SIS Request

## System Design and Requirements Analysis

v. 1.1

10/26/15

**Group D**

Caleb Albertson

David Casper

Ben Friedman

Derek Kedigh

Emeka Okoye

Josh O’Steen

Josh Reeves

Table of Contents

Revision History

Glossary

1. Introduction
2. General Description
3. Specific Requirements
4. System Design

| Date | Version | Description | Author |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## 1.1 Purpose

The purpose of this project is to develop a system for the University of Missouri SIS to manage security requests from departments within the University.

## 1.2 Scope

SIS requires a system that can digitally manage requests by department administrators to add security privileges to new University of Missouri Employees. Administrators will make a request for one of their employees, at which point an SIS employee can process the request. Options for processing the request are “confirm,” “deny,” or “print.” Printing will generate a form that is auto-filled from the database if the employee’s information already exists in the system. The system will eventually be upgrade so that the required signatures can be collected electronically, but such a feature will not be implemented for the first release.

## 1.3 Overview

This document details the requirements and structure of the SIS permission adjustment application. Organization of the document is as follows:

* 1. General Description
  2. Specific Requirements
  3. System Design

# General Description

## 2.1 User Characteristics

The users of the system fall into two categories:

* 1. Administrator
  2. SIS Office Staff

These two categories grant access to two separate different functionalities within the system.

Any undergraduate or graduate student, employee or potential University of Missouri employee is a possible applicant. The system is tied to the University active directory, from which applicant information will be gathered.

Only qualified staff members coordinating the hiring process within a department are allowed to access the system as an Administrator. The account and login information of all administrators are in the database with the required permissions checked. These users can submit security requests for students/employees.

Only qualified SIS office staff can access the system as such. The developers assume that the SIS office staff are qualified to approve or reject security permission requests. They are also part of the University's active directory. This user may also approve Administrator access to the system for any qualified department hiring coordinator or similar staff position.

## 2.2 General Constraints

The system allows administrators to view certain information about their employees/applicants. Measures must be taken to protect sensitive information and prevent access from 3rd parties and employees no longer in position. For this reason, pending requests and filled-out forms should be encrypted while not being viewed by the SIS office.

If a request is approved the information is added to the database. If a request is denied, its associated form is deleted. All generated .pdf files are discarded after the process is finished.

Employees that have had their permissions changed must be updated in a timely manner. Since the database is populated with the University's active directory, consistent maintenance is expected to keep all information up to date

As Group D volunteers their time and manages the appropriate man hours of work on each part of the system, no monetary constraints are worth mentioning.

The system will be hosted on, and operated via the OpenShift server.

## 2.3 Assumptions & Dependencies

The developers assume that OpenShift will be up and available throughout the development process, release, and operation of the system. Should OpenShift become unavailable at any time, the system would also become unavailable to users.

The system requires Internet access, without which no would be able to interact with the system. On the assumption that a user has Internet access, the system shall work on both Google Chrome and Mozilla Firefox browsers; and, any operating system.

# Specific Requirements

## 3.1 External Interfaces

#### 3.1.1 - User Interfaces

The SIS Request System will have two different users who will each have their own respective aspects of the system available to them. After logging in, the user interface will be determined by the functional requirements of the given user. The Department Administrator user will have several views that allow them to: search for a current member of the University of Missouri System Active Directory, request access for an individual in the UM System, and view and edit active requests. The SIS Employee user will have the ability to view active requests made by Department Administrators, deny or approve each request and add a new security form type.

The interfaces shall be formatted in a way such that all functionalities and views of the system are compatible with both Mozilla Firefox and Google Chrome.

#### 3.1.2 - Hardware Interfaces

This application can run on any hardware that is able to support the Internet browsers Mozilla Firefox and Google Chrome, as well as support user input from both a mouse and keyboard used to access either browser. The data saved during the application processes will not be saved to the local disk, but instead to a database that will hold data for all users involved. The user’s computer will use basic network protocols to access information stored in the database.

#### 3.1.3 - Software Interfaces

The following software is required for the system to work properly:

PHP 5.0+

Source: [www.php.net](http://www.php.net)

“PHP is a popular general-purpose scripting language that is especially suited to web development” (php.net). The PHP language and interpreter will be used to process the webpage, and files containing the language will be denoted by the “.php” file extension.

MySQL 5.5+

Source: [www.MySQL.com](http://www.MySQL.com)

MySQL is an open source Relational Database Management System that uses the Structured Query Language. All data store in the SIS Request System Application will be stored in a MySQL relational database.

CodeIgniter 3.0

Source: [www.CodeIgniter.com](http://www.CodeIgniter.com)

“Code Igniter is a powerful PHP framework with a very small footprint, built for developers who need a simple and elegant toolkit to create full-featured web applications” (CodeIgniter.com). The system uses CodeIgniter, which provides a MVC framework for web application development using PHP.

#### 3.1.4 - Communication Interfaces

All data that is accessed by the user shall be transferred over the network using the HTTP network protocol to ensure security and stability in the MySQL database. The data in the database will not be changed directly by the user, but instead will be transferred through the web application to be used in an effective and functional manner.

## 3.2 Functional Requirements

#### 3.2.1 - Login

For a user to login to the application they must use their Employee ID and the password assigned to them when the SIS office established their account. Once the user has successfully logged into the system, the application will indicate to the user that they have successfully logged in. In addition, the application will also inform the user of any login errors. Once the user has successfully logged into the application, it will then divert the user to the home page where they may interact with the application.

#### 3.2.2 - Forgot Username or Password

In the event that a user forgets their username and/or password, that user can can go to a recovery link from the login page. This page will require the user to input various unique identification information that will allow the system to recover such a user’s login information. Once verified by the system, instructions on how to reset the password will be sent to the email on record. Once the password has been reset, another email will be sent to confirm the change and for the user’s personal records.

#### 3.2.3 - Reset Password

If the user wishes to reset their password, they must request to do so through the forgotten password link described in section 3.2.3 or under their account settings once logged in. In the case of the latter, the user must first enter their current password, followed by their new password and a confirmation field. Once the reset password has been accepted and recorded in the system, a confirmation email will be sent to the user for their records.

#### 3.2.4 - Create Request

The system will provide a link to allow administrators to create an application to request security access to a chosen system for one of their employees. The administrator will be given the option to select which systems the employee will need access to in order to limit the size of the displayed application.

#### 3.2.5 - View Request

The administrator user will be able to view applications which they have created. When presented, the application will show whether or not the application is still pending or if it has been closed. If the application is closed the administrator will not be able to edit it.

The SIS employee user will be able to view any application which they have searched for in the system. The SIS employee will be able to select an application from the list to review the security requests. If an application is listed as “pending” they will be able to close it if necessary.

#### 3.2.6 - Print Request

The administrator user will be able to, with one button, print out a form with all chose security request sections pre-filled with the input from creating the application. This will also save the application to the database with a “pending” status. The user will be given the option to attempt to save again if the previous save attempt fails.

#### 3.2.7 - Review Request

After the user has filled out the security access form, the user should click the “Review” button to be taken to the next page to Review their completed request form before submitting and generating the pdf file. The review request page will display a clean and formal version of the selections that the user made on the previous page. If the user feels that they made a mistake or forgot to fill out an aspect of the request form, the user can utilize the “edit request” button. This button will send the user back to the previous page so that they may make changes.

## 3.3 Non-Functional Requirements

#### 3.3.1 - Ease of Use

The system shall be easy to operate. An administrator should be able to do any and all of the system use cases without any formal training.

#### 3.3.2 - General Requirements

Administrators, acting as hiring managers or similar position within their respective department, should have access to the University’s active directory in order to search for students or employees.

This system should be hosted on University owned servers and should fall under the management and maintenance of the University’s Department of Information Technology as such.

#### 3.3.3 - Administrator Requirements

Administrators should not be able to create, edit, or view requests for their own system identities. In order to do such, a colleague within the department with positional superiority should make the request through the system or make a direct request with SIS.

#### 3.3.4 - SIS Employee Requirements

SIS employees should only be able to view, edit, and approve pending requests. The ability to create requests is reserved for administrators.

#### 3.3.5 - Performance

The system should permit concurrent access to any number of users at any time. Any reduction in performance should be handled by hardware upgrades performed by the Department of Information Technology as well as system upgrades by the developers appropriately.

#### 3.3.6 - Reliability

The system should be available at anytime once deployed unless necessary system upgrades, whether hardware or software, were to occur in a fashion that prohibits the system to operate while such upgrades occur.

#### 3.3.7 - Security

For purposes of maintaining user privacy and securely transferring user information across this system and its dependencies, the system will rely on the HTTPS protocol to handle such sensitive data transfers. All web pages should be presented using this protocol and in the event a user attempts to load a page through the HTTP protocol, intentional aside, the system shall redirect to a secure version of the page.

#### 3.3.8 - Legal Requirements

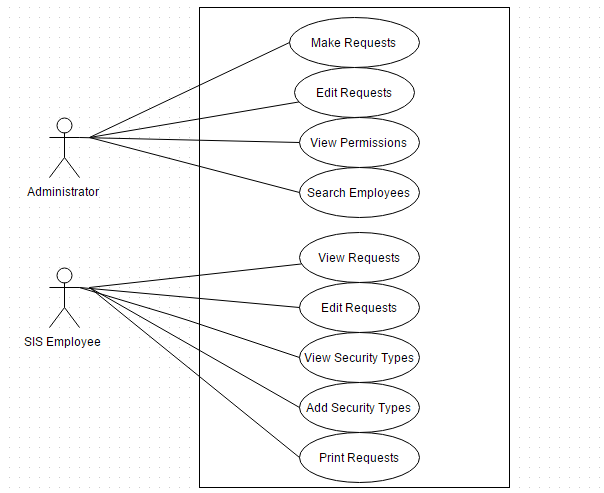
Given the nature of the data available about users, this system and all its users shall abide by the regulations provisioned concerning personal information about University students and employees in accordance with FERPA and state laws.

# System Design

## 4.1 Database ERD

This Entity Relationship Diagram depicts the SIS Request System’s entities and relationships using Crow’s Foot Notation. Each entity is made up of several attributes that describe and indirectly define its purpose. Also shown in the diagram are the types of relationships between entities that include: many-to-many, one-to one optional, and one-to-many relationships. Included in the diagram are the individual types of security access entities, which have a one-to-one optional relationship to the security access entity. Also, there is a weak entity that relates the Person and Request entities using foreign key attributes from each respective entity.

## 4.2 Use Cases



Shown above is the use case diagram for the system. As you can see each actor has different roles while interacting with the system. The diagram shows the respective accessibility to the available features in both cases.

## 4.3 Activity Diagram

#### 4.3.1 - User Login

#### 4.3.2 - User Request