Project Specification

Michael Lee (raehyunl), Winstone Yang (winstony)

1. The product backlog: a complete list of all functionality (i.e., the actions) of your project, and an English description of each action. We strongly suggest that you organize these features into groups/modules based on related functionality.

Module 1: User Authentication

- We plan to have a login button upon user entry to the website, which will prompt each user to log into their account using username and password
- In addition, users should be able to register, taking them to a page where they will be inputting their name, login, password, and retype password
- For logout, a logout button will be placed on the top right corner of the page banner that will be displayed on every page of the website once user is logged in

Module 2: Voting System (the Main Website)

- On the main page, users will be able to view the list of classes ordered from lowest to highest average credit hours. Sorting should happen based on user votes.
- Users should be able to vote by clicking on 'vote' button and select credit hours for each class to vote for the said class. In addition ratings should be defined for a specific range, perhaps the maximum capped at 25 hours maximum (the FCE hours for 15-410 Operating Systems) just in case mischievous users try to input unrealistically large hours for a class.
- We also have checkboxes for filtering - for example, if you want to view minis or courses under 9 units as well, the users will have an option to do so. Users could also be able to filter classes based on the specific department offering the course (ex: 15-XXX or 17-XXX)
- Upon clicking on the class (maybe with hidden details showing up upon clicking), users will be able to see the details for that class like basic course description, department, prerequisites, corequisites, etc

Module 3: Class adding

- Users can add a class with numbers and they can select a credit hour for each class that will count as a first vote for that said class. Or they could just submit the class without selecting credit hours. Either way, this newly added class will be demonstrated on the main page for other users as well.
- The courses should have five digit ID, and the system should also prevent users from being able to add redundant classes that already exist on the database

Module 4: User profile

- User dashboard should display the voting history: aka the classes that the user voted for and what credit hours they assigned for that class during their vote

- Edit profile: users should be able to change their profile information such as name, image, and password

2. The first sprint backlog: a complete list of the functionality you will complete during your first sprint, and how that work is allocated among your team members.
We aim to complete the following functionalities for the sprint:

- Implement the user authentication module (registration, login/logout)
- Create the class voting feature with basic functionalities like classes and voting
- Implement basic class management through addition
- Design a simple homepage layout to display classes and voting options. We won't have the filters yet

For the division of work, I (Michael) have decided to work on the class voting feature as well as the basic adding of classes, since they both mainly deal with the mechanics of how the main page works and primarily deal with backend technologies.
On the other hand, Winstone will take on the frontend design of the homepage as well as the authentication protocols dealing with registration, login, and logout.

3. The name and Andrew ID of the product owner for the first sprint.

- Michael Lee (raehyunl)

4. A complete specification of the data models used by your application. This may be written in as Django models, SQL, or an ERD or other diagram showing all your models. (models.py is best.)

```
from django.db import models
from django.contrib.auth.models import User

class Course(models.Model):
    name = models.CharField(max_length=255)
    description = models.TextField(blank=True, null=True)
    average_credit_hours = models.FloatField(default=0)

    def __str__(self):
        return self.name

class Student(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    course = models.ForeignKey(Course, on_delete=models.CASCADE)
    estimated_credit_hours = models.FloatField()
```

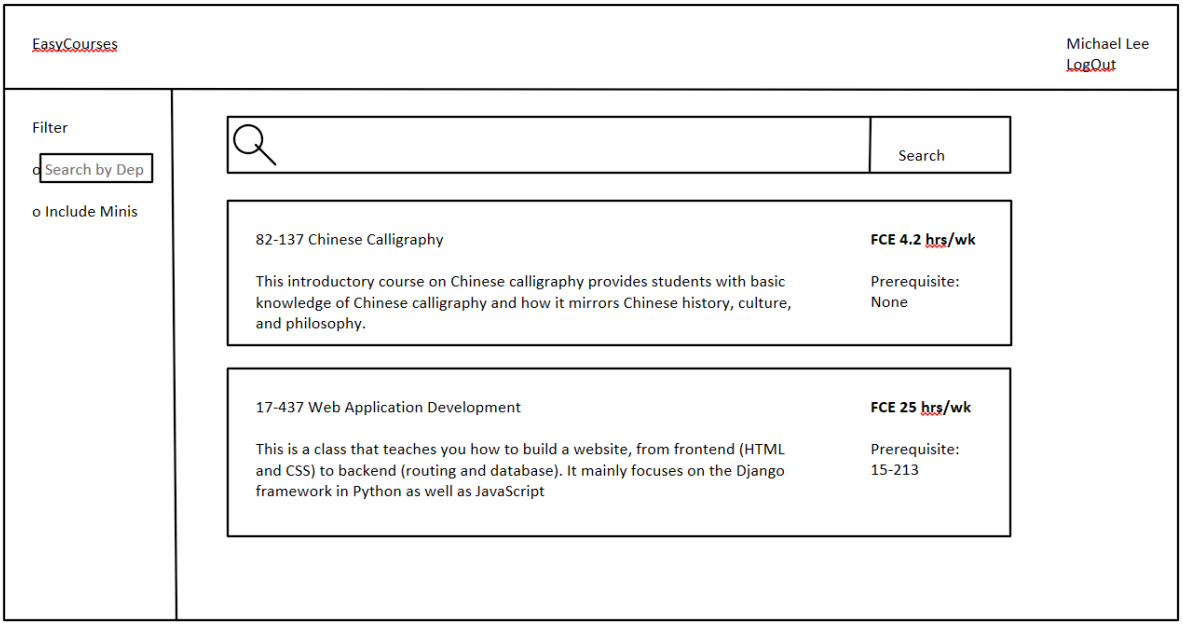
```
    def __str__(self):
        return f"{self.user.username} - {self.course.name}
({self.estimated_credit_hours} hours)"

    class Meta:
        unique_together = ('user', 'course')
```

The models that we have are Course and Student.

- Course model takes in fields name, description, and average credit hours.
- Student model takes in fields user, course, estimated_credit_hours, and has constraint unique_together which ensures each user can vote only once per class.

5. A complete set of drawn wireframes or HTML mockups for your application, for all non -trivial views within the application



EasyCourses

Michael Lee
LogOut

Add a Course

Search

Course Name: 57-209 The Beatles

Add Credit Hours: *5 hours*

SUBMIT

Welcome! Log In with Your Account:

Username: Insert Your Username Here

Password: Insert Password Here

Login

Forgot username?

Forgot password?

EasyCourses

Michael Lee
LogOut

My Profile Page

Edit Profile

My Profile Page

Add new image

My Votes

82-137 Chinese Calligraphy

You have voted:

6 hours

17-437 Web Application Development

You have voted:

25 hours