Software Requirements and Specification:

MCS 593 Tutorial Management Web Application

**Version 1.0**

**Prepared by:  
Alec Meyer  
Robert Padilla  
Anthony Prejean  
James Kubena  
Britton Ware**

**CS 3398: The Back Row**

March 10, 2015

Table of Contents

1. Preface 1

1. Introduction 1

2. Purpose 2

3. Document Conventions 2

4. Intended Audience 2

5. Definitions 2

2. Overall Description 2

1. Document Functions 2

2. User Characteristics 2

3. End-User Operating Environment 3

4. Design and Implementation Constraints 3

3. Functional Requirements 3

1. Use Case 1 - Logging in to the System 3

2. Use Case 2 - Scheduling a tutoring session 4

3. Use Case 3 - Cancelling a tutoring session 4

4. Non-Functional Requirements 5

1. Reliability 5

2. Robustness 6

3. Performance 6

# Preface

## Introduction

This requirements document has been completed by the students of Group “The Back Row” in CS 3398.251: Alec Meyer, Robert Padilla, Anthony Prejean, James Kubena, and Britton Ware. All members of the group agreed on all aspects of this document. Each member completed a portion of this document.

## Purpose

The purpose of this document is to define and describe the structure of a system to manage the Computer Science tutorial services offered in MCS 593. The system is comprised of a web application accessible from desktop and mobile browsers, with student and staff views.

## Document Conventions

This document uses a dot-number outline format to organize the information presented by the group. This information contained in this document has been organized categorically. The table of contents may be used as a reference to the sections of information.

## Intended Audience

This requirements document is intended for end users and developers of the system.

## Definitions

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **Authenticated User** | A TXSTATE student that is currently enrolled in at least one CS course. |
| **MCS 593** | The location of the tutorial service. Open 9AM-9PM Monday through Thursday and 9AM-5PM on Fridays. |
| **Tutoring Session** | Access to a workstation in MCS 593 during a specified time with hour long increments. |
| **Walk-In** | A student may still come to the tutoring lab and receive tutoring without scheduling an appointment (these spots will be limited). |
| **Workstation** | A Dell Optiplex 990 computer with dual screen setup. |

# Overall Description

## Document Functions

This document shall describe the various functionalities of the tutorial management system.

## User Characteristics

The tutorial system is intended for students and lab staff. Since the lab is meant to be used solely for Computer Science tutoring, the system will only be accessible to students that are enrolled in at least one Computer Science during the current semester. The second set of users will be comprised of lab staff that have scheduled working shifts in the MCS 593 tutoring lab. The tutorial application will exist at the URL “tutoring.cs.txstate.edu” (tentative URL until full approval by department chair, Hong Shi)

## End-User Operating Environment

The system will be accessible on desktop and mobile devices with internet browsing capabilities. All users must have internet access in order to interact with the application.

## Design and Implementation Constraints

Application can only be accessed by CS students enrolled in at least ONE CS course during the current semester and current CS Lab Staff employees.

# Functional Requirements

## Use Case 1 – Logging in to the system.

Priority: Essential  
Trigger: Accessing the URL  
Precondition: server accessible, user not authenticated.

Basic Path:

1. User enters NET ID and Password.

2. Application checks status of student being enrolled in at least one CS course during the current semester.

3. Branch:

A. Either accepts user login information and proceeds to scheduling a tutoring session

B. OR denies user access and prints information to the screen about the steps to request an exception override.

Requirements:

1. The application must have access to the database of current CS students each semester.

2. The user must have uninterrupted internet access to be authenticated.

## Use Case 2 – Scheduling a tutoring session.

Priority: Optional  
Trigger: Successful Authentication  
Precondition: Application running, user authenticated, No more than two hours already scheduled.

Basic Path:

1. User selects week to view tutoring availability.

2. Check currently scheduled tutoring sessions and returns only dates and times that are available to be booked.

3. User selects available time to receive tutoring.

4. User receives confirmation of selection and an email is sent to their TXSTATE email (“<<NETID>>@txstate.edu”) from tutoring@cs.txstate.edu.

Requirements:

1. There must be an internet connection and the application must have access to the schedule database.

2. The student can only select a tutoring session if they have not scheduled more than 2 other future sessions (max of 3 sessions scheduled at any one time).

## Use Case 3 –Cancelling a tutoring session.

Priority: Optional  
Trigger: Successful Authentication.  
Precondition: Application on, user authenticated, at least one tutoring session scheduled.

Basic Path:

1. The user selects the tutoring session they wish to cancel.

2. The tutoring session is removed from the database.

3. Increases allowed future scheduled logins by one (to a max of three) for the user.

Requirements:

1. The user is authenticated and has an internet connection.

# 4. Non-Functional Requirements 1. Reliability

The system has a few points of failure concerning the data sent and received by the application. First, the student may try to schedule a tutoring session at the same time as another student. In the event of this conflict, priority will be given to the earliest call to the database. If it is determined that the tutoring session has become unavailable, the application should tell the user of conflict and prompt the user to make another selection.

Another point of failure could be the user’s tutoring profile settings could not be loaded from the database. The user information is organized in a relational database. This design should make it simple to retrieve information using simple requests.

## 2. Robustness

If the verification process for scheduling finds a conflict between the requested tutoring session and an existing session in the database, the application should report an error to the user and not overwrite the existing scheduled sessions.

If the verification process for accessing the database fails, the user will be prompted to try again at later point and an email logging the error will be sent to “helpdesk@cs.txstate.edu”.

If the verification process for loading the user profile fails to load in a predetermined amount of time, the user will be logged out and asked to re-login to the tutoring application and also given the option to email “helpdesk.cs.txstate.edu” for further assistance.

## 3. Performance

To reduce potential scheduling conflicts, the application will use AJAX calls to update the calendar information so that users will be able to see the most recent schedule while attempting to schedule their appointment.