**Software System Design**

**Document Specification**

By Quintus J Lin

Liang Li

Yingfa He

Fall 2016

Table of Contents

Table of Contents 2

Table of Figures 2

1. Introduction 3

3. Required System Functions 4

4. User Interface 5

5. System Components 7

6. Database 8

7. Required Software and Hardware 9

Table of Figures

Table of Contents 2

Table of Figures 2

1. Introduction 3

3. Required System Functions 4

4. User Interface 5

5. System Components 7

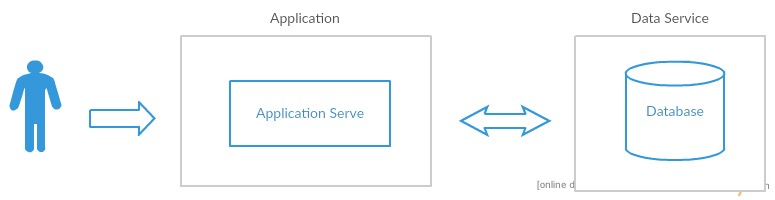
6. Database 8

7. Required Software and Hardware 9

1. Introduction

This system is needed to help colleges when students are registering for classes. Its intention is to present available classes and times to which the student can add to their future semester schedule. It is beneficial to the college institutions since it eliminates the need for written documents and the process is done quicker and the students will be aware of the classes offered quickly and easily. This earns the institution profits from classes enrolled in. The Requirements Document is crucial to this project since it lists all the necessary information needed to complete it. This project is intended to be versatile but small and effective so it is adaptable and marketable. Compare with CunyFirst, this system can be more easily arrange student schedule. The rest of this document contains technical information related to the project.

**2. Application Architecture**

****

This class scheduling application basically takes information that user had input, send request to the database, then sort all the data that have received and present the available sequences for the user.

This class scheduling application has three major components on application service:

Class scheduling section:

Institution, Term, Subject, Days of week, Available starting and ending time, Max Units and Submit button. Institution, Term and Subject section will implementing as drop down list, which listed the corresponding data from database. The Subject option interconnects to the second component Available Class. Once user selected a subject, the application sends request to database, which requesting all opened class of the subject and represent at the Available Class section. Days of week will representing as click box with label(Default is all week available). Available starting and ending time will representing as drop down list as well(Default is all day available). Max Units is a input box that only takes int(Default are 18 units.).

The Available Class box is real time reacting section that monitoring the Subject option. Once user selected a subject, the application sends request to database, which requesting all opened class of the subject and represent only class number(like CSC 126) and units at the Available Class box.

The Picked Class box showing all the class that user had selected from Available Class section is listed.

User Log In section:

A button with label “Log in” will showing right under the Max unit box that redirect user to second screen. The user log in section has one mainly function which keeps the class schedule user has favorited previously. In Login page, user can login to the application, create a new user account and reset the password via email as well.

For adding favorite schedule to user history, a star button will represent at the top-right corner of each scheduling result page.

Class Review section:

A button with label “Write A Class Review” will also showing right under the Max unit box that redirect user to second screen. User can write a class review neither logged in or using as a guest. The class reviews are store to the database and every user could view it at the scheduling result page. A Review button will present at the end of each class information section that connecting to the corresponding professor.

3. Required System Functions

In main screen of the scheduling application, there are three information condition that user must to provide:

1 Institution (The college of user’s, for database choosing purpose)

2 Term (The term of user’s, for database choosing purpose)

3 Class (The class that user wants to schedule. The user can find the opened classes by clicking the Subject option, and it will drop down a catalog of subjects. After the subject is selected, all the opened classes, which are without the section number and meeting time but units, will show up on the middle of the main screen line by line, for example: CSC 126 4 units, CSC 211 4 units. All the selected classes will showing on the right side of the main screen.)

For optional information option:

1 Days of week (User can filter the schedules by providing the days information to the application. Default is whole week available.)

2 Maximum units (User can limits the maximum units of each schedules. Default are 18 units.)

3 Adding adding favorite schedule to user history, user has to be logged in.

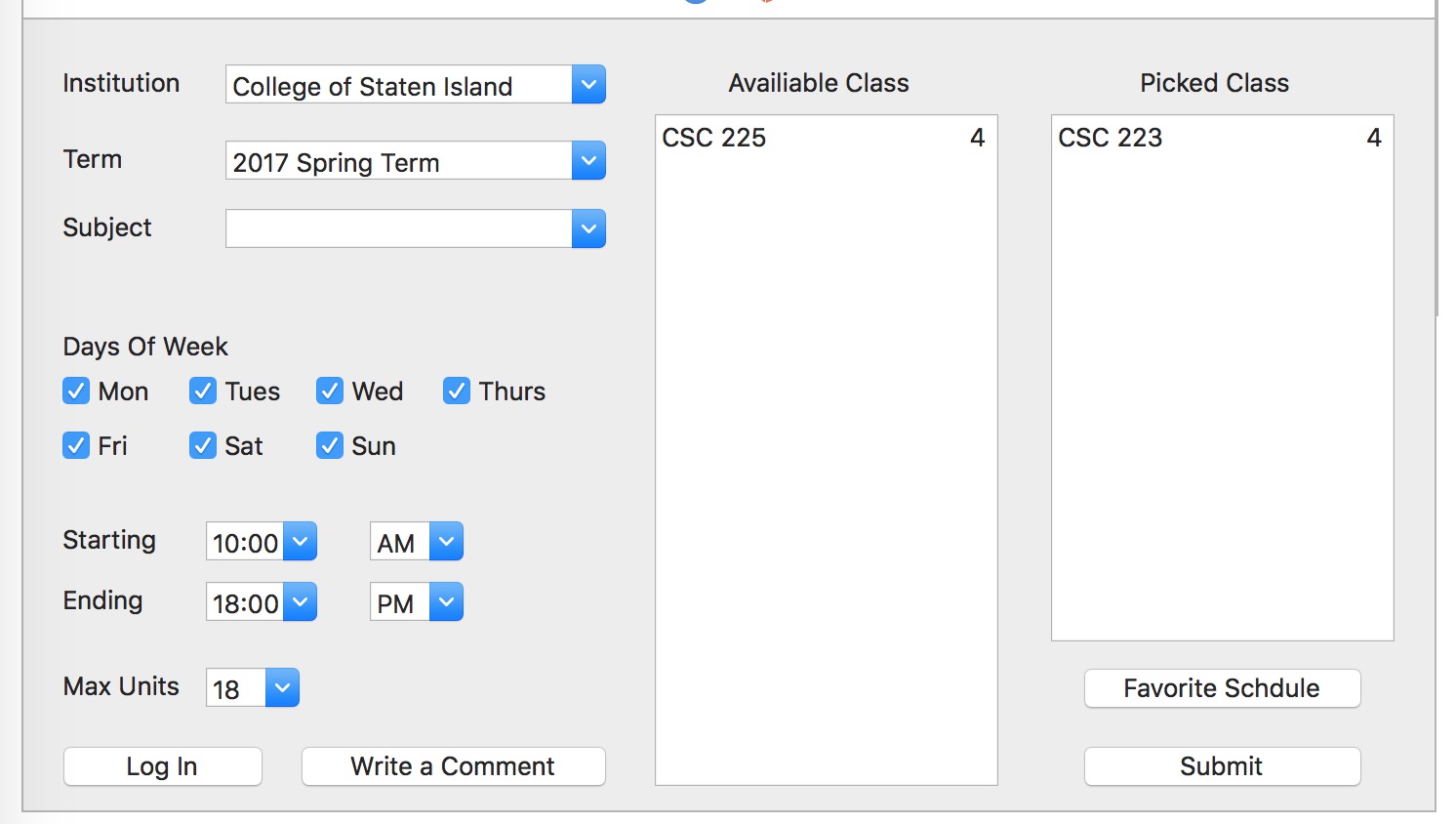
The application will be able to provide :

1 A series of sorted class’ schedule for user. Each schedule are including all the available combinations of section and meeting time that limited by maximum units.

2 Based on all available schedules, filter the schedules by the optional conditions that user had selected(Days, Units).

3 Checking class reviews with corresponding professor that written by previous students.

4 Check the class schedule that favorited.

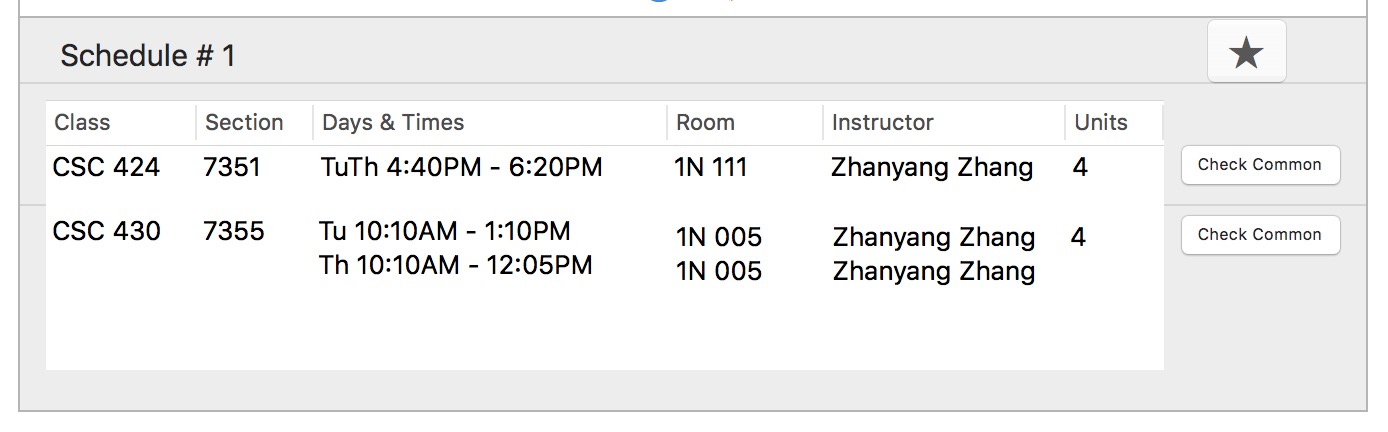
4. User Interface

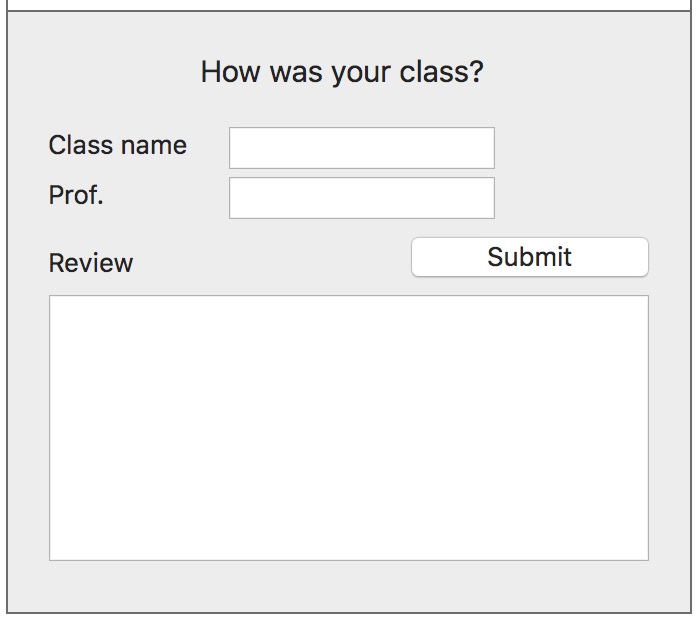
Main Page

Institution, Term and Subject section will implementing as drop down list, which listed the corresponding data from database. The Subject option interconnects to the second component Available Class. Once user selected a subject, the application sends request to database, which requesting all opened class of the subject and represent at the Available Class box. Days of week will representing as click box with label(Default is all week available). Available starting and ending time will representing as drop down list as well(Default is all day available). Max Units is a input box that only takes int(Default are 18 units.).

The Available Class box is real time reacting section that monitoring the Subject option. Once user selected a subject, the application sends request to database, which requesting all opened class of the subject and represent only class number(like CSC 126) and units at the Available Class box.

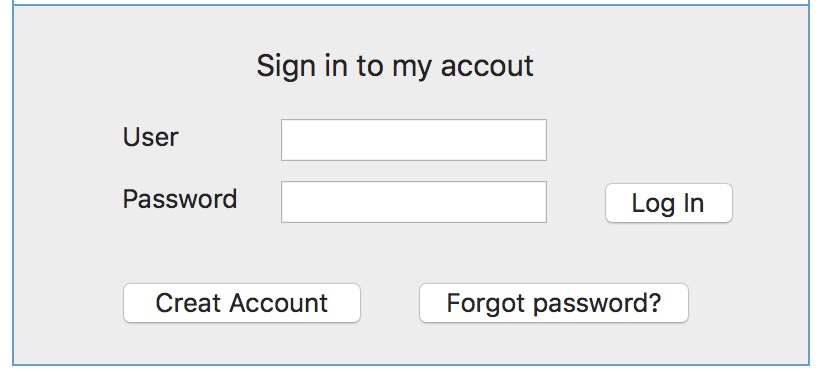
The Picked Class box showing all the class that user had selected from Available Class section is listed.

Schedule Page

After user selected the classes, the application will generate a series of available schedules for user. On the right corner of each schedule, a favorite button will present as a star, which user can press to add to favorite. Further more, there are a Check common button showing on the end of each class, which connected to all the commons of the instructor that stored in database.

Add common page

User can write a class review neither logged in or using as a guest. This page will pop up after the user press the Write a common button from Main page.

Log In page

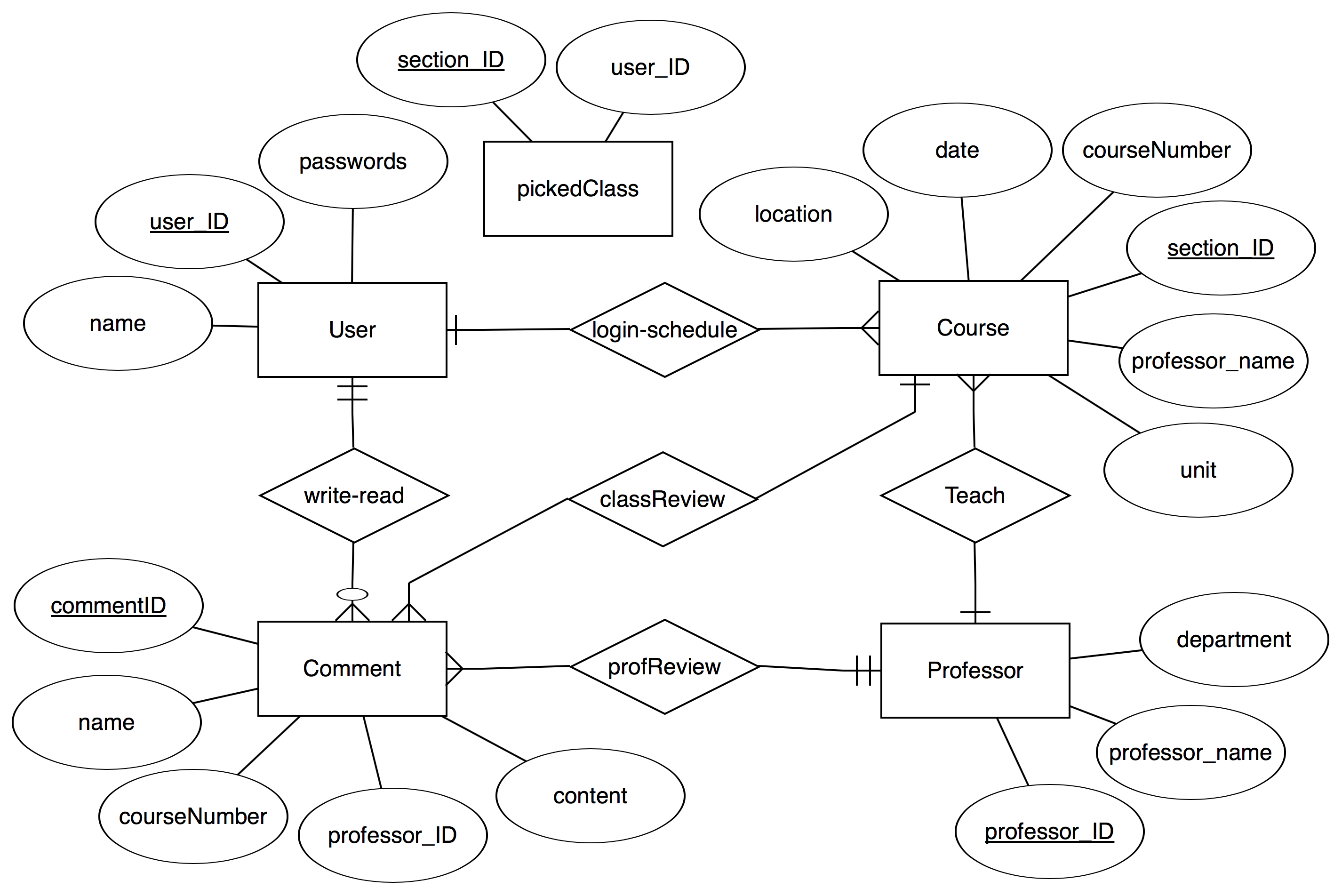
This page will pop up after the user press the Log in button from Main page. After logged in, users can store the favorite schedules to the database.

5. System Components

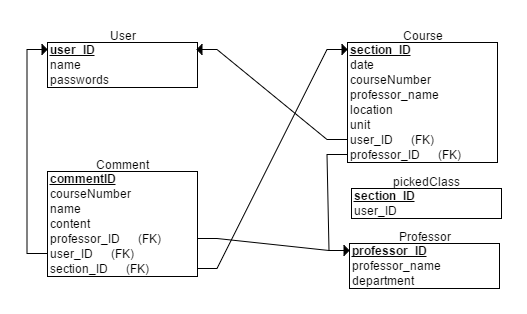
List of system components or modules, their functions and how they interact with each other.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Function Description | Input Data | Output Data | Dependencies |
| Log in | User log in | User name and password |  |  |
| submit schedule | After user pick all the classes, submit the classes | classes from Picked class | a series of schedules | Picked class |
| Write a Common | user can write a common for a class and professor | text input |  |  |
| Favorite schedule | user can store their favorite schedule |  | a series of schedules from database | Log in |
| Check Common | user can check commons with the corresponding professor before they choosing class | button click | a table page of commons |  |

6. Database

ER Data Model

Database Schema



7. Required Software and Hardware

|  |
| --- |
| Internet connection |
| Windows based |
| Database server |

**Appendix** – If any.