Design Document

CSE 361 - Spring 2018

Group 1

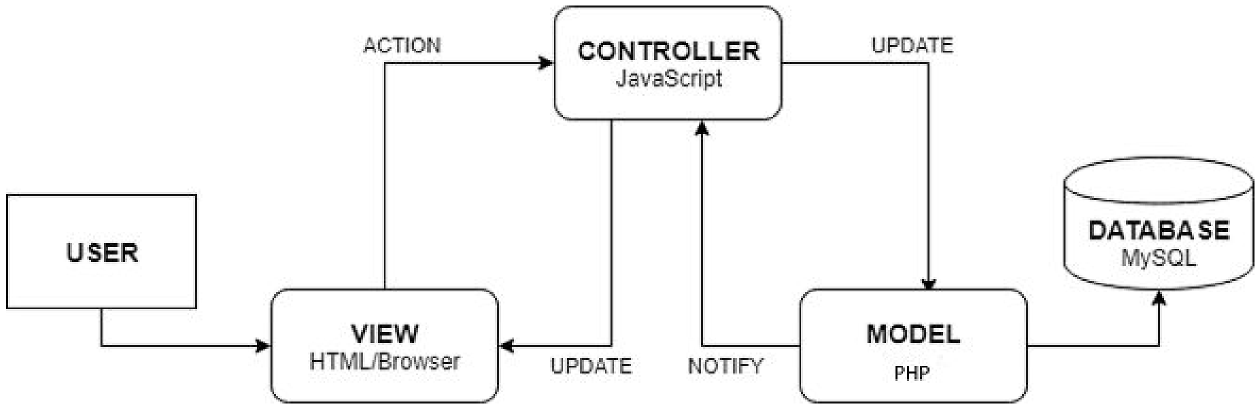
Alex Clough, Christian Farmer, Addison Higley, Alejandro Pages, Matt Shanahan

1. Introduction

The purpose of this document is to demonstrate the high level architecture and entity relations for the Degree Planner system. This document will include architecture and entity relation diagrams showing the relationship between different parts of the system, as well as the relationships between tables in the database. The audience of this document is software engineers and system architects who will be implementing and maintaining the described project.

2. Architecture

2.1 Introduction



The high level architecture design of the system will use a model view controller design. PHP will be used to model the business logic of the system and PHP will also be used as the controller to control interactions between the user and the model and between the model and the view. HTML will be used for the view of the project. The PHP model will interact with the database which stores data relevant to the system.

2.2 Modules

2.2.1 View

This is the primary module that the user will interact with and is responsible for generating the user interface. It is the module that the user will be able to login to the system and access their degree plans and planning board.

A combination of HTML, CSS, and Javascript will be used in this module to dynamically generate user web pages from the data and allow for seamless user interaction. Users will be able to interact with the view, via the controller, to create degree plans on their planning board. Data will be called from the database as users search for and plan classes. Data will be stored in the database to represent the plan created by the user so that it can be recalled each time the user logs in. In future phases, it will be possible for the user to export their planning board and degree plan to a PDF.

2.2.2 Controller

This is the module that will take user input and relay said input to the model of the system. Output from the model will then be taken in by the controller and converted so that it can be displayed via the view module.

2.2.3 Model

The module for this system will be responsible for database management. When the controller submits an instruction to the model, a PHP script will be responsible for interpreting the instruction and querying the database to retrieve information and return it to the controller. In Phase II of implementation, the controller will also allow the user to create a user account, the model will be responsible for storing and verifying user credentials. The purpose of the user account will be to store progress with their course plan, when the user logs in, the model will start a user session and upload all their information from the database to the controller. As soon as the user saves their progress, the state of their course plan will be submitted to the model to store their progress overwriting their previous state.

2.2.4 Database

This module is responsible for holding all the persistent data for the system and for defining the relationships between this data. The system will be using a MySql database that can be easily incorporated into the project. The data relationships are described in more detail below in 3.1.1 where the data table schema is displayed.

3. Class Diagrams

3.1 Data Table Classes

The system will be using a MySql database to hold all the persistent data necessary. This information includes information on users, courses, majors, minors and the relationships between these entities. The figure below shows each table in the database with its respective columns as well as the relationships between the different tables.

3.1.1 Schema

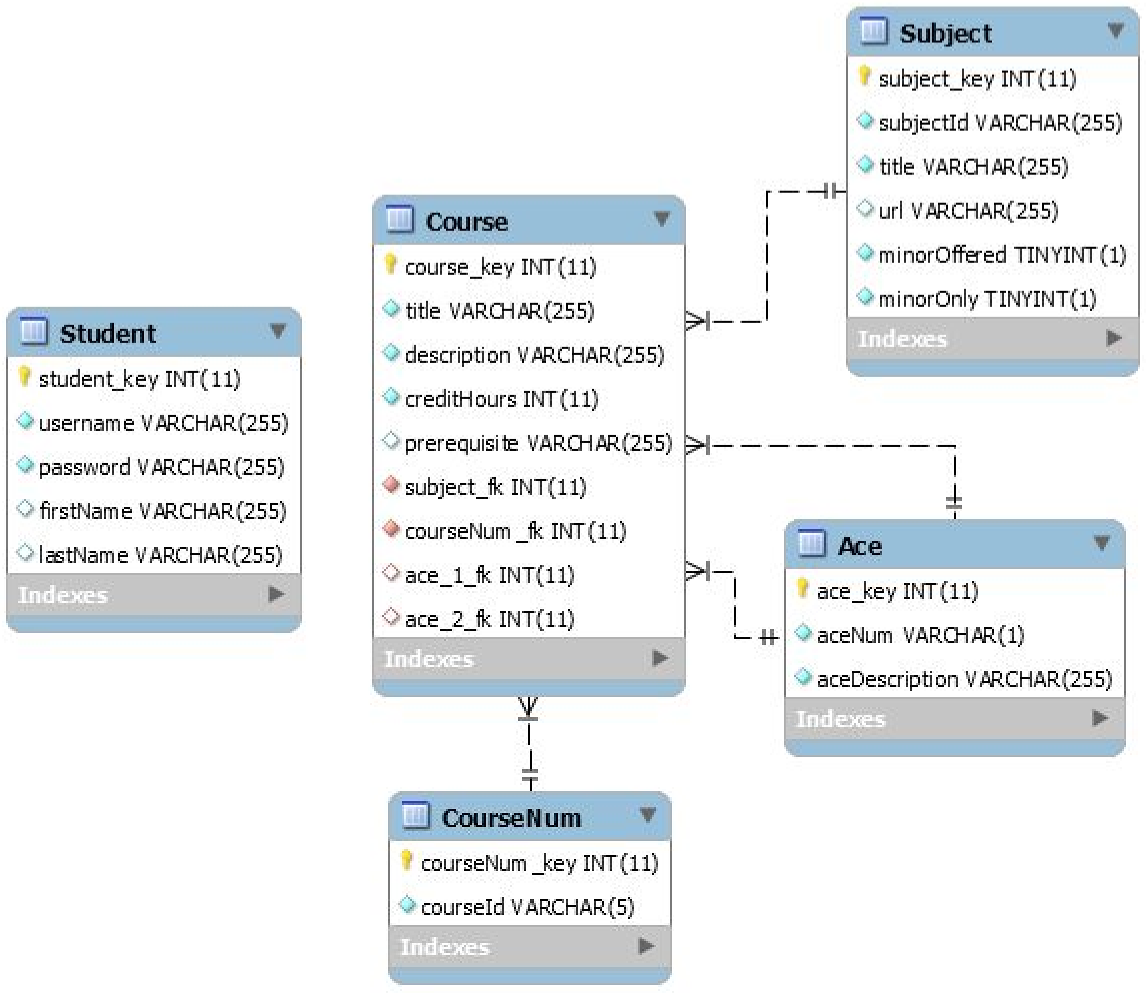


Figure 3.1.1: Course Planner database schema storing course information and student user information.

3.1.2 Schema Information
The data in the database shall be organized in the following tables and columns.

Subject:
The subject table will include information about each subject offered at UNL including the subject id and title such as ‘ECON’ for Economics or ‘CSCE’ for Computer Science and Engineering. The database will also include a link to the subject’s entry in the UNL course catalog for users to conveniently view subject information such as graduation requirements while making their plan. Finally, the subject table will also include two fields specifying whether a minor is offered and whether there is only a minor available. This way when the student wishes to select a major, the app will issue a query to the database excluding fields where the boolean field ‘minorOnly’ is true. When the user wishes to select a minor, the app will query the database excluding all information where the field ‘minorOffered’ is not true.

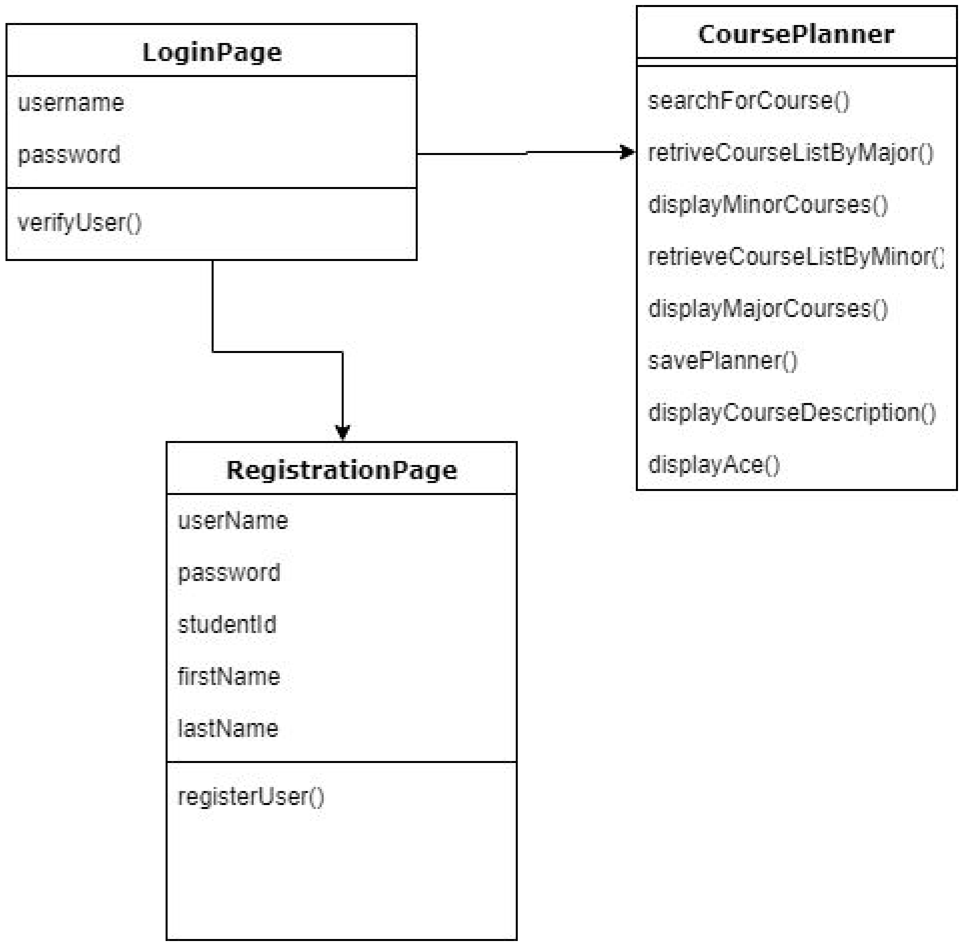
Ace:
The ace table will only have 9 entries corresponding to the Ace requirements at UNL. The ace table will include information about the Ace number and its corresponding ace description.

CourseNum:
This table includes all the course numbers for each subject offered at UNL. This table is included to enforce normalization of the database. Course numbers are repeated for each subject and are intended to give an idea of the advancement in the subject. Course numbers such as 101 are usually freshmen level introductory courses while 400 level courses are advanced level course intended for Senior students. Each subject such as CHEM (chemistry) or PHYS (physics) may both have a 101 course number.

Course:
The course table stores a grouping of course information with foreign keys referencing subject, course number and ace satisfaction. There are two fields for ace for classes that can satisfy multiple ace requirements. The course table includes information such as course title, description, credit hours and prerequisite description string.

Student:
The student table stores information about users who wish to create a login to save their progress between sessions. The database will store a username and password for the students along with optional first and last name.

3.2 Class Information
Classes will be implemented in PHP in the model module. These classes will replicate the database schema above with each table corresponding to a different PHP class. The model will relay this information to the controller which will in turn convert the data and display it via the view.



3.3 GUI

The front-end of the system consists of only three pages. A main login page, a registration page, and a planning board page. Subwindows may appear on the planning board page but functionally it will be a single page. When a user first navigates to Degree Planner the login screen is presented where a user will enter their username and password. In the case of a new user the user will be redirected to a registration page where they can set up a username and password. For registered users, the information is verified against the database and if invalid an appropriate error is displayed to the user. If the information is valid the user's planning board page is loaded.

The planning board page is much more complex and dynamic. It will store data that the user who is logged in has selected for the degree plan. Initially the planning board will be plan and will prompt the user to select a Major and if applicable, a minor as well. Searching for classes will allow them to be added to planning board. Alternatively the user can select relevant courses from a displayed list of degree requirements that varies depending on selected major. Selecting a course will display additional information about the class such as the course description, credit hours, and so on.