SpartaDocs

Created by MASSware

CS 160 Section 2 Project

Team: Frank Daniels, Malik Khalil, Arjun Nayak, Sajay Shah, Shweta Sugnani

Prof. Ahmad Yazdankhah

Department of Computer Science

San Jose State University

Fall 2016

## Table of Contents

### 1. Introduction 3

#### 1.1 Overview 3

#### 1.2 References 3

#### 1.3 Acronyms and Abbreviations 3

### 2. System Overview 3

#### 2.1 Problem Statement 3

#### 2.2 Scope of the System 3

#### 2.3 Technological Requirements 4

### 3. System Architecture 5

### 4. Detailed Design 6

#### 4.1 Use-case UML diagrams 6

#### 4.2 Class UML diagrams 11

#### 4.3 Sequence UML diagrams 12

#### 4.4 State UML diagram 13

#### 4.5 Database architecture 15

#### 4.6 User Interface 16

## 

## Table of Figures

### System Architecture Diagram 5

### Sign Up Use Case 6

### Log In Use Case 7

### Hold Release Form Use Case 8

### Graduation Form Use Case 9

### Log Out Use Case 10

### Class Diagram 11

### Sequence Diagram 12

### Document State Diagram 13

### User State Diagram 14

### ER Diagram 15

### Dashboard & User Menu 16

### Hold Release Form 16

### Hold Release Successful 16

### Graduation Form 18

### Graduation Form Successful 18

# Introduction

## Objective

This document describes the technical specifications of our project. It serves as a reference and guide to the more technical aspect detailed in the Functional Spec. Detailing functionality allows for a more cohesive understanding of the software and how it interacts internally.

The intended audience for this document are the project manager, developers, and client.

## References

N/A

## Acronyms and Abbreviations

N/A

# System Overview

## Problem Statement

Currently, obtaining advisor verification for school documents can be very problematic as office hours can be during classes and modifications create wasted papers. In the 21st century, such an antique system needs to be updated for modern systems.

## Scope of the System

This system should be able to a large amount of users and administrators access different parts of the application. The system should have adequate resources to handle large amounts of requests from students and faculty for different internal resources. The system should be able to handle a version control system for multiple documents per student.

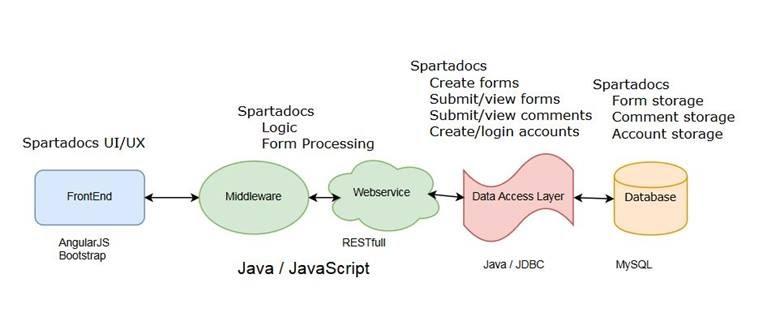
## Technological Requirements

For our project, we use various technologies to create our application. To service requests, we use Apache’s Tomcat server. This combines Apache’s ability to service HTTP web pages and Tomcat’s ability to create dynamic web pages using Java. Our web pages are created by using HTML and CSS to stylize the pages. Information is then inserted into the pages using AngularJS. For our backend, we have a RESTful architecture build using Java to pass information to Angular modules. RESTful uses Java classes to fetch data from a MySQL database that contains all student documents and user information.

In order to deploy our application, a specific environment needs to be set up. Two main programs are needed: Apache Tomcat to send WAR (Web application ARchive) files which contain all files necessary to create dynamic web pages; and a MySQL server to handle all information needed by Tomcat.

Users need to have a stable internet connection and a computer capable of computing JavaScript functions. Users have no restriction on what browser the user needs to use however, it is recommended to use the latest versions to reduce any security exploits created by using an out-of-date version.

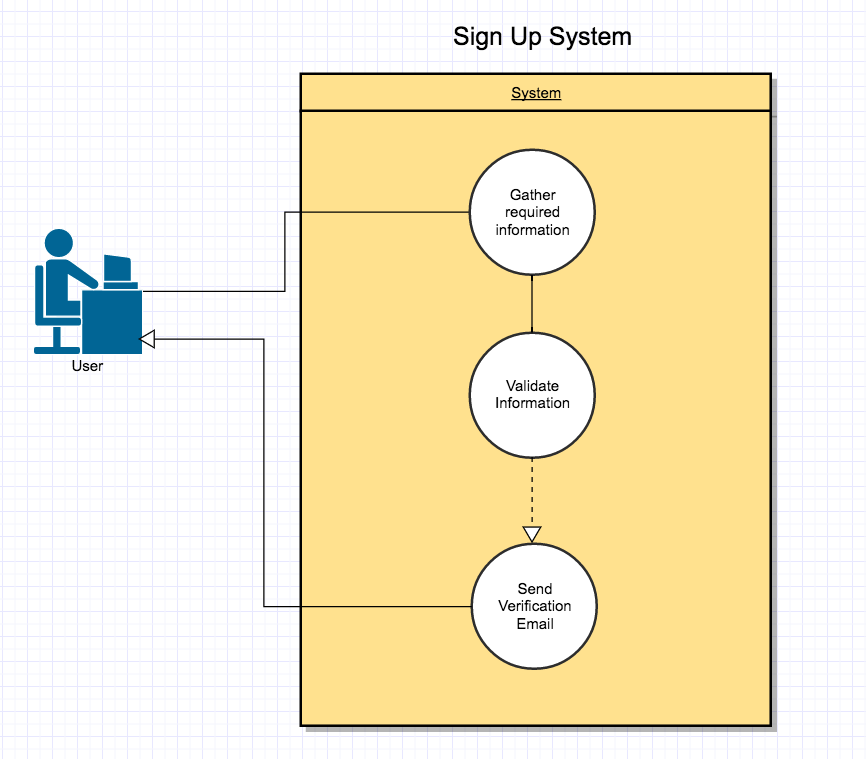
# System Architecture



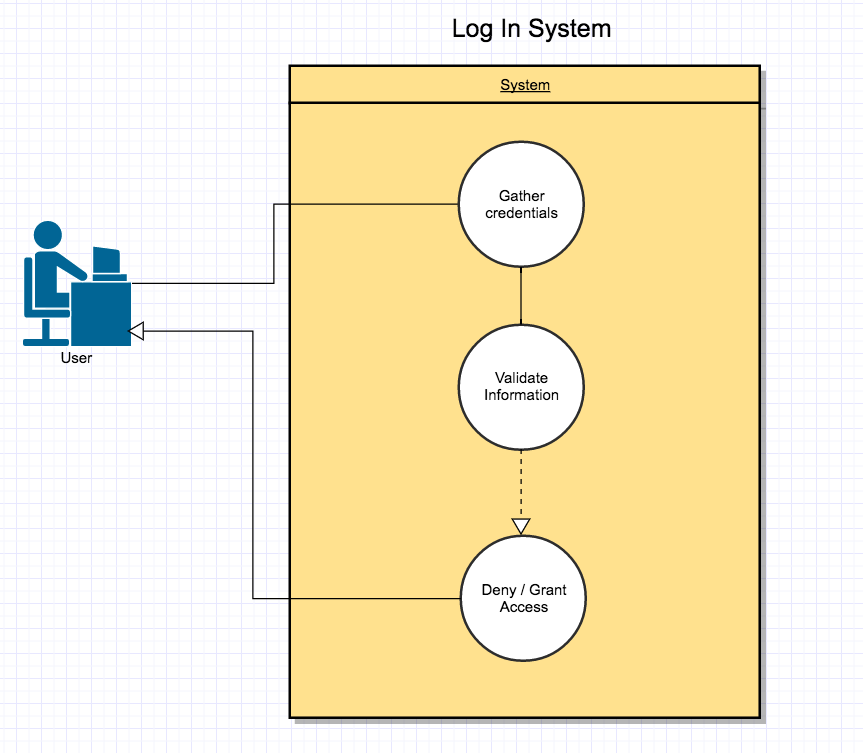
# Detailed Design

## Use-case UML diagrams

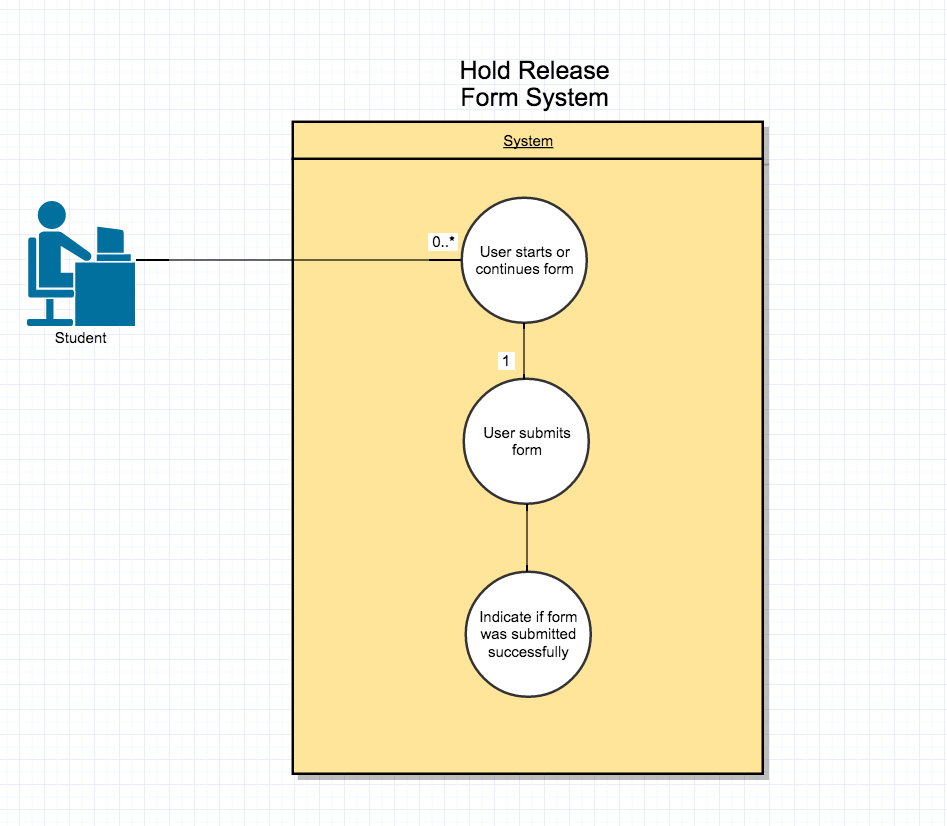
1. Sign Up Use Case UML



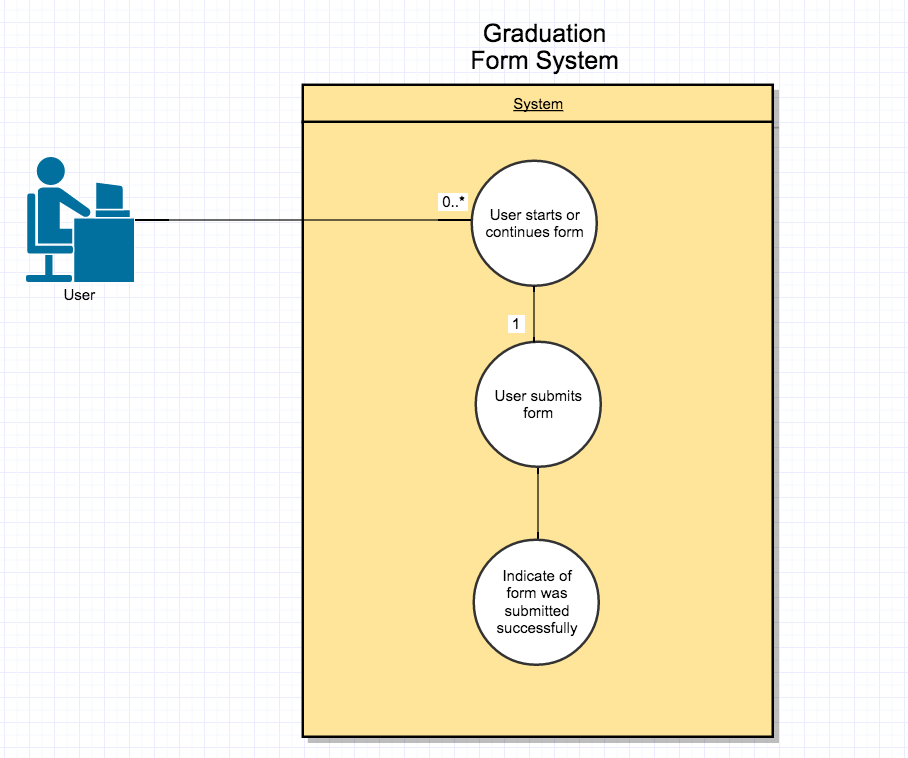
1. Log In Use Case UML



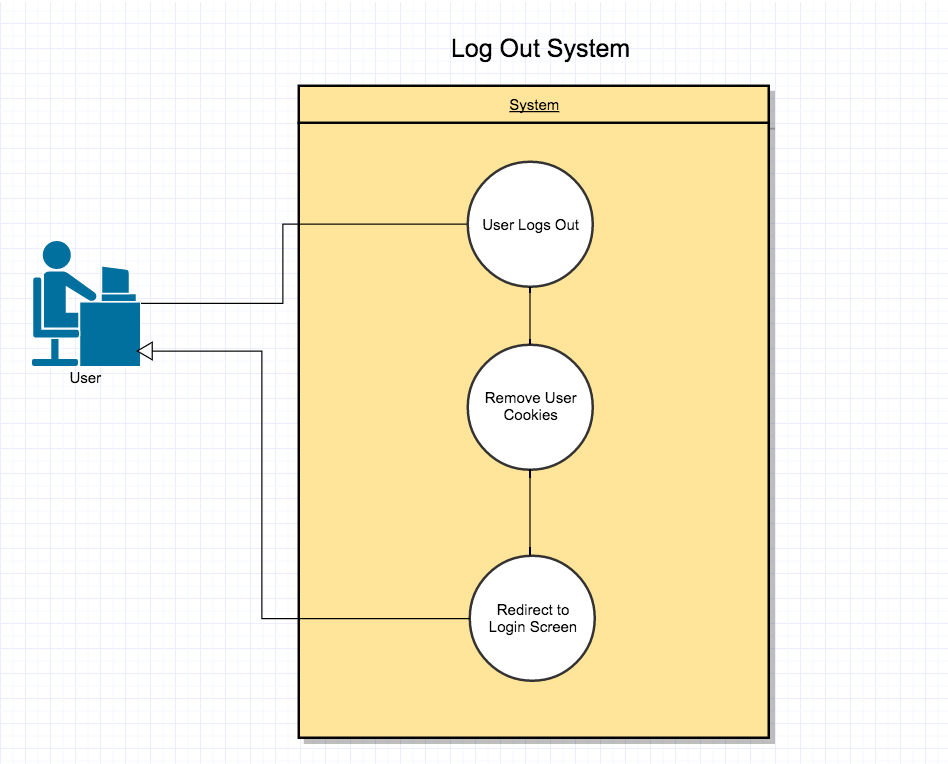
1. Hold Release Form Use Case UML



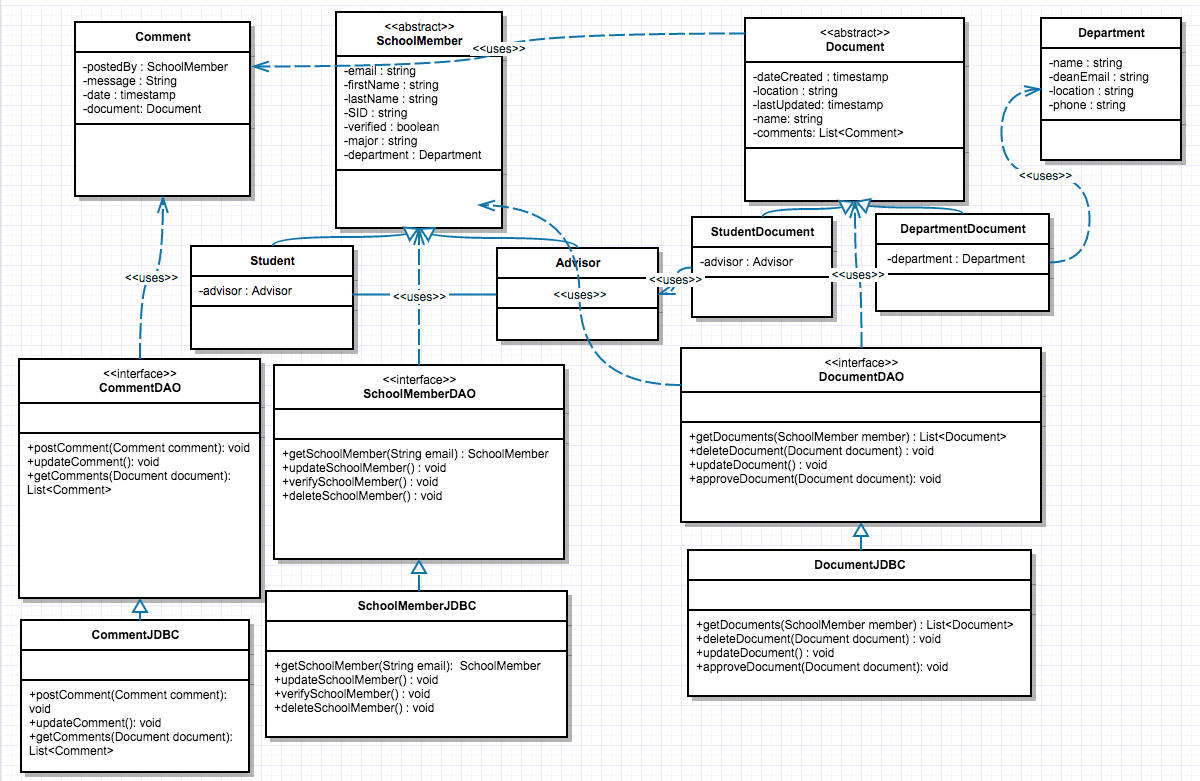
1. Graduation Form Use Case UML



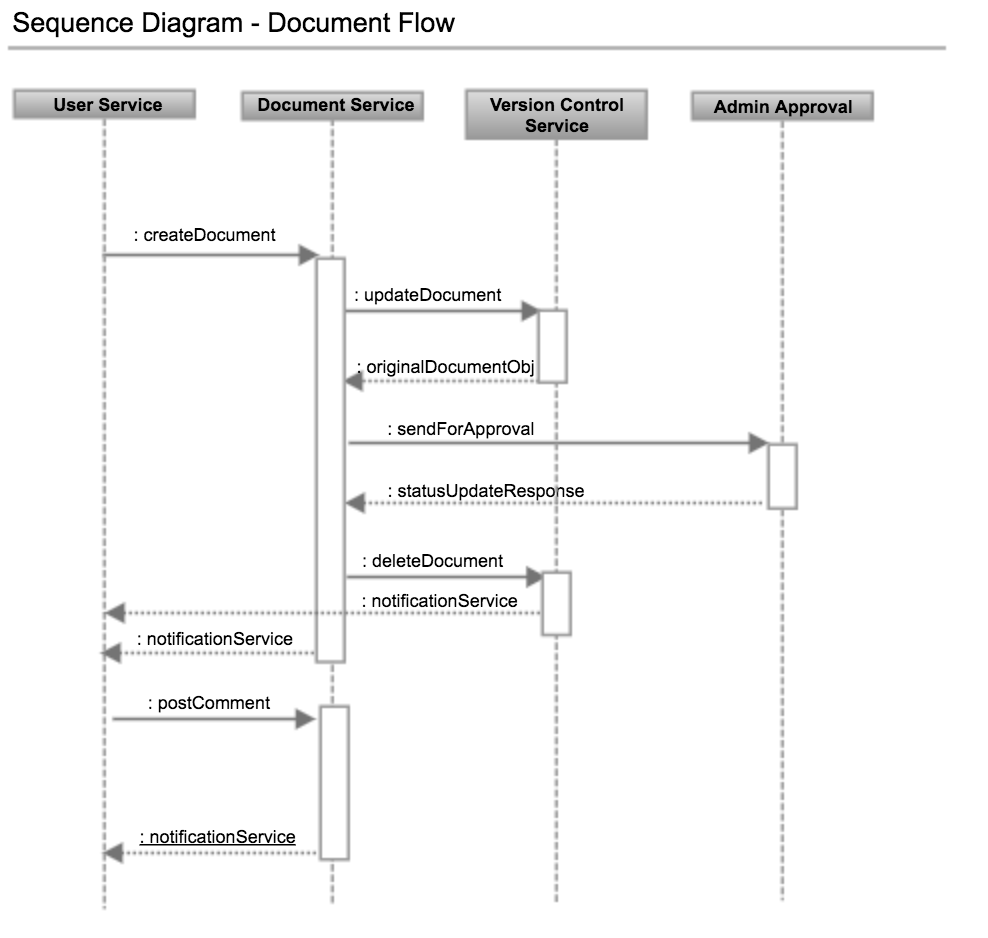
1. Log Out Use Case UML



## Class UML diagrams

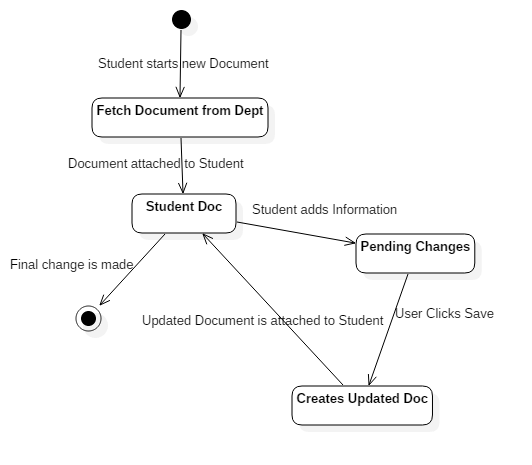


## Sequence Diagrams

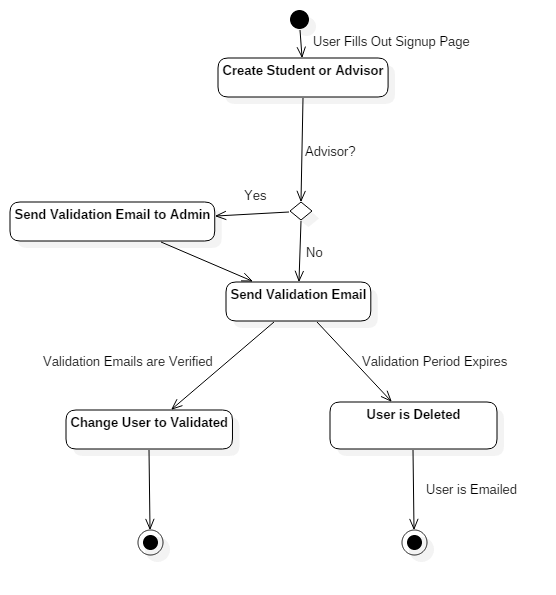


## State UML Diagrams

### Document State Diagram

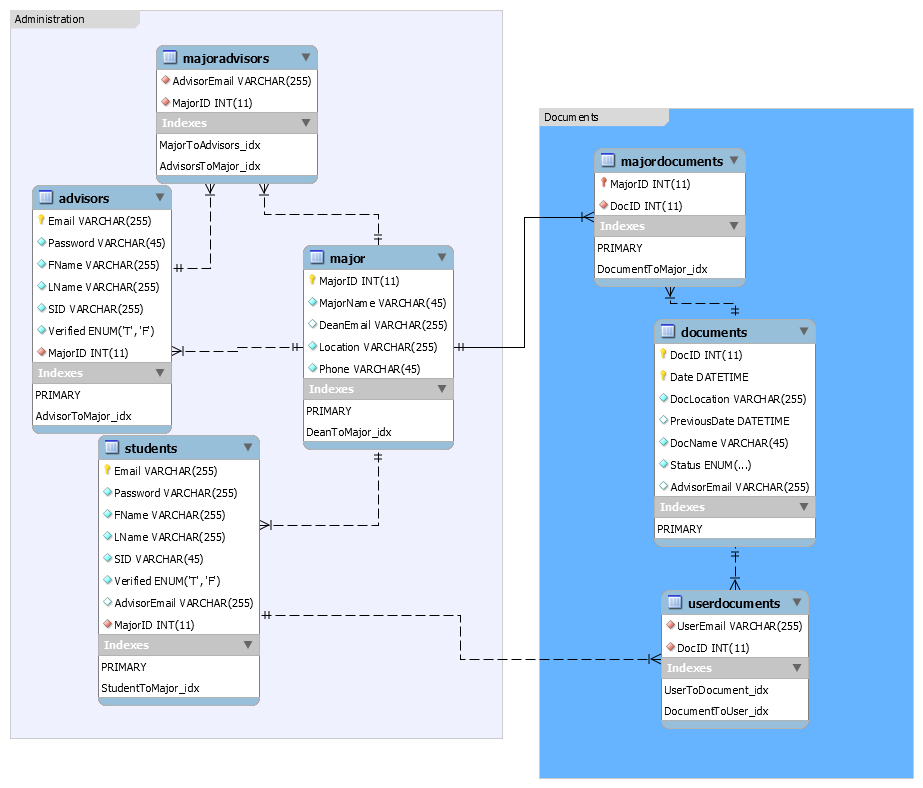


### User State Diagram



## Database Architecture

* + 1. ER Diagram

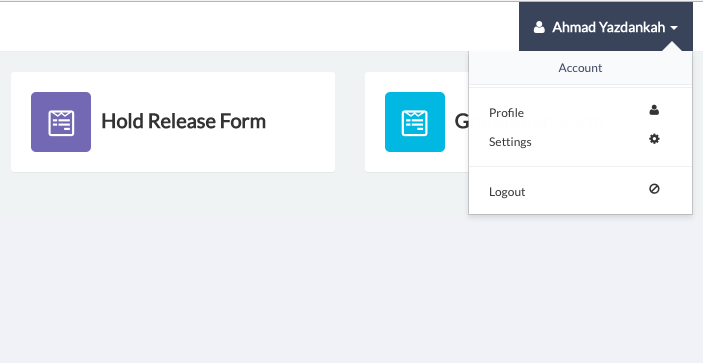


## User Interface

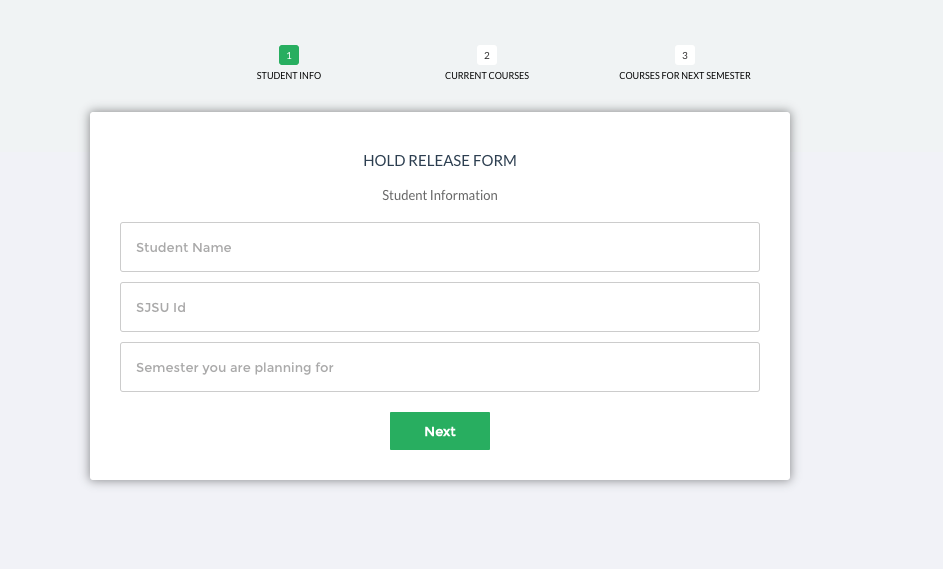
4.6.1. Dashboard w/ side menu bar

## 

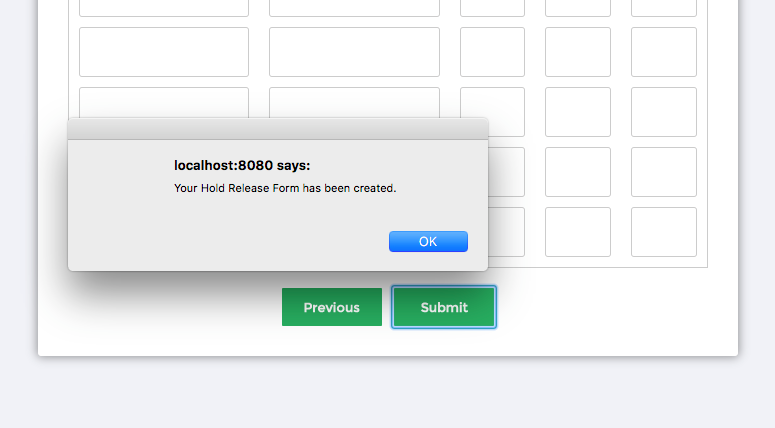
4.6.2 User Menu



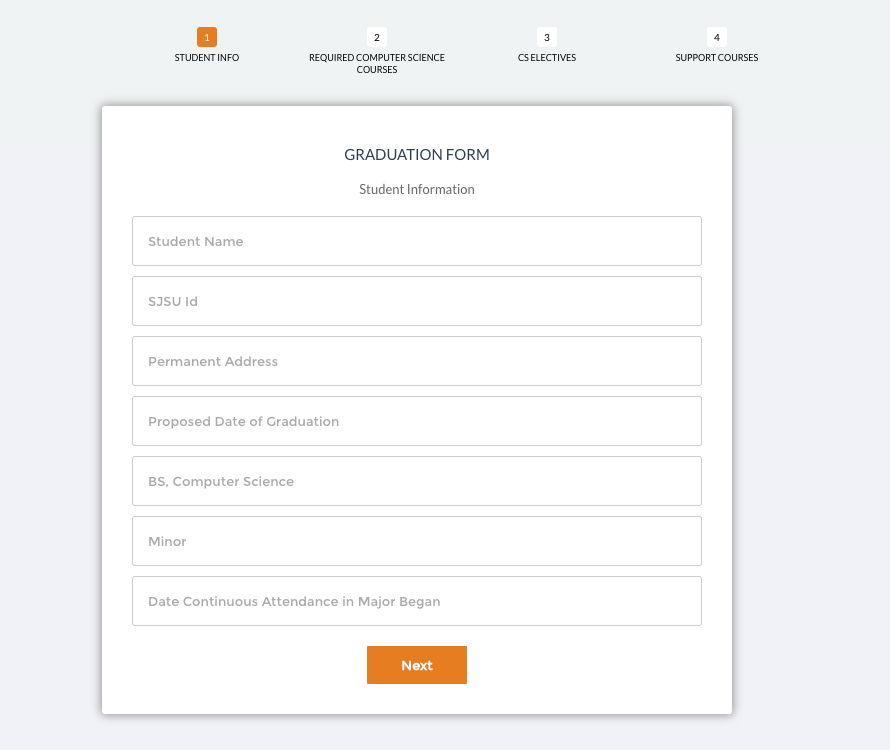
4.6.3 Hold Release Form



4.6.4 Hold Release Submit Success



4.6.5 Graduation Form



4.6.6. Grad Form Submit Success

