area the student needs help with.

  Perfective maintenance is maintenance that is performed in order to make the system function more optimally. Examples of perfective maintenance may include rewriting programs in a more efficient language or even transitioning to a more efficient database software.

  Preventive maintenance involves reviewing the system periodically to make sure that it is still operating correctly and attempting to foresee system problems that may occur in the future so that they can be corrected before they become a problem. Preventative maintenance can also be used to begin preparing for foreseen opportunities that could be taken advantage of through modifications to the system.

  The maintenance work order design is a form that is filled out when maintenance is needed and/or requested.

❖ Implications for Client

- Change of tutoring session ticket system – the way that tickets are entered, processed, and tracked will change with the new system – client will need to learn and become used to the way things are done on the new system

❖ Items for Approval

- Douglas Bertelsen's presence is requested in the CSLC April 17 – 21, 2017 during regular CSLC hours, for database and server setup
- The presence of two tutors is requested in the CSLC April 19, 2017 10:30 AM - 2:00 PM, for human testing of system documentation
- The presence of several tutors is requested in the CSLC (student volunteers will also be in attendance) April 19, 2017 2:30 PM – 5:00 PM, for human testing of system
- Douglas Bertelsen's presence is requested in the CSLC April 21, 2017 10:30 AM – 2:00 PM, for preparing administrative data
- The presence of all available tutors is requested in the CSLC from April 24 – May 5, 2017 during regular CSLC, for training
- The presence of Douglas Bertelsen and any available tutors is requested May 5, 2017 from 10:30 AM – 3:00 PM, for Acceptance testing
- The presence of Douglas Bertelsen is requested May 11 – May 14, 2017, for the move to full reliance on the new system (old system taken offline)
- The presence of Douglas Bertelsen and available tutors is requested August 14 – August 18, 2017 during regular CSLC hours, for Post implementation review
- Anytime between May 8 – August 18, 2017 it is requested that Douglas Bertelsen or another appointed individual or individuals acquire the status display screen and archive data from the old portal

Updated Tracking Gantt Chart

Please see Appendix C

Software Testing Plan

Purpose

The purpose of the STP, as a whole, is to make sure everything works and get all the errors corrected and then ultimately for the client to "test drive" and hopefully give the go-ahead for the implementation of the system. The different phases of software testing test the various levels of the software, fixing errors at the lowest level they occur before advancing to testing the next hierarchical level of the system's software. Software testing saves costs because the sooner an error is found, the less it costs to fix that error; it costs much more to fix an error in a system that is in a production environment than it does to fix the error before the system is deployed and also does not require downtime. Testing also involves introducing the system to the client and giving them the chance to "test drive" the system before accepting it. Testing ensures to the client that the system works correctly and that it contains all of the necessary/requested functionalities.

Module Testing

- Testing of modules to see if they work correctly individually
- Complexity grows exponentially when modules are connected together - module testing identifies problems within the modules while relatively easy to trace and identify in the code – easier than trying to find cause of errors when multiple modules have been connected together
- Black box testing of modules performed with various valid and invalid inputs for each module - simulated data meant to mimic the data that the system would handle in production will be used for testing – in some cases this may include actual data, such as a class name or class number, but would not include an individual's information, such as actual tutor, student, or professor name
- Incorrect output received from black box testing and the data input that caused it are documented – white box testing is then performed to identify the problem in the code and the code is fixed, the correction is documented and the module is retested.
- All members of AGB will perform this testing
- Begin April 10, 2017
- Conclude April 15, 2017
- Errors to be corrected as found
- Errors must be corrected before April 17, 2017 to remain on schedule

Integration Testing

- Testing the interactions that occur between modules and the transfers of execution
- Identify and correct any problems that occur from data being passed between modules
- Identify and correct any transfer of execution errors - i.e., when a button is pressed, check to ensure the proper module is called
- Structure chart is consulted, modules that pass data between each other are integrated in isolation from other modules – black box testing is done with various valid and invalid inputs

- Incorrect output received from black box testing and the data input that caused it are documented – white box testing is then performed to identify the problem in the code and the code is fixed, the correction is documented and the integrated modules are retested.
- All members of AGB will perform this testing
- Begin April 17, 2017
- Conclude April 20, 2017
- Errors to be corrected as found
- Errors must be corrected before April 21, 2017 to remain on schedule

System Testing

- Testing the properties of the system over the whole system – system-wide testing
- Identify and correct any problems from normal (and maximum) usage in order to correct them before system is deployed in production
- Peak testing – 50 dummy tickets are created and submitted nearly simultaneously using a script – results are documented – criteria for success is all tickets being properly processed and, as applicable, stored properly on the database
- Performance testing – retrievals from database are timed and compared between low levels of three at a time to high levels of 50 at a time (high levels performed using a script) - results are documented – criteria for success: low-level volume transactions take less than 750 milliseconds and cause less than 750 milliseconds of "screen freeze"; high-level volume transactions take less than 1750 milliseconds and cause less than 1750 milliseconds of "screen freeze
- Recovery testing – data is entered and a backup is made – the database is recovered on a different machine and compared to the original to ensure that all data was accurately recovered
- Security testing – all controls in the Comprehensive Systems Controls Plan (CSCP) that are testable before implementation will be tested to ensure that they are all in place and are functioning properly (the CSCP can be found in Milestone 6 page 22) – example of a control not testable before implementation: the anti-virus/anti-malware software employed by the University of Nebraska Omaha
- Procedure testing and Human factors testing – the end user documentation will be tested to see if users not familiar to the system (i.e., two tutors who are not Brian) can follow the documentation to perform tasks on the system and troubleshoot – results are documented – a set of students and tutors not familiar with the system will be chosen to attempt to operate the system and provide feedback about the quality of the user interface, the system's ease of use, how well it meets their needs, and other opinions – results are documented
- All members of AGB will perform this testing
- Begin April 21, 2017
- Conclude May 2, 2017
- Errors to be corrected as found
- Errors must be corrected before May 5, 2017 to remain on schedule

Acceptance Testing

- Ensure to users that thorough testing has been performed and system will be usable and properly function once fully implemented
- Present all testing documentation to client and review this documentation with them
- Demo the system to the users, allow them to try the system, present the training plan so that the users can quickly learn the new system, present the implementation plan
- Client will sign off on system, giving the go-ahead for implementation, or closure items that need to be addressed before sign off will be made known by the client – all closure items will be documented
- May 5, 2017 10:30 AM at the CSLC
- All members of AGB will perform/participate in this testing
- Errors to be corrected as found – closure items to be resolved as soon as possible

Implementation Plan

Purpose

The purpose of the implementation plan is to ensure a smooth transition from the old tutoring portal to the new portal with as little loss of productivity as possible.

Technical Implementation

A new database instance and Linux container will be allocated for the new portal in the Holland Computing Center. A Google Administration Console will be set up for the authentication protocol. The portal itself will be cloned into the Linux container from GitHub. The administrator will then enter the operating data for the portal (courses, semesters, professors, sections, tutors, problem types, messages, and tutor proficiencies). Once the new system is running, the browser bookmarks on the computers in the tutoring center will be changed to the URL for the new portal.

The new system will be tested and tutors will be trained with a pilot test of the new system. During the pilot test the majority of CSLC traffic will be handled through the old portal, but selected students will use the new portal. This will help test for unexpected scenarios. At the end of the semester the post pilot review will begin to determine if the new system is worth keeping. If the system is accepted it will be fully implemented before the beginning of the Summer Semester.

A small TV will be acquired to display the status screen in the tutoring center. This TV will be attached to a microcomputer (such as a Raspberry Pi) to handle the web page.

The data from the old tutoring portal will be downloaded in a .xlsx format using Access's export tools before the old portal is taken offline. The old data will then be stored on Box for archival purposes. This information will not be converted to the new database.

All technical implementation actions are the responsibility of Doug Bertelsen.

Organizational Changes

Very little needs to be done to accommodate the changes in the tutoring portal. The system is similar to the existing portal, and tutor training will occur during the System Testing phase. A small number of tutors will receive in person training and will then train other tutors as their shifts overlap. The primary content of this training will involve training tutors to claim tickets before helping a student. Students will not need training on the new system as the only action available to students (opening tickets) is designed to be operated with no prior knowledge of the system.

Organizational changes are the responsibility of all tutors employed by the CSLC.

Post Implementation Review

After the summer semester, the new portal will be reviewed to ensure that it meets all requirements. Forms will be evaluated by tutors and administrators to ensure efficiency and reporting tools will be evaluated by administrators for clarity and conciseness. Additionally, some students may be interviewed to evaluate their experience with opening tickets and the status screen. If there is an overwhelming negative opinion, the old system may be reimplemented. Otherwise, any lingering issues will be recorded. These records will be passed to the developers to alter the system accordingly. In addition, tutors will evaluate the transition period and implementation of the new system, providing feedback on how smooth the transition was and how it could have been improved. Doug Bertelsen will evaluate the development process based on time, cost, his level of involvement, and the final product's benefits.

Task ID	Task Description	Date/Time	Implementer	Escalation Path
I-1	Allocate database and server instance Initials: _____	April 17 - April 21	Doug Bertelsen	IS&T "Attic"
I-2	Set up Google Administration Console Initials: _____	April 17 - April 21	Doug Bertelsen	IS&T "Attic"
I-3	Clone code from GitHub Initials: _____	April 17 - April 21	Doug Bertelsen	IS&T "Attic"
I-4	Prepare administrative data for new portal (see above for list of data types) Initials: _____	April 21	Doug Bertelsen	Brian Hodges
I-5	New portal pilot period Initials: _____	April 24 - May 5	All Tutors	Doug Bertelsen
T-1	Train primary tutors on use of new portal Initials: _____	April 24	Doug Bertelsen & Brian Hodges	None
T-2	Primary tutors train other tutors on use of new portal Initials: _____	April 24 - April 29	Primary Tutors	Doug Bertelsen & Brian Hodges
I-6	Post pilot review	May 5 - May 10	Doug Bertelsen	Brian Hodges

	Initials: _____			
I-7	Handover: take old system offline and bring new system to full live status Initials: _____	May 11 - May 14	Doug Bertelsen	IS&T "Attic"
I-8	Post implementation review Initials: _____	August 14 - August 18	All Tutors & Developers & Select Students	Doug Bertelsen & Brian Hodges
I-9	Acquire status display screen Initials: _____	May 8 - August 18	Doug Bertelsen	Mike Grove
I-10	Archive data from old portal in .xlsx format Initials: _____	May 8 - August 18	Doug Bertelsen	None

Maintenance Plan

The maintenance plan is a description on how the system is handled after implementation. Our lead coder (Brian Hodges) is a tutor in the CSLC. The CSLC will not require a maintenance plan involving AGB as the CSLC will be assuming maintenance responsibilities after implementation.

Corrective Maintenance

Corrective maintenance is the handling of system errors not found during the testing process. These errors present themselves once the system is in operation.

Brian Hodges will be available following implementation to handle corrective maintenance. At the point when Brian is no longer attending UNO and/or no longer employed in the CSLC, a tutor designated by the CSLC manager (or the manager) will take over this responsibility. Staff and/or students will use a Trouble Ticket Form (TTF) to identify issues (tickets). The tutoring staff will give the tickets to the CSLC manager. The CSLC manager will use the tickets to prioritize maintenance corrections and assign them to the designated tutor.

Adaptive Maintenance

Adaptive maintenance is maintenance not intended to correct faults or errors in the system, but rather to allow the system to accommodate to new demands. Adaptive maintenance may deal with changes in the operating environment or accommodate new regulations. Adaptive maintenance may also be necessary to enable the system to perform previously unintended functions - perhaps something such as automatically assigning a ticket to the first available tutor for that subject.

A tutor designated by the CSLC manager will handle future adaptive maintenance. Decisions on adaptive maintenance are the responsibility of the CSLC manager.

Perfective Maintenance

Perfective maintenance is maintenance performed to make the system function more optimally. Examples of perfective maintenance may include rewriting programs in a more efficient language or even transitioning to a more efficient database software.

A tutor designated by the CSLC manager will handle future perfective maintenance as needed. Decisions on perfective maintenance are the responsibility of the CSLC manager.

Preventative Maintenance

Preventive maintenance involves reviewing the system periodically to make sure that it is still operating correctly. This process also attempts to foresee future system problems. Preventative maintenance can involve preparations for foreseen opportunities that could be taken advantage of through modifications to the system.

Before each semester, the CSLC manager will evaluate the system regarding preventative maintenance. The CSLC manager will create a plan for any necessary changes and designate a tutor to perform the maintenance.

Computer Science Learning Center
Trouble Ticket Form

Name:		Your name
Type of Issue	☐ Bug ☐ Improvement ☐ Question	What does the problem involve?
Priority	☐ High ☐ Medium ☐ Low ☐ When we have Time	How urgent is the problem?
Severity	☐ Stopping work ☐ Must Have ☐ Nice to Have	How important is it that this gets fixed?
Description of the issue:		Be as specific as you can. Provide details on how and when the issue occurred, the steps taken to deal with the issue, and any other information that you want to provide.

Questions regarding this form can be directed to any tutor in the CLSC. If you have further questions please contact the CSLC manager at dbertelsen@unomaha.edu.

Appendices

Please see AGB_M8_Appendicies.docx