

SOFTWARE REQUIREMENTS SPECIFICATION

**Career Services Department Survey System (CSDSS)**

**By:**

Nicholas Estrada

Matt McGee

Andy Sorensen

Maxwell Winston

January 21, 2018

Table of Contents

[Table of Contents 2](#_Toc505535249)

[1. Introduction 4](#_Toc505535250)

[1.1 Purpose of this Document: 4](#_Toc505535251)

[1.2 Scope of the Development Project: 4](#_Toc505535252)

[1.3 Definitions, Acronyms, and Abbreviations: 4](#_Toc505535253)

[1.4 References: 5](#_Toc505535254)

[1.5 Overview of Document: 5](#_Toc505535255)

[2. Overall Description 6](#_Toc505535256)

[4. Product Functions 8](#_Toc505535257)

[5. User characteristics 10](#_Toc505535258)

[6. Constraints 10](#_Toc505535259)

[7. Assumptions 11](#_Toc505535260)

[8. Functions 11](#_Toc505535261)

[8.1 User Class – Survey-Taking User 11](#_Toc505535262)

[8.1.1 Functional Requirement 1.1 11](#_Toc505535263)

[8.1.2 Functional Requirement 1.2 11](#_Toc505535264)

[8.1.3 Functional Requirement 1.3 11](#_Toc505535265)

[8.1.4 Functional Requirement 1.4 12](#_Toc505535266)

[8.1.5 Functional Requirement 1.5 12](#_Toc505535267)

[8.1.6 Functional Requirement 1.6 12](#_Toc505535268)

[8.1.7 Functional Requirement 1.7 12](#_Toc505535269)

[8.2 User Class – Pollster 12](#_Toc505535270)

[8.2.1 Functional Requirement 2.1 12](#_Toc505535271)

[8.2.2 Functional Requirement 2.2 13](#_Toc505535272)

[8.2.3 Functional Requirement 2.3 13](#_Toc505535273)

[8.2.4 Functional Requirement 2.4 13](#_Toc505535274)

[8.2.5 Functional Requirement 2.5 13](#_Toc505535275)

[8.2.6 Functional Requirement 2.6 13](#_Toc505535276)

[8.2.7 Functional Requirement 2.7 13](#_Toc505535277)

[8.2.8 Functional Requirement 2.8 14](#_Toc505535278)

[8.2.9 Functional Requirement 2.9 14](#_Toc505535279)

[8.2.10 Functional Requirement 2.10 14](#_Toc505535280)

[8.2.11 Functional Requirement 2.11 14](#_Toc505535281)

[8.2.12 Functional Requirement 2.12 14](#_Toc505535282)

[8.2.13 Functional Requirement 2.13 14](#_Toc505535283)

[8.2.14 Functional Requirement 2.14 14](#_Toc505535284)

[8.2.15 Functional Requirement 2.15 14](#_Toc505535285)

[8.2.16 Functional Requirement 2.16 15](#_Toc505535286)

[8.2.17 Functional Requirement 2.17 15](#_Toc505535287)

[8.2.18 Functional Requirement 2.18 15](#_Toc505535288)

[8.2.19 Functional Requirement 2.19 15](#_Toc505535289)

[8.2.20 Functional Requirement 2.20 15](#_Toc505535290)

[8.2.21 Functional Requirement 2.21 15](#_Toc505535291)

[8.3 User Class – Site Administrator 16](#_Toc505535292)

[8.3.1 Functional Requirement 3.1 16](#_Toc505535293)

[9. Performance Requirements 17](#_Toc505535294)

[10. Logical Database Requirements 17](#_Toc505535295)

[Tables will be formatted as follows: 17](#_Toc505535296)

[Pin/Survey Table: 17](#_Toc505535297)

[Question Table: 17](#_Toc505535298)

[Results Table: 17](#_Toc505535299)

[Pollster Account Table: 17](#_Toc505535300)

[11. Software System Attributes 18](#_Toc505535301)

[11.1 Security 18](#_Toc505535302)

[11.1.1 Security Requirement 1 18](#_Toc505535303)

[11.1.2 Security Requirement 2 18](#_Toc505535304)

[11.1.3 Security Requirement 3 18](#_Toc505535305)

[11.1.4 Security Requirement 4 18](#_Toc505535306)

[11.1.5 Security Requirement 5 19](#_Toc505535307)

[11.2 Availability 19](#_Toc505535308)

[11.2.1 Availability Requirement 1 19](#_Toc505535309)

[11.2.2 Availability Requirement 2 19](#_Toc505535310)

[11.3 Maintenance 19](#_Toc505535311)

[11.3.1 Maintenance Requirement 1 19](#_Toc505535312)

[11.4 Portability 20](#_Toc505535313)

[11.4.1 Portability Requirement 1 20](#_Toc505535314)

[Document Ratification 21](#_Toc505535315)

# 1. Introduction

## 1.1 Purpose of this Document:

The purpose of this SRS document is to outline requirements for Eastern Washington University’s Career Services Department Survey System (CSDSS). The system will be web-based, built using a combination of HTML, CSS, JavaScript, a mySQL database, and responsive webforms. It will be operating system independent and accessible with any standard compliant browser.

## 1.2 Scope of the Development Project:

The Career Services Department Survey System (CSDSS) will be a web platform managing the surveying of various departments at Eastern Washington University to assess awareness and investment in the Career Services program.

## 1.3 Definitions, Acronyms, and Abbreviations:

Admin – Administrative User, in this case a member of Career Services

User – A person who uses this application in either a survey-taking or administrative capacity

Database – A collection of information monitored by this system

CSDSS – Career Services Department Survey System

EWU – Eastern Washington University

MySQL – My Structured Query Language, a programming language used to create and interact with databases

Webform – A collection of user interface components on a website designed to solicit information from a user

HTML – Hyper Text Markup Language, a language for content creation and formatting in the creation of a website

CSS – Cascading Style Sheets, a programming language to define appearance and animation of web elements.

SRS – Software Requirements Specifications, a written document that describes all the functions of a proposed system and the constraints under which it must operate. For example, this document.

Bootstrap – A robust CSS and Javascript Library full of functions to allow for easy use of deployment to multiple web browsers

## 1.4 References:

https://getbootstrap.com/

https://www.mysql.com/

https://www.ewu.edu

https://www.javascript.com/

## 1.5 Overview of Document:

This document contains all of the software requirement specifications. It contains an Overall Description of the product, the purpose and goal of the product, how it is going to work, and what technologies are to be used to make it work. We will also outline and describe specific components of the project.

# 2. Overall Description

This section will give an overview of how the project will work and how the individual parts of each system interact and how each of the parts function. This section will also talk about the potential users of the system, and what level of access each level of user has. Finally, there will be a discussion of the possible constraints and assumptions of the system.



3. Product Perspective

This will be a stand-alone web application. The only planned integration between this and other systems will be through hyperlinks to resources pertinent to the results of the survey. Other forms of integration may be applied (as required or restricted by Eastern Washington University), such as matching color and design schemes of other web pages related to EWU.

It will be run on a web hosting service to be determined either by EWU standards, or by agreement between EWU Career Services and the Group6 development team. The final product (software) will be delivered to, and remain in the possession of, Nate Bryant at EWU Career Services. Any further distribution of the software will be at his discretion.

All information gathered by the software will be stored in a database on the same host as stated above, and accessibility to it will be determined by EWU Career Services.

Hardware requirements are minimal. Internet access is required for users and administrators. For the administrator, the ability to collect and/or disseminate information will require text and/or spreadsheet editing software.

Space requirements are dependent on the size of the finished software product plus the size of the database created. The size of the table containing questions is determined by the administrator of the survey. The size of the table containing answers will be relative to that, multiplied by the number of people who take the survey. In total, while it is impossible to determine the actual size required of all processes, it should be insignificant in light of the planned functionality and intended use of the product in relation to the storage capabilities of modern computers and web-hosting services.

Communication between users and administrators of this software will be, initially, in the form of an invitation sent by the administrator to a department chair, faculty member, or other person with whom Career Services collaborates. Invitations will be compartmentalized: Each invitation sent will include a department-specific PIN number that can be further distributed among faculty, who may then access and complete the survey on behalf of the department they work in. The pin number will expire one week from the time it is created. The interaction between the survey-taker and the system will consist of multiple questions in the form of true/false, multiple choice, and rating scales. There will also be at least one text box for the user to provide feedback explaining answers and/or to offer suggestions or questions. This will not be a required field. Results from this field will be separate from the rest of the survey, so that anonymity is preserved, and so that the administrator can receive and view them without the possibility of survey results being unintentionally skewed by their content. No personal information will be collected from survey recipients, keeping the results completely anonymous, except in terms of department. Information collected will be fully accessible to the administrator in any form they choose, and may also be removed or deleted at their discretion.

## 4. Product Functions

1. Invitation page/function: Survey administrator enters departments and then email addresses by department. When submitted, software generates a PIN number for each department entered, and sends an invitation with the appropriate PIN to each email recipient entered.

All pages will be responsive, and so, will be functional on any size device.

1. Recipients receive an email titled (something like) “You have been invited to participate in a Career Services survey”, with an explanation of the survey, the PIN number for their department, and link (or “Accept Invitation” button) to the Survey web page.
2. When the user clicks on the invitation link, they are taken to a “landing page” for the user (survey-taker), with an extended welcome message describing the goals of the survey, a description of the survey itself (explanation of anonymity, time required, types of questions included), and a “start” button.
3. When the start button is clicked, the user is asked a series of questions designed to assess their knowledge and opinions about Career Services programs and offerings, and their willingness to collaborate on future projects. These are in the form of multiple choice, true/false, and rating/ranking type questions. There will also be a question to determine whether the person taking the survey is someone whose responsibilities include engaging with Career Services. This may be used to weight specific results within a department. If the survey is not completed, the results are not saved. The same survey may be re-taken by the same person for as long as the PIN is active.

There will also be a comment box provided, which will not be a required field for the user. This may be used to provide explanations, ask questions, or submit commentary.

1. When the survey is complete and submitted by the user, they are redirected to another page with content determined according to the results of the survey. If the user’s ranking is in the higher levels (above the bottom level), they are provided with the level they have achieved, how they compare to other departments (as an incentive to engage further), encouraged to take steps to progress further on the continuum, and provided with links to Career Services offerings that they may not know about (or that address questions that they didn’t answer in the affirmative/correctly). These links and resources will be provided and updated by Career Services.

If the user’s ranking is at the bottom level, they will be directed to more basic Career Services offerings, and encouraged to get more involved. This page may also include success stories from other departments, in order to encourage further development. The user will not be told that they were ranked at the bottom-most level.  
If the survey results for a department are removed or deleted for reassessment (possibly in the case of changing positions among faculty), or if the response rate from a department is very low, the relationship level for a department may be reduced when new surveys are submitted.

1. The results of the survey will be stored in a database. Size permitting, this database (detailed further in the Product Perspective section of this document) can be stored indefinitely. Any entries in the text box will be stored separately and/or emailed to the administrator of the survey for review.
2. Another web page, accessible to only the survey administrator, will provide functionality to:

-Retrieve survey results by individual, by department, by groups of departments, or by attributes (relationship level or individual question) in a way that can be saved or printed. Probably an Excel spreadsheet.

-Aggregate results: Mean, median, and mode for survey scores by user, department, or attribute (other functionality may be added later, time permitting, as agreed to between Career Services and Group6).

- Change questions in the survey

- Change information and links to Career Services resources provided to users after the survey is completed

- Change the weight of each question, and whether a given question is used to determine the relationship level of a department.

- Change survey administrator and/or password

## 5. User characteristics

There will be two types of users that use this system: staff and department heads, and the admin of the system in the Career Center. Each type has different levels of access due to their different uses of the system.

The staff and department heads may interact with the web page to take the various surveys that have been set up. This means that the users in this group must be able to see the questions presented, as well as choose the potential survey they wish to take from some sort of menu system.

The admin will also interact with the web page, but will be given more access to the information within. With this access, the admin will be able to view information on the surveys taken based on results from each department to find out each department’s relationship with the Career Center. The admin will also be able to create a print screen to print off the information. This will enable the admin to present the results to EWU University officials to show the relationship of each department with the Career Center. The administrator will also have access to the database containing the questions for each survey and will have the ability to update the questions to fit changes in requirements for each relationship level as well as create new surveys.

## 6. Constraints

The user’s Internet connection is one of the possible constraints for this project. Since the system is hosted as a webpage, and the system has to process data from the database over the Internet, the connection must be there and be fully functioning in order for the system to properly function.

The survey requires a period of 5-7 minutes maximum for the user to complete. The survey questions should be of a number and complexity such that they absolutely do not take longer than that time period to complete.

Another constraint will be the sizes of the database of the system. Since the database is shared among all users, even if the level of access is the same, there may be need for a queue of incoming requests. This could involve, for example, preventing read access to the results database while data is being written to it by users taking the survey. This will increase the time to fetch and process data, but it should be a negligible amount.

The final constraint is that the user must be in some way connected to EWU servers either by signing in to campus wifi, using a wired connection on campus, or by using a VPN to connect to campus networks while off-campus. This does limit the portability of the application, but will provide better security from possible outside attacks.

## 7. Assumptions

One assumption made is that if a user uses a mobile web browser the system will function exactly the same. There will be additional design work needed in order to ensure that the web page will be readable on any screen size, but this should be the case.

Another assumption is that the user is already connected to the EWU network somehow, or has the capability to do so. If the user is not, then the webpages cannot be accessed due to security concerns.

# 8. Functions

## 8.1 User Class – Survey-Taking User

### 8.1.1 Functional Requirement 1.1

**USE CASE**: View questions in the database.

**DESCRIPTION**: A user should be able to view questions in the database via GET request.

**WHY**: To allow user access to the survey questions

### 8.1.2 Functional Requirement 1.2

**USE CASE**: Survey Taker Page: Login  
**DESCRIPTION**: The user should be able to log in to the survey page using their registered department PIN.   
**WHY**: To ensure that only registered staff members can take the survey, which will ensure security and ensure that one department can’t flood the survey with results to pad their stats.

8.1.3 Functional Requirement 1.3  
**USE CASE**: Survey Taker Page: Start Survey Button  
**DESCRIPTION**: Once the user has logged in with their assigned username and PIN, they will be prompted to begin the survey by pressing a button  
**WHY**: To let the system know when to start displaying questions

**Figure 1: Example of what radio buttons look like**

8.1.4 Functional Requirement 1.4  
**USE CASE**: Survey Taker Page: Select Response To Question  
**DESCRIPTION**: A user should be able to choose a response to a question by selecting a radio button with the appropriate response that applies to them.  
**WHY**: To link user relationship level to the responses chosen

8.1.5 Functional Requirement 1.5  
**USE CASE**: Survey Taker Page: Display Links  
**DESCRIPTION**: After the survey has been completed, a new page will display links to the user, providing Career Services resources  
**WHY**: To help educate the user on the benefits of the Career Center and to improve their relationship level

### 8.1.6 Functional Requirement 1.6

**USE CASE**: Survey Taker Page: Comment Box  
**DESCRIPTION**: At the end of the survey, before submission, the user will be able to leave feedback on the survey  
**WHY**: To ensure that the survey is to the liking of the staff, as well as to provide the pollster with ideas of how to improve the questions asked by the survey

### 8.1.7 Functional Requirement 1.7

**USE CASE**: Survey Taker Page: Submit Survey Button

**DESCRIPTION**: A button at the end of the survey will submit user input to the database

**WHY**: To provide a logical end to the survey for the benefit of the user and to log results for comparison

## 8.2 User Class – Pollster

### 8.2.1 Functional Requirement 2.1

**USE CASE**: Pollster account setup page: Pollster account setup

**DESCRIPTION**: Pollster will be able to define a username, password, and recovery email address

**WHY**: To secure survey results, and allow for the reset of password

### 8.2.2 Functional Requirement 2.2

**USE CASE**: Pollster login page: Pollster Login

**DESCRIPTION**: The pollster will be able to log in to the pollster home page by entering a correct username and password

**WHY**: To ensure the pollster can access the pollster home page securely

### 8.2.3 Functional Requirement 2.3

**USE CASE**: Pollster login page: “Forgot your username/password?” link

**DESCRIPTION**: Hyperlink that, when clicked, sends an email to the pollster that will automatically provide the username, and that will allow him/her to change the password to the pollster account

**WHY**: To maintain security in case the pollster forgets their username and/or password

Figure 2: Pollster Login Concept

### 8.2.4 Functional Requirement 2.4

**USE CASE**: Pollster registration page: Pollster Registration

**DESCRIPTION**: A potential pollster may enter in necessary information to register for pollster privileges.

**WHY**: To allow the site administrator to approve and instantiate pollster accounts, giving them access to the system.

### 8.2.5 Functional Requirement 2.5

**USE CASE**: Pollster home page: Modify pollster information

**DESCRIPTION**: Pollster will be able to change username, password, and email

**WHY**: In case the pollster of the survey changes, or in the case that the pollster needs to change any of their personal information

### 8.2.6 Functional Requirement 2.6

**USE CASE**: Pollster home page: View survey results

**DESCRIPTION**: Admin will be able to access survey results page

**WHY**: To view and analyze results by department, perform aggregate operations on the results, and create graphs, tables, or lists from the results

### 8.2.7 Functional Requirement 2.7

**USE CASE**: Pollster home page: Modify survey questions

**DESCRIPTION**: Pollster will be able to add, modify, or remove survey questions

**WHY**: It is to be expected that the survey will change over time because of changing needs and/or a streamlined analysis system

### 8.2.8 Functional Requirement 2.8

**USE CASE**: Pollster home page: Modify feedback

**DESCRIPTION:** Pollster will be able to modify the links presented to the user on submission

**WHY**: Resources may change, survey may change, and some information may become more or less relevant over time

### 8.2.9 Functional Requirement 2.9

**USE CASE**: Pollster home page: Add questions to survey

**DESCRIPTION**: A pollster will be able to add new questions to the database via a simple form

**WHY**: To enable the pollster to populate the survey with questions

### 8.2.10 Functional Requirement 2.10

**USE CASE**: Remove question in database

**DESCRIPTION**: A pollster will be able to remove questions in the database via a simple form

**WHY**: To enable the pollster to remove incorrect or unneeded questions

### 8.2.11 Functional Requirement 2.11

**USE CASE**: Edit question in the database

**DESCRIPTION**: A pollster will be able to edit questions in the database via a simple form

**WHY**: To enable the pollster to edit existing

### 8.2.12 Functional Requirement 2.12

**USE CASE**: View questions in the database

**DESCRIPTION**: A pollster will be able to view questions in the database via a simple form

**WHY**: To enable the pollster to access questions for the survey

### 8.2.13 Functional Requirement 2.13

**USE CASE**: View specific questions in the database

**DESCRIPTION**: A pollster will be able to view all questions meeting custom-defined criteria in the database via a simple form

**WHY**: To enable the pollster to view all questions matching criteria for review

### 8.2.14 Functional Requirement 2.14

**USE CASE**: View results in the database

**DESCRIPTION**: A pollster will be able to view the results of surveys via a simple form

**WHY:** To enable the pollster to view/collect the results of the surveys

### 8.2.15 Functional Requirement 2.15

**USE CASE**: View specific results in the database

**DESCRIPTION**: A pollster will be able to view the results of surveys matching custom-defined criteria via a simple form

**WHY**: To enable the pollster to view specific results of the surveys matching a given criteria

### 8.2.16 Functional Requirement 2.16

**USE CASE**: View pin/dept. name pair

**DESCRIPTION**: A pollster will be able to view pin/dept. name pairs

**WHY**: To enable the pollster to view which pin corresponds to which department

### 8.2.17 Functional Requirement 2.17

**USE CASE**: Add pin/dept. name pair

**DESCRIPTION**: A pollster will be able to add pin/dept. name pairs

**WHY**: To enable the pollster to distribute the survey to departments

### 8.2.18 Functional Requirement 2.18

**USE CASE**: Edit pin/dept. name pair.

**DESCRIPTION**: A pollster will be able to edit pin/dept. name pairs

**WHY**: To enable the pollster to edit existing pin/dept. pairs

### 8.2.19 Functional Requirement 2.19

**USE CASE**: Delete pin/dept. name pair

**DESCRIPTION**: A pollster will be able to delete pin/dept. name pairs

**WHY**: To enable the pollster to delete existing pairs

### 8.2.20 Functional Requirement 2.20

**USE CASE:** Timer

**DESCRIPTION:** A timer to measure the duration of a survey by an individual user. Stored with user record in database.

**WHY:** To be used by the pollster to gauge the number and type of questions in the survey, and to gauge response rates by average time of completion.

### 8.2.21 Functional Requirement 2.21

**USE CASE**: View Comments in Dashboard  
**DESCRIPTION**: The pollster should be able to see comments made by the survey takers in the pollster page  
**WHY**: Will help let the pollster know if there’s any changes that need to be made to a particular survey

## 8.3 User Class – Site Administrator

### 8.3.1 Functional Requirement 3.1

**USE CASE**: Pollster account approval page: Pollster account approval

**DESCRIPTION**: Site Administrator will be able to approve or deny pollster account requests

**WHY**: To ensure only authorized users are allowed within the system.

Figure 3: Comment Box Concept



# 9. Performance Requirements

The system will be able to handle any number of concurrent logins from any number of terminals. As outlined by the client, the survey should be simple enough that the user should be able to complete it in 5-7 minutes.

# 10. Logical Database Requirements

The system must store all current pin/department number pairs, the current set of database questions as well as all results that are returned by surveys. All data shall be stored in mySQL format files administered via phpMyAdmin. The database will have 4 tables: one for pin/survey pairs, one for question data, one for results and one for pollster login information.

Additionally three account types will be created in order to access the various databases, an admin level with full general access, a pollster level with the ability to create surveys/view results and a user level account with read-only access to the questions and pin/survey table.

## Tables will be formatted as follows:

### Pin/Survey Table:

* + (Primary Key) **INTEGER** PIN – The Personal Identification number for use in login
  + **TEXT** SurveyName – The name of the survey associated with this PIN

### Question Table:

* + (Primary Key) **INTEGER** QuestNum – The Question Number
  + **TEXT** QuestionType – The type of question it is (i.e. True-False, Multiple Choice, Opinion-Based out of 4)
  + **TEXT** Question - The actual question
  + **TEXT** AnsA – First Answer to the question
  + **TEXT** AnsB – Second Answer to the Question
  + **TEXT** AnsC – Third Answer to the Question
  + **TEXT** AnsD – Fourth Answer to the Question
  + **INTEGER** QuestionAnswer – Correct answer to the question (if applicable)
  + **INTEGER** QuestionWeight – The weight of the question (or how important it is)

*(note: Answer variables can be null or blank depending on Question Type).*

### Results Table:

* + (Primary Key) **INTEGER** SurveyNumber - The record number
  + **TEXT** DeptName – Name of the department associated with the number
  + **TEXT** surveyAnswers – The answers selected by the person
  + **TEXT** relationLevel - The Relationship level associated with this record number based on the answers to the question

### Pollster Account Table:

* + (Primary Key) **INTEGER** Login - The login name of the account
  + **TEXT** pass – Hash of the password associated with the account

# 11. Software System Attributes

## 11.1 Security

### 11.1.1 Security Requirement 1

**TITLE**: Admin Login Security  
**DESCRIPTION:** If a person tries to login to the admin page with a non-existing account then the person will not be logged in and will be notified of the login failure with the message “Login Failure: Username and/or Password is incorrect”  
**WHY**: To ensure that non-admin users will not be able to access the admin page  
**HOW WILL IT BE TESTED?**: A test admin account will be created, and attempts will be made with incorrect information to ensure the admin account cannot be accessed without the correct information

### 11.1.2 Security Requirement 2

**TITLE:** Admin Account Security  
**DESCRIPTION**: An admin and IP address will not be able to login for a certain time period after three failed login attempts. Locking period will be approximately five minutes, during which time no further login attempts can be made until the time is over  
**WHY**: To ensure security of contained data as well as to protect from possible brute force attacks  
**HOW WILL IT BE TESTED?**: The lock will be triggered and login attempts will be made to ensure the system has been locked

### 11.1.3 Security Requirement 3

**TITLE:** User PIN Security  
**DESCRIPTION**: A non-admin user must use a valid PIN number in order to login to the survey page. If an improper PIN is used, the user will not be logged in and will receive the message “Login Failure: Invalid PIN Entered”  
**WHY**: To ensure that the system cannot be logged into by unauthorized users  
**HOW WILL IT BE TESTED?**: A set of example PINs will be created and login attempts will be made with random combinations to ensure that login attempts with improper information will not work

### 11.1.4 Security Requirement 4

**TITLE:** User Bruteforce Prevention  
**DESCRIPTION**: A one second timer will be implemented on the backend to ensure that a PIN from any source cannot be tried more than once per second  
**WHY**: To prevent bruteforce attacks from unauthorized users  
**HOW WILL IT BE TESTED?**: Multiple PIN login submissions will be sent with the intention of guessing a PIN to ensure that repeated attempts become discouragingly slow for attackers

### 11.1.5 Security Requirement 5

**TITLE:** PIN Expiration Timer  
**DESCRIPTION**: Any user PIN will expire one week after it is created, and any login attempt using expired PINs will be unsuccessful  
**WHY**: To prevent outdated surveys from being completed and contaminating database results  
**HOW WILL IT BE TESTED?**: Multiple PINs will be created and logins will be verified. Then, a week later, they will be tested again to ensure they are unusable

## 11.2 Availability

### 11.2.1 Availability Requirement 1

**TITLE:** Internet Connection  
**DESCRIPTION:** An internet connection to the EWU campus network is required to access the system  
**WHY**: In order for the webpage to connect to the database servers

### 11.2.2 Availability Requirement 2

**TITLE**: Accessibility Standards  
**DESCRIPTION**: The system will be tooled to meet the stands of the Web Content Accessibility Guidelines set out by the World Wide Web Consortium, just like the EWU website. This will be accomplished through adding accommodation options such as allowing the entire survey page able to be controlled through keyboard without mouse input and ensuring that any text used can be read through screen readers for visually impaired users.  
**WHY:** Will allow for a greater range of users, as well as ensuring that any user will be comfortable using this system.

## 11.3 Maintenance

### 11.3.1 Maintenance Requirement 1

**TITLE:** Testability  
**DESCRIPTION**: A test environment will be built to ensure that a survey works before it is pushed out to users  
**WHY**: To allow the admin to make sure there are no errors in the survey and that all questions are answerable and correct

## 11.4 Portability

### 11.4.1 Portability Requirement 1

**TITLE:** Mobile Portability  
**DESCRIPTION**: Both the admin and survey webpages will be accessible from mobile browsers as long as the user or admin is connected to campus Wi-Fi or can connect via VPN into the campus network  
**WHY**: To provide more options for the users to interact with the system even while away from their desktop  
**HOW WILL IT BE TESTED?**: A set of example PINs will be created and login attempts will be made from a wide variety of mobile browsers including stock Android Browser, mobile versions of Firefox, Google Chrome, and other mobile versions of desktop internet browsers

# Document Ratification



Date Signed



Date Signed