

# VirgilPlus - Course Selections Made Easy

## Our Team

Wei Luo (wl2671), Zhuangyu Ren (zr2209), Chi Zhang (cz2517), Danyang He (dh2914)

## Overview

VirgilPlus is a software that makes course planning effortless for students. Whereas currently students have to browse through an entire catalog that contains mostly irrelevant or unavailable courses, VirgilPlus lets students bypass the frustration of picking out the courses right for them. In particular, given a set of courses that a student has already taken and a set of preferences such as graduation year, day of the week or time window, VirgilPlus outputs the courses that fit these constraints up to a threshold, and eases the process of choosing desired courses.

## Specifications

### *Data:*

Our data will come from the Columbia CS course catalog. We will collect information such as course number, course name, course description, class times, and prerequisites. These raw data will be stored and retrieved using a combination of a database management system and a querying language, such as pgAdmin and PostgreSQL. Notice that for every course, we obtain the prerequisites. So by using these topological constraints, we can construct a forest (a graph consisting of multiple components, each of which is a tree). Naturally, such data structure will be stored in our database with respect to the nodes' adjacency relations.

### *Algorithm:*

The core algorithm is a graph search. After prompting for user input, we know which courses (i.e. nodes) that the students will prefer. Using this information, we can perform an A\* or BFS search starting from the nodes that correspond to the courses that the student has already taken, and assigning weights to the nodes dynamically according to the preferences. After performing the search for only one layer outward (which corresponds to the courses that are available to the student right now), we output the top-weighted terminal nodes as the course recommendations.

### *Language & Platform:*

We will use Java 8 and Windows.

### *Technical Specs:*

The software will be implemented in Java 8. External libraries will include Apache for general purposes, and possibly Alchemy API for performing NLP on user input and Hibernate for mapping between Java objects and database tables.

## User Stories

*User1:*

As a new graduate student at Columbia University, I want to get suggestions on how to arrange courses for my study so that I can complete track requirements. My conditions of satisfaction are: for every search, I can get multiple feasible course paths which lead me to fulfill track requirements successfully.

*User2:*

As a student who has studied in Computer Science department for some time and has already completed several courses, I want to check what other courses I should take to meet my track requirements. My conditions of satisfaction are: after inputting my previous courses and my track, I can get multiple course paths to follow to achieve the graduation requirement.

*User3:*

As a high-school student who is enthusiastic about pursuing B.S. degree in Computer Science, I want to know what kind of courses the department offered and search keywords to get related course information so that I can gather more information about CS@CU knowledge system. My conditions of satisfaction are: after inputting my interested course keywords, I can get a list of courses sorted by relevance, shown with the course description and instructor. If there are no relevant courses, it should return 'no relevant course in record'.

*User 4:*

As a student who is at the next semester, I want to select suitable courses for graduation. My conditions of satisfaction are : after inputting my previous courses and my track, I can get multiple combinations of the last two courses. If I already fulfill graduation requirement, it should remind me to start graduation process.

## **Testing Plan**

We support two types of inputs: from .json and from command lines. The inputs from .json will be shown as below, as the inputs from command lines are similar so we just list json test inputs.

*User1:*

```
{
  "test1":[
    {"trackID": 1},
    {"trackID": "Machine Learning"},
    {"trackID": "Computer Science"}
  ]
}
```

Expected outputs:

- 1: return "No such track exists";
- 2: 10 courses that fits the "Machine Learning" track requirements;
- 3: return "No such track exists";

*User2:*

```
{
  "test2":[
    {"trackID":1,"taken_courses":["WCOMS4111","ECOMS6998","WCSOR4246","WCOMS4121"]},
    {"trackID":"Machine
Learning","taken_courses":["WCOMS4111","ECOMS6998","WCSOR4246","WCOMS4121"]},
    {"trackID":"Machine Learning", "taken_courses":["Machine Learning for Data Science","Intrusion
Detection Systems","Computer Graphics","Natural Language Processing"]},
    {"trackID":"Computer
Science","taken_courses":["WCOMS4111","ECOMS6998","WCSOR4246","WCOMS4121"]},
    {"trackID":"Machine
Learning","taken_courses":["8000","ECOMS6998","WCSOR4246","WCOMS4121"]}
  ]
}
```

Expected outputs:

- 1: return “No such track exists”;
- 2: 6 courses that fits the “Machine Learning” track requirements except those courses user has taken;
- 3: 6 courses that fits the “Machine Learning” track requirements except those courses user has taken;
- 4: return “No such track exists”
- 5: return “Input courses do not exist”

*User3:*

```
{
  "test3":[
    {"keywords":"Machine Learning"},
    {"keywords":"System"},
    {"keywords":"history"},
    {"keywords":1}
  ]
}
```

Expected outputs:

- 1: return all the courses whose names or descriptions contain the keyword “Machine Learning”;
- 2: return all the courses whose names or descriptions contain the keyword “System”;
- 3: return “No relevant courses in record”;
- 4: return “No relevant courses in record”;

*User4:*

```
{
  "test4":[

    {"trackID":1,"taken_courses":["WCOMS4111","ECOMS6998","WCSOR4246","WCOMS4121","WCO
MS4236","WCOMS4995","ECOMS6998","WCOMS4172"]},
```

```

    {"trackID":"Machine
Learning","taken_courses":["WCOMS4111","ECOMS6998","WCSOR4246","WCOMS4121","WCOMS
4236","WCOMS4995","ECOMS6998","WCOMS4172"]},
    {"trackID":"Machine Learning", "taken_courses":["Machine Learning for Data Science","Intrusion
Detection Systems","Computer Graphics","Natural Language Processing","Security Architecture and
Engineering","Introduction to Computational Learning Theory","Introduction To
Cryptography","Networking Laboratory"]},
    {"trackID":"Computer
Science","taken_courses":["WCOMS4111","ECOMS6998","WCSOR4246","WCOMS4121","WCOMS4
236","WCOMS4995","ECOMS6998","WCOMS4172"]},
    {"trackID":"Machine
Learning","taken_courses":["8000","ECOMS6998","WCSOR4246","WCOMS4121","WCOMS4236","
WCOMS4995","ECOMS6998","WCOMS4172"]}
  ]
}

```

Expected outputs:

- 1: return "No such track exists";
- 2: 2 courses that fits the "Machine Learning" track requirements except those courses user has taken;
- 3: 2 courses that fits the "Machine Learning" track requirements except those courses user has taken;
- 4: return "No such track exists"
- 5: return "Input courses do not exist"