System Design Document (SDD)

Project: Student Information Management System

Purpose of the document

This document mainly discusses the system design aspect of our student information management system. Design goals will be provided in the introduction of the document to identify the qualities that our system will focus on. An overview of the current system architecture will be included for comparison with the proposed system architecture. Besides that, the proposed system architecture, its subsystem decomposition, hardware and software mapping, persistent data management, access control and security, global software control , and subsystem services will also be included in this document.

Ultimately, the goal of this system design document is to provide design specification of SIMS to facilitate project implementation or development process.

About the System Design Document

Version Control

Version No.	Name (Title)	Author	Date	Authorization
1.0	SDD Report	Zhi Kang Wong Fan Wu Gabriel Alvarez	1/23/2019	

Document Properties

Items	**Details**
Document Title	ZFG System Design Document
Author	Zhi Kang Wong Fan Wu Gabriel Alvarez
Creation Date	1/23/2019
Last Updated	

Table of Contents

1. Introduction

1.1 Purpose of the system (Gab)

The purpose of this system is to provide an all-in-one system that will handle student related information, schedule courses, and handle exit surveys whilst maintaining an user friendly interface. Users should be able to update the information given the correct authorization. Additional features include adding, deleting, sorting, searching, updating. Each user will have a role which consists of graduate program director, undergraduate program director, faculty, thesis advisor, project advisor, and student. Depending upon the role/authorization, each user will have access to certain features.

1.2 Design Goals (ZK)

Performance:

In terms of response time, our Student Information Management System (SIMS) should be able to handle all requests from users such as logging in, searching and updating information, etc. within seconds. The throughput of the system will be high with the help of load balancing. The ideal case will be the system can accomplishes up to 100 tasks in a fixed period of time. Since it is a web application, no much memory will be required for the system to run.

Dependability:

The robustness of SIMS will be strong as it could survive invalid user input such as displaying notification or warning messages to users when their credentials are wrong during the login process. SIMS will also be reliable as it will deliver results as requested by users instead of producing unrelated output. The availability of this system will be 24/7 to allow users use it anytime. There will be implementation of encryption on SIMS database to protect user's private information from malicious attacks. The system will be safe and will not endanger human lives.

Cost:

The development cost for this system is $10. Subsequent cost such as deployment, upgrade, maintenance, and administration cost will be covered by our stakeholders SIU CS department staff.

Maintenance:

SIMS will be extensible to allow the addition of new functionalities to the system besides the current features. The system will also be easy to modify as long as the developer(s) has the knowledge of Angular 6 JS, ASP.NET Core, etc. In terms of portability, our system should be able to run on any platform such as computer and mobile devices with the presence of internet. Readability for this system will be maintained with the addition of comment in system code. Ultimately, SIMS should implement all functional requirements listed in the requirement analysis document to facilitate traceability of requirements.

End User Criteria:

The proposed SIMS contains functions that are requested by the stakeholders. To increase the user's utility and usability to our system, SIMS will be having user interface that are user friendly, reactive, and scalable.

1.3 Definitions, acronyms, abbreviations (Fan)

- SIMS: Student Information Management System
- SIU: Southern Illinois University
- CS: Computer Science
- MVC: Model- View- Controller
- SDD: System Design Document
- RAD: Requirement Analysis Document
- TCP/IP: Transfer Control Protocol/Internet Protocol
- DBMS: Database Management System
- API: Application Programming Interface

1.4 References (Fan)

Before reading this system design document (SDD), it is better to read the prerequisite document, Requirement Analysis Document (RAD), which gives all the functional requirements and nonfunctional requirements of this system. For some relevant technics, the main reference is an online course from Udemy.com. In addition, if more detailed information is needed, some online resources can be referred. The following is a table of references.

Document Name	Document Location or URL	Date
Requirement Analysis Document (RAD)	Project Deliverable	1/20/2019
Build an app with ASPNET Core and Angular from scratch	https://www.udemy.com/build-an-app-with-aspnet-core-and-angular-from-scratch/learn/v4/content	11/16/2018

1.5 Overview (Gab)

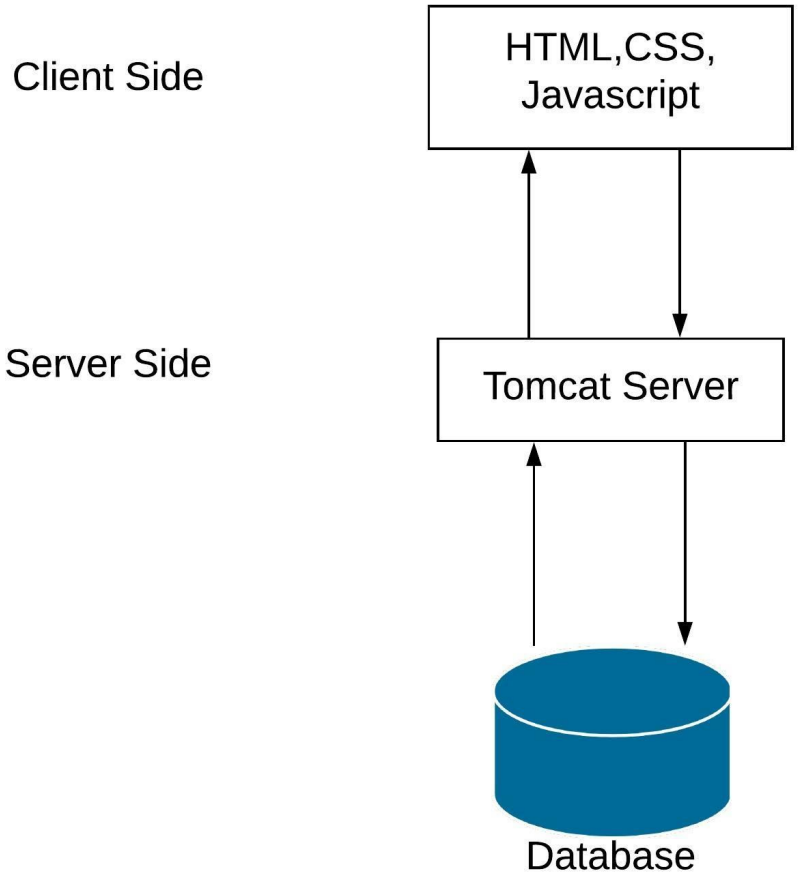
To help with design, this document was constructed to facilitate comprehension of the Student Information Management System (SIMS) and it's inner workings. Hopefully, this documents will aid development of the system and provide a well structured blueprint.

In this document we will break down the many aspects of the Student Information System. First we will delve into the current software architecture, to have a better understanding of the next topic which will focus on the proposed software architecture. We will decompose the system into subsystems, then we will look at hardware/software mapping, data management, access control and security, global software control, and boundary conditions. Finally we will discuss subsystem services provided by SIMS.

2. Current Software Architecture

2.1 Overview (ZK)

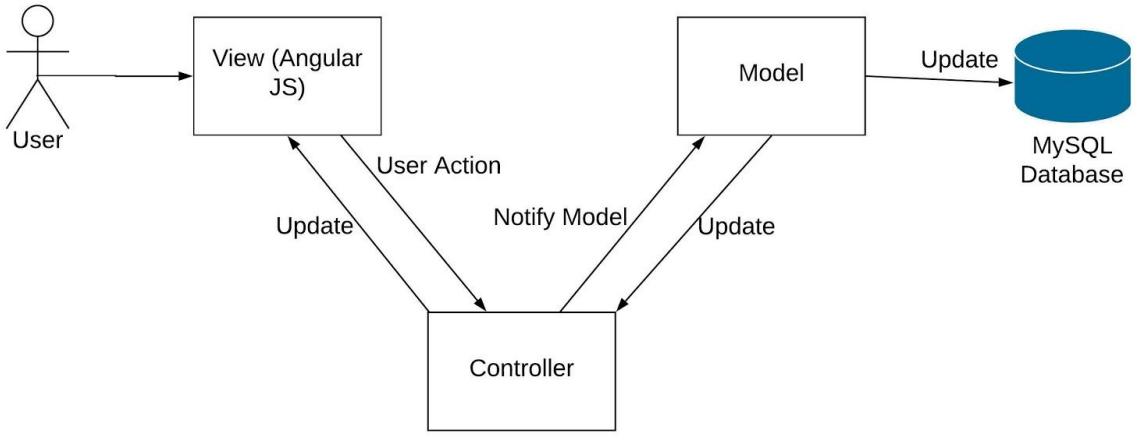
In our requirement analysis document, the exit survey system and course scheduling system are no longer being used due to the lack of maintenance with the absence of previous developer. Hence, the software architecture of the current system are not known extensively. Below is a tentative three-tier software architecture diagram for the existing student exit survey system.

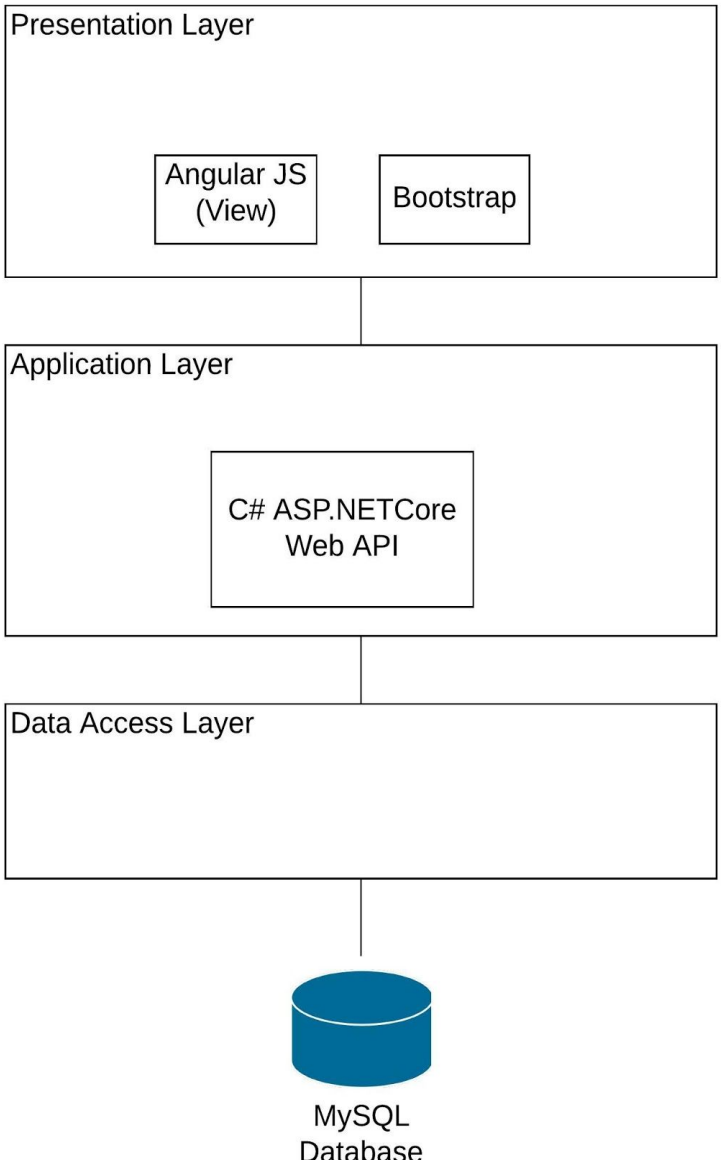


3. Proposed Software Architecture

3.1 Overview (ZK)

Our proposed system make use of frameworks and APIs such as Angular JS, ASP.Net Core, Entity Framework, RESTful API, and ASP. NET Web API. Since our system is a web application, we will be using MVC (Model View Controller) architecture. However, the view engine from ASP. NET Core (razor) will not be implemented, instead we will be using Angular for the client side templates (or views). In addition, SIMS will be consisting of several subsystems that carry out different functionalities. The subsystems of SIMS include login subsystem, logout subsystem, registration subsystem, delete user subsystem, role assignment subsystem, student exit survey subsystem, update thesis / project subsystem, course scheduling subsystem, searching subsystem, sorting subsystem, and edit user profile subsystem.



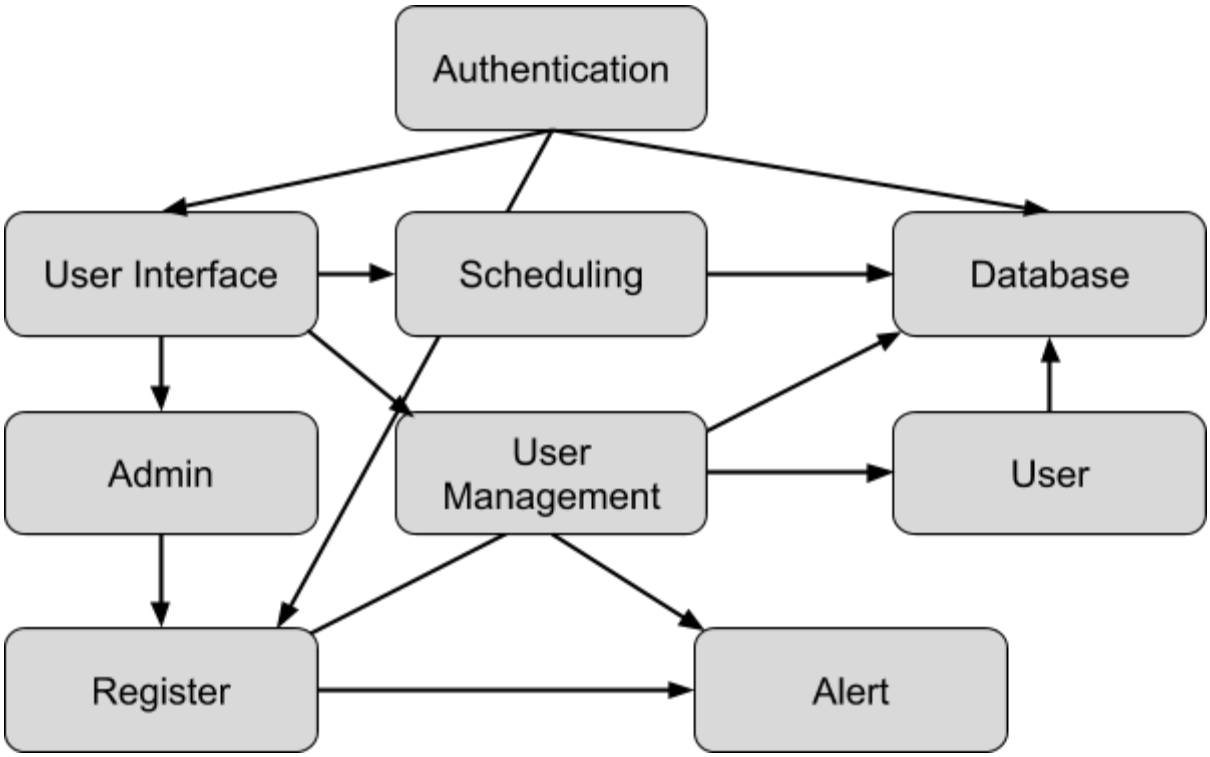


Physical Software Architecture

ZFG SIMS is a client-server system following the 3-tier architecture. The first layer is the presentation layer, which represents the user interface and the frontend of the system. This layer will be built on web technologies like Angular 6 JS, Bootstrap, etc and communicates with subsequent layers through API calls The purpose of this layer is to provides user interface for users and allow them to interact with the system. The application layer can be considered as the

controller of the system as it contains the subsystems and its services. This layer will receive user actions from the presentation layer and processes them accordingly. The data access layer contains DBMS for data storage.

3.2 Subsystem decomposition (Gab)



Authentication

The main function of the Authentication subsystem is to authenticate user login before allowing access to the system. Authentication is needed in order to use the system. Requires services by the Database subsystem.

User Interface

The User Interface subsystem will provide each user with a predefined interface depending on their respective roles within the system. From here the users will have access to the system's

functions. Authentication services are required in order to login and display the interface. Provides communication between the users the rest of the subsystems.

Admin

The Admin subsystem provides services to those that are authorized such as adding, deleting, searching the database. The intended function of the admin is to register new users into the system. Authorization is required.

Register

The Register subsystem is responsible for checking credential and registering users. Uses the Database subsystem and is intended for use with admin privileges.

Scheduling

The Scheduling subsystem will take care of course scheduling. Time slots, rooms, professors, availability, and class info will all be handled here. Will need access to the Database subsystem.

User Management

One of the biggest subsystems. It will be responsible for managing user related info. The subsystem will provide services such as editing profile information, adding/removing files, adding/removing pictures.

User

The User subsystem will take care of fetching individual or multiple users from the database as requested by the users. Needs services from the Database subsystem.

Database

The Database subsystem is responsible for storage and retrieval of all persistent data in the system. Other subsystems are reliant on the Database subsystem service for completing queries.

Alert

The Alert subsystem provides other subsystem the means of displaying alerts to the user when something has gone wrong, missing items, invalid request, or they have done something illegal.

3.3 Hardware/ Software Mapping (ZK)

Hardware configuration. Off-the-shelf components - Chrome, Firefox Browser. Do we use SIU server machine before and after deployment?

3.4 Persistent Data Management (Fan)

For persistent data management, a relational database created by MySQL will be used as the database management system (DBMS) of the Student Information Management System (SIMS). This database is used to store all data related to the SIMS, including user information data, course scheduling data, student exit survey data, and thesis/project data. SIMS shall implement some authentication mechanism for the requests of the access to the database.

Mapping

The following figure shows the relational database schema of the SIMS.

3.5 Access Control and Security (Gab)

Access control and security will be enforced with required logins, JSON tokens, and data encryption.

Actor	Privileges
Admin	● Register users ● Delete users ● Assign roles ● Manage student information ● Manage exit surveys ● Update students' thesis/projects

	● Schedule courses ● Search students ● Edit profile
Staff	● Delete users ● Assign roles to user ● Manage student information ● Manage exit surveys ● Update students' thesis/projects ● Schedule courses ● Search students ● Edit profile
Thesis/Project Advisor	● Update students' thesis/project ● Search and sort students ● Edit profile
Faculty	● Manage exit surveys ● Search and sort students ● Edit profile
Student	● Manage exit surveys ● Edit profile

Required Login

In order for an user to gain access and utilize the system they must first login to the system using valid credentials

JSON Tokens

After successful login, the system will generate a token for the user that will be used to indicate the identity of the current user, his/her roles, and authorizations.

Data Encryption

Delicate data will be encrypted using SHA512 hashing.

3.6 Global Software Control (Fan)

4. Subsystem Services

RESTful API, web api, talk about subsystem in details.