

**CSCE 361**

**Group 6**

**Software Requirements Specification  
Document**

**Version: (1.0)**

**Date: (09/22/2017)**

## Table of Contents

### **1. Introduction**

- 1.1 Purpose*
- 1.2 Scope*
- 1.3 Definitions, Acronyms, and Abbreviations*
- 1.4 References*
- 1.5 Overview*

### **2. The Overall Description**

- 2.1 Product Perspective*
  - 2.1.1 System Interfaces*
  - 2.1.2 Interfaces*
  - 2.1.3 Hardware Interfaces*
  - 2.1.4 Software Interfaces*
  - 2.1.5 Communications Interfaces*
  - 2.1.6 Memory Constraints*
  - 2.1.7 Operations*
  - 2.1.8 Site Adaptation Requirements*
- 2.2 Product Functions*
- 2.3 User Characteristics*
- 2.4 Constraints*
- 2.5 Assumptions and Dependencies*
- 2.6 Apportioning of Requirements*

### **3. Specific Requirements**

- 3.1 External interfaces*
- 3.2 Functions*
- 3.3 Performance Requirements*
- 3.4 Logical Database Requirements*
- 3.5 Design Constraints*
  - 3.5.1 Standards Compliance*
- 3.6 Software System Attributes*
  - 3.6.1 Reliability*
  - 3.6.2 Availability*
  - 3.6.3 Security*
  - 3.6.4 Maintainability*
  - 3.6.5 Portability*

### **4. Change Management Process**

### **5. Document Approvals**

### **6. Supporting Information**

## **1. Introduction**

### **1.1 Purpose**

The purpose of this document is to inform the system designers and implementers of the software specifications and functionalities required of this system.

### **1.2 Scope**

The software that will be created involves an attendance collection system and a web app used by students and professors that displays the attendance data and other analytic graphs and tables derived from the attendance data. The attendance collection system will generate timestamps of when students and teaching assistants (TAs) have entered and left the resource center using an identification card such as UNL's NCard. The web app will display attendance data publicly to students, and professors will be able to login to see further data analytics for students in their classes. The example use case will be the computer science Student Resource Center (SRC). The SRC is where most undergraduate TAs hold their office hours and it is a place for students to work on their computer science homework if they wish. For this proof of concept we are modelling the behaviors of students in the SRC. We will use the attendance data to tell students how busy the SRC currently is and which TAs are currently signed in. Professors will be able to see further analysis of the students in their classes in graph and chart format such as which students spend the most time in the SRC, what is the average time spent in the SRC per student or per class, how many students are there when each TA is present, how time in the SRC correlates with students' grades, etc. This software system will help students and professors increase the usage and efficiency of the SRC and other resource centers similar to it around campus.

### **1.3 Definitions, Acronyms, and Abbreviations.**

NCard - An ID badge issued by UNL for students

SRC - Student Resource Center

TA - Teaching Assistant

UNL - University of Nebraska - Lincoln

### **1.4 References**

This document does not reference any other documents.

### **1.5 Overview**

The SRS is divided into 6 sections, section 1 is the introductory section which gives a high level description of the system. Section 2 describes business requirements

including expected system functionality, interfacing requirements for the system, and constraints and dependencies. Section 3 describes in-depth functional use cases expected of the system for the developers to include. Section 4 describes the process by which any changes to the system's design will occur, including what types of changes are allowed, who approves them, and how they will be integrated into the final product. Section 5 lists the document approvers, the stakeholders on both the business and development side. Section 6 is supporting documents including any appendices or index items mentioned throughout the document.

## **2. The Overall Description**

### **2.1 Product Perspective**

#### **2.1.1 System Interfaces**

The system is independent of other systems. It does not need to interface with another system that is currently deployed.

#### **2.1.2 Interfaces**

The system shall have two user interfaces. The attendance collection system shall interface with users through an ID card reader, allowing users to check in and out of the SRC. The web app shall interface with users through a website, allowing users to see data collected by the system.

#### **2.1.3 Hardware Interfaces**

The system shall interface with a card reader to update the SRC attendance.

#### **2.1.4 Software Interfaces**

The system shall interface with the firefox, chrome and safari web browsers.

#### **2.1.5 Communications Interfaces**

The system has no communication interface requirements.

#### **2.1.6 Memory Constraints**

The system has no memory constraint requirements.

#### **2.1.7 Operations**

The web-app will be available to students and professors at all times so system maintenance will need to be scheduled beforehand and communicated to users of the

web-app. Nighttime is the best option for system downtime. The data analytics function will need access to historical data, at most a school year. If a professor requests historical data be saved for longer than a school year, this will need to be accommodated for. The new data will need to be backed-up daily to ensure data integrity. The attendance collection system only needs to be responsive during the hours of operation in the SRC, it can be deactivated other times if necessary.

### **2.1.8 Site Adaptation Requirements**

The system has no site adaptation requirements.

## **2.2 Product Functions**

The system will allow users to check in and out from the SRC. These comings and goings will be timestamped and stored by the system.

The system will allow users to view activity levels of the SRC, and see what TAs are currently in the SRC.

The system will allow professors to login to accounts to see what classes the students in the SRC are in, and see student and TA attendance records.

## **2.3 User Characteristics**

There are four users who will interact with the system: students, professors, TAs and admins. Students will use the system to gauge how busy the SRC is, and find out which TAs are currently hosting hours. The TAs will use the system to signify if they are in the SRC or not. Professors will use the system to gather data on SRC usage and student use of TA resources. Admins will maintain the system and ensure the integrity of the system.

## **2.4 Constraints**

The connectivity from the SRC to the web app needs to be near real-time for accurate population representation of people in the SRC. This may cause issues working with the hardware connectivity to reduce delays as much as possible.

## **2.5 Assumptions and Dependencies**

Students and TAs are assumed to be reliably checking in and out of the SRC. Software checks will need to be placed in the case of a student checking in and not out, or checking out but not checking in.

The TA schedule portion of the system assumes that the professors will upload and update TA schedules to accurately represent the most current TA SRC schedule.

The data analytics portion also assumes that students in the SRC are getting help from their TA when a TA from their class is there. This may not always be the case, the students could be working on other homework or seeing other TAs, therefore there may be some duplication of data in that case.

## **2.6 Apportioning of Requirements.**

The first iteration will store user data and information and provide a basic UI for the web app. The UI will contain functionality such as attendance history, TA schedule, TAs present, and current SRC population. The attendance collection system will allow for input, but will not yet implement a hardware card reader.

The second iteration will improve the UI and add responsiveness for mobile platforms. Login functionality will be added for professors and administrators. The attendance collection system will be upgraded to use a card reader to gather data.

Interfacing with live UNL student data such as grades, name, and class enrollment is out of scope for this proof of concept, but would need to be integrated if the system was deployed.

## **3. Specific Requirements**

### **3.1 External Interfaces**

#### **3.1.1 User Interfaces**

The system shall have two user interfaces. The attendance collection system shall interface with users through an ID card reader, allowing users to check in and out of the SRC. The web app shall interface with users through a website, allowing users to see data collected by the system.

#### **3.1.2 Hardware interfaces**

The system shall interface with a card reader to update the SRC attendance.

#### **3.1.3 Software interfaces**

The system shall interface with FireFox, Chrome and Safari web browsers.

#### **3.1.4 Communications interfaces**

The system has no communication interface requirements.

## **3.2 Functions**

### **Phase 1**

### **3.2.1. Timestamp Data Storage**

- 3.2.1.1. DESC: The system shall store timestamps.
- 3.2.1.2. STIM/RESPONSE: Upon receiving a timestamp, the system shall update the associated students timestamp record.
- 3.2.1.3. ADDITIONAL FUNCTIONALITY:
  - 3.2.1.3.1. The timestamp data storage shall keep track of if a student is currently in the SRC or not.
  - 3.2.1.3.2. The timestamp data storage shall keep a log of students who are TAs.
  - 3.2.1.3.3. The timestamp data storage shall keep a log of TA's office hours.
  - 3.2.1.3.4. The timestamp data storage shall automatically check a student out 5 hours after the student checked in.
  - 3.2.1.3.5. At 7 p.m. daily the timestamp data storage shall check out all students who are currently checked in.

### **3.2.2. Attendance Collection System**

- 3.2.2.1. DESC: The system shall have an attendance collection system.
- 3.2.2.2. STIM/RESPONSE: Upon receiving a card swipe, the attendance collection system shall alert the timestamp data storage of the user badging in/out.
- 3.2.2.3. DEPENDENCIES: 3.2.1

### **3.2.3. Web App Main Page**

- 3.2.3.1. DESC: The web application shall have a main page.

### **3.2.4. Responsive to Desktop**

- 3.2.4.1. DESC: The web application shall be formatted for desktop displays.

### **3.2.5. Current Population**

- 3.2.5.1. DESC: The web app main page shall display the current population of the SRC.
- 3.2.5.2. STIM/RESPONSE: Upon the user loading the main page, the system shall populate a graphic showing the current population of the SRC.

3.2.5.3.       DEPENDENCIES: 3.2.1, 3.2.3

3.2.5.4.       ADDITIONAL FUNCTIONALITY:

3.2.5.4.1.     A student shall be counted towards the total population if the timestamp data storage says the student is currently in the SRC.

3.2.5.4.2.     If a student is a TA and is in the SRC during their office hours, the student shall still be counted in the population.

### **3.2.6. TA Schedule**

3.2.6.1.       DESC: The web app main page shall display the schedule for TAs.

3.2.6.2.       STIM/RESPONSE: Upon the user loading the main page, the system shall populate a graphic showing the weekly schedule of TAs.

3.2.6.3.       DEPENDENCIES: 3.2.1, 3.2.3

### **3.2.7. TA Population**

3.2.7.1.       DESC: The web app main page shall display the current TAs who are in the SRC.

3.2.7.2.       DEPENDENCIES: 3.2.1, 3.2.3

3.2.7.3.       ADDITIONAL FUNCTIONALITY:

3.2.7.3.1.     A TA shall show up in the population **IFF** the TA is in the SRC during the TA's scheduled hours.

3.2.7.3.2.     If a TA badges in during their office hours, the TA will appear in the TA population.

3.2.7.3.3.     If a TA is in the SRC before their scheduled hours, and the scheduled hours begin, the TA will be displayed in the TA population.

3.2.7.3.4.     If a TA is in the SRC when their scheduled TA hours end, the TA will be removed from the TA population.

### **3.2.8. Mock Data**

3.2.8.1.       DESC: The system shall have the ability to be populated with mock data for testing and demonstration purposes.

3.2.8.2.       DEPENDENCIES: 3.2.1

3.2.8.3.       ADDITIONAL FUNCTIONALITY:

3.2.8.3.1.     Admins shall have the ability to add new students to the database.



Phase 2

**3.2.9. Professor Login**

- 3.2.9.1. DESC: The system shall provide a method for professors to create accounts and access more sensitive data.
- 3.2.9.2. STIM/RESPONSE: The user shall select the option to login, and the system shall redirect them to a login page.
- 3.2.9.3. DEPENDENCIES: 3.2.1, 3.2.3

**3.2.10. Professor Functions**

- 3.2.10.1. DESC: The system shall allow professors to login to the professor's main page, view graphs and charts displaying metrics collected from the scanner in the SRC and integrating student and TA data, and submit updates to the TA schedule.
- 3.2.10.2. STIM/RESPONSE: Upon professor login, professors can view SRC data analytics on their main page.
- 3.2.10.3. DEPENDENCIES: 3.2.1, 3.2.2, 3.2.8, 3.2.9
- 3.2.10.4. ADDITIONAL FUNCTIONALITY:
  - 3.2.10.4.1. Professors will have the option to export data to use for own analyzation purposes.

**3.2.11. Hardware**

- 3.2.11.1. DESC: The attendance collection system shall use a hardware device to read the ID cards of students.
- 3.2.11.2. STIM/RESPONSE: Upon badging in, the hardware will be able to obtain the ID code/number from the card.
- 3.2.11.3. DEPENDENCIES: 3.2.2

**3.2.12. Responsive to Mobile Screens**

- 3.2.12.1. DESC: The main web app shall be formatted for mobile screens.
- 3.2.12.2. STIM/RESPONSE: Upon a user loading the web app main page, the system will verify if the user is a mobile or desktop user and format the web page accordingly.
- 3.2.12.3. DEPENDENCIES: 3.2.3

### **3.3 Performance Requirements**

90% of all timestamps will be updated in the database and on the website within 1 minute. The system will be able to store 100 people at once in the SRC. The system will be able to hold information on 1000 people including students and TAs. In normal workload hours the system will receive 1 process per minute and in heavy workload hours the system will receive up to 20 processes per minute.

### **3.4 Logical Database Requirements**

The database will contain fields with strings and numbers in each of its tables. The database will be used anytime a student enters or leaves the SRC or a professor runs a report or populates a web page. The database updated every time someone enters the SRC and scans in using the provided hardware. The information on the database is viewed when the website is visited and when a professor asks for more analytics. The user input is constrained and therefore the input cannot cause unwanted reactions from the database. The database will keep information for 1 school year, unless requested otherwise by a professor.

### **3.5 Design Constraints**

#### **3.5.1 Standards Compliance**

In a future case where this proof of concept system is deployed, UNL students' grades will fall under FERPA regulations and will require additional security to protect students' privacy.

### **3.6 Software System Attributes**

#### **3.6.1 Reliability**

The network and hardware connectivity shall be reliable because the UNL network is well maintained by Information Technology Services so traffic to and from the database will most likely keep its integrity in transit. The application will maintain design integrity across different web browsers.

#### **3.6.2 Availability**

The system shall be accessible at any time via the website. The attendance collection hardware will need to be available and sending data to the database during the hours of operation in the SRC.

### **3.6.3 Security**

The login system will use a sufficiently complicated password hashing mechanism such as SHA-1 for encrypting passwords and will transfer data securely. The passwords themselves will need to be sufficiently complex to promote data security and the password file will need to be protected. Any user changes to the databases will be logged to have a record of any malicious attempts to gain access to the information. No user input is allowed which reduces the risk of malicious actors trying to gain access through the web app. Admins will have unique usernames and passwords and logins to the database will be logged.

### **3.6.4 Maintainability**

Since our system is a proof of concept, database admins will be able to add new student records and remove old student records. If this system is deployed, student information would be retrieved from a UNL server that would handle maintenance of student data.

### **3.6.5 Portability**

The system shall be backed up to a portable file type, preserving the SQL database structure, making porting the database very simple. The hardware attendance collection system and the web app would need to push and pull data to and from the new location.

## **4. Change Management Process**

All changes to requirements requested by the client must be submitted in writing to the team of developers. The development team will have 2 weeks to decide if the requirement change is feasible at that point in the project. 75% of the development team must vote to add the requirement change based on feasibility and the effect the change will have on the deadline and budget for the project for it to be added to the software requirements. The development lead will then email the client back with the approval or denial of their requirement change request. This email must be acknowledged by the supervisor of the development lead and the contact on the client side and this document will be updated accordingly to reflect the additional requirement.

## **5. Document Approvals**

### **Client Approvers**

## Software Requirements Specifications Document

Tyler Barker

Date

\_\_\_\_\_  
Andrew Heck

\_\_\_\_\_  
Date

\_\_\_\_\_  
Colton Harper

\_\_\_\_\_  
Date

\_\_\_\_\_  
Hannah Finnegan

\_\_\_\_\_  
Date

### **Development Approvers**

\_\_\_\_\_  
Tyler Barker

\_\_\_\_\_  
Date

\_\_\_\_\_  
Andrew Heck

\_\_\_\_\_  
Date

\_\_\_\_\_  
Colton Harper

\_\_\_\_\_  
Date

\_\_\_\_\_  
Hannah Finnegan

\_\_\_\_\_  
Date

## **6. Supporting Information**

There is no supporting information or documents at this time.