Pocket Campus Design Document

Version 1.0.0

The Corner members:
Zechariah Fisher-Coleman
Vigneshwaran Madappan Chinnasami
Drew Dabe
Kyle Akers
Isaac Mayernik

Table of Contents

Introduction

System Overview

The system aims to provide all college students at the University of South Carolina with a clear and concise path towards completing their chosen major. Target users include students, advisors, parents, and other faculty and staff. The system will be accessible on both computers and mobile devices, catering to the diverse student body of the university. The system's interface will be intuitively designed to ensure ease of use for users from various departments and majors, promoting accessibility and user-friendly navigation.

References

- Team Corner Requirements Document
- Requirements List

Design Overview

Introduction

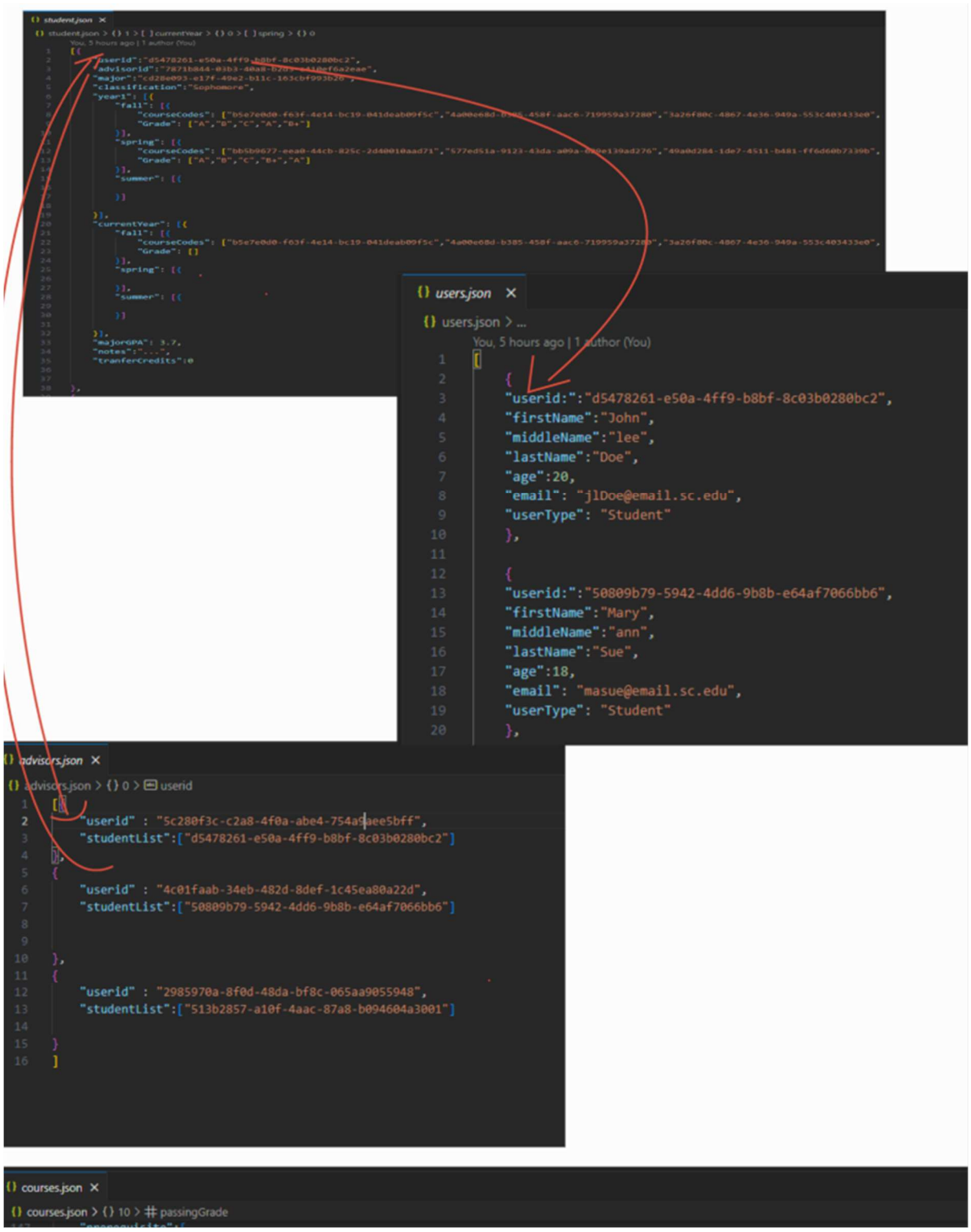
The system is designed to display and manage a major map and semester plan for students who are completing their degree. It has features such as storing the classes they have taken, adjusting the major map based on those classes, and storing grades. The system will also allow advisors to track students' progress through their degree and notify them if they may lose their scholarship.

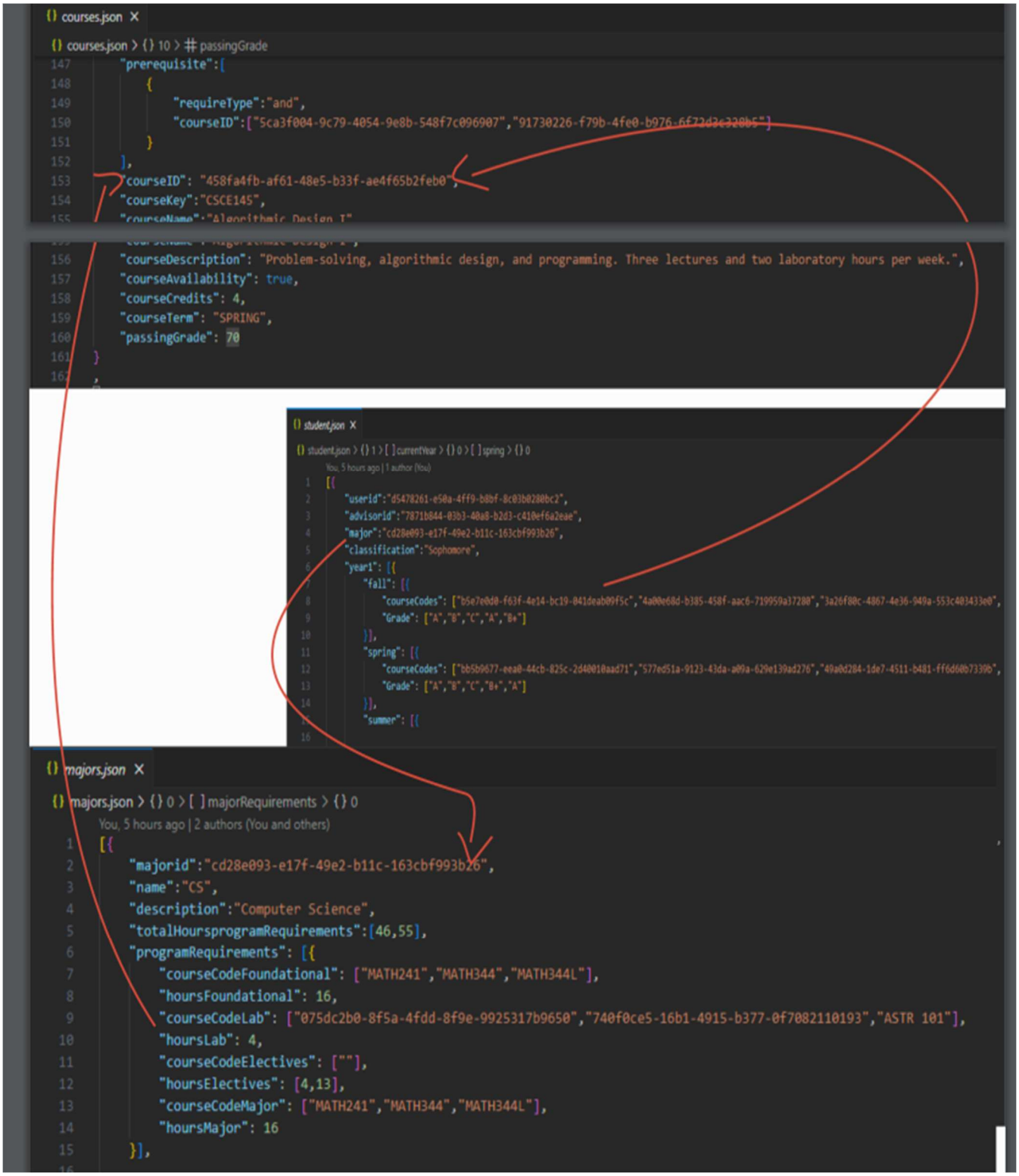
Environment Overview

The system will run within the University of South Carolina's network environment, accessible to users via various devices and operating systems including Mac, iOS, Windows, and mobile platforms. Meetings and collaborations will primarily occur online due to the lack of a shared physical workspace.

Data Storage

Data will be stored as JSON files containing information about majors, courses, students, advisors, and other relevant entities. Five JSON files will be utilized: `advisors.json`, `courses.json`, `majors.json`, `students.json`, and `users.json`. Each file will contain an array of objects with unique identifiers, facilitating connections between entities.


student.json
users.json
advisors.json
courses.json



JSONs Image

Information pointed to is described below.

advisors.json:

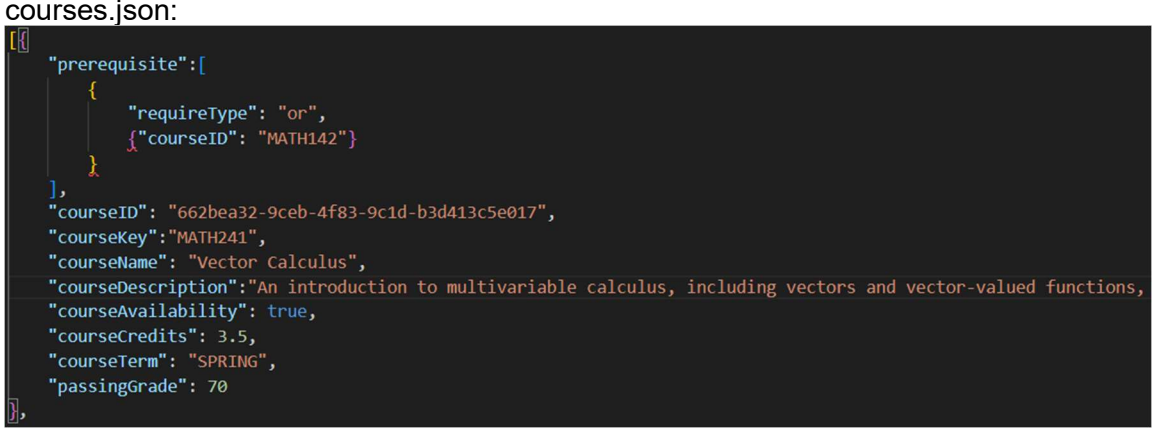
```
[{
    "userid" : "5c280f3c-c2a8-4f0a-abe4-754a9aee5bff",
    "studentList":["d5478261-e50a-4ff9-b8bf-8c03b0280bc2"]
},
{
    "userid" : "4c01faab-34eb-482d-8def-1c45ea80a22d",
    "studentList":["50809b79-5942-4dd6-9b8b-e64af7066bb6"]


},
{
    "userid" : "2985970a-8f0d-48da-bf8c-065aa9055948",
    "studentList":["513b2857-a10f-4aac-87a8-b094604a3001"]

}
]
```

The advisor will have an id which refers to the users.json. The UUID in the "studentlist" refers to a student id, and will refer to the students.json.

courses.json:



```
[{
    "prerequisite":[
        {
            "requireType": "or",
            {"courseID": "MATH142"}
        }
    ],
    "courseID": "662bea32-9ceb-4f83-9c1d-b3d413c5e017",
    "courseKey":"MATH241",
    "courseName": "Vector Calculus",
    "courseDescription":"An introduction to multivariable calculus, including vectors and vector-valued functions,
    "courseAvailability": true,
    "courseCredits": 3.5,
    "courseTerm": "SPRING",
    "passingGrade": 70
},
```

Much of the course information is stored here. The "courseid" may be pointed to by majors or the student if they have taken that course.

majors.json:

```
[{
    "majorid":"cd28e093-e17f-49e2-b11c-163cbf993b26",
    "name":"CS",
    "description":"Computer Science",
    "totalHoursprogramRequirements":[46,55],
    "programRequirements": [{
        "courseCodeFoundational": ["MATH241","MATH344","MATH344L"],
        "hoursFoundational": 16,
        "courseCodeLab": ["075dc2b0-8f5a-4fdd-8f9e-9925317b9650","740f0ce5-16b1-4915-b377-0f7082110193","ASTR 101"],
        "hoursLab": 4,
        "courseCodeElectives": [""],
        "hoursElectives": [4,13],
        "courseCodeMajor": ["MATH241","MATH344","MATH344L"],
        "hoursMajor": 16
    }],

    "carolinaCore": [{
        "courseCodeCarolinaCore": ["ENGL101","ENGL102","SPCH101"],
        "hoursFoundational": [35,41]
    }],
    "majorRequirements": [{
        "courseCodeMajor": ["CSCE311","MATH344","MATH344L"],
        "hoursFoundational": 21,
        "courseCodeMajorElectives": ["075dc2b0-8f5a-4fdd-8f9e-9925317b9650","740f0ce5-16b1-4915-b377-0f7082110193","ASTR 101"],
        "hoursElectives": 9,
        "courseCodeConcentration": ["MATH241","MATH344","MATH344L"],
        "hoursMajor": 12
    }],
    "totalMajorHours":120
```

Much about the major information is stored here. Each major has an id that may be pointed to. The UUIDs in courseCodeLab will refer to the courses.json.

students.json:

The “userid” will refer to the users.json. Their advisor has an id and will refer to advisors.json. Their major will point to either the CS or CIS major in the majors.json. The various courses codes and their grades will point to the courses.json.

user.json:

Information will be stored here about each user and may be pointed to by the other json files.

Structural Design

Class Diagram
UML Diagram

Class Descriptions

Application:
Purpose: This is the main system the user will interact with.

UserList:
Purpose: A class that holds the list of all users within the system. The class also holds information about each user.

ClassList:
Purpose: A class that holds the list of all classes a student may take under the two majors within the system. The class also holds information about each class.

MajorList:
Purpose: A class that holds the list of majors and information about them.

User:
Purpose: This class initializes the user based on information provided by the list and called from by the system.

Class:
Purpose: Different information about the course such as its referable key, name, and description are stored here. The courses may be checked if there are a list of courses that are its prerequisites and/or core-requisites.

CourseChoice:
Purpose: For some variabilities, such as Carolina core courses, there are many courses a student may pick from, and this class stores a hashmap of those. It may also check if the course has a prerequisite and/or core-requisities.

Major:
Purpose: Information about the two majors are stored here such as the name, a description, and array lists like electives for the major. It may get the array list of required courses for the major upon call.

Advisor:
Purpose: The array list of students the advisor is supervising is stored here. Information about the advisor comes from extended the user class. The user may search for a student based on their

userID or last name. They may also add or remove a student from the supervising list. The advisor may also edit the student's advisor note.

Admin:
Purpose: The purpose of the admin is to add or remove students from the system.

Student:
Purpose: Information about the student is derived from initialization of the user class, such as first name, last name, an array list of courses taken, and what major they have. The student may calculate their GPA, get their major map, and make an eight-semester plan.

Major Map:
Purpose: The major map creates a major map based on the student's major, concentration, and the liberal arts, Carolina core, and electives they may want. It also considers the student's taken courses. The student may also make a what-if major map, to see about changing majors.

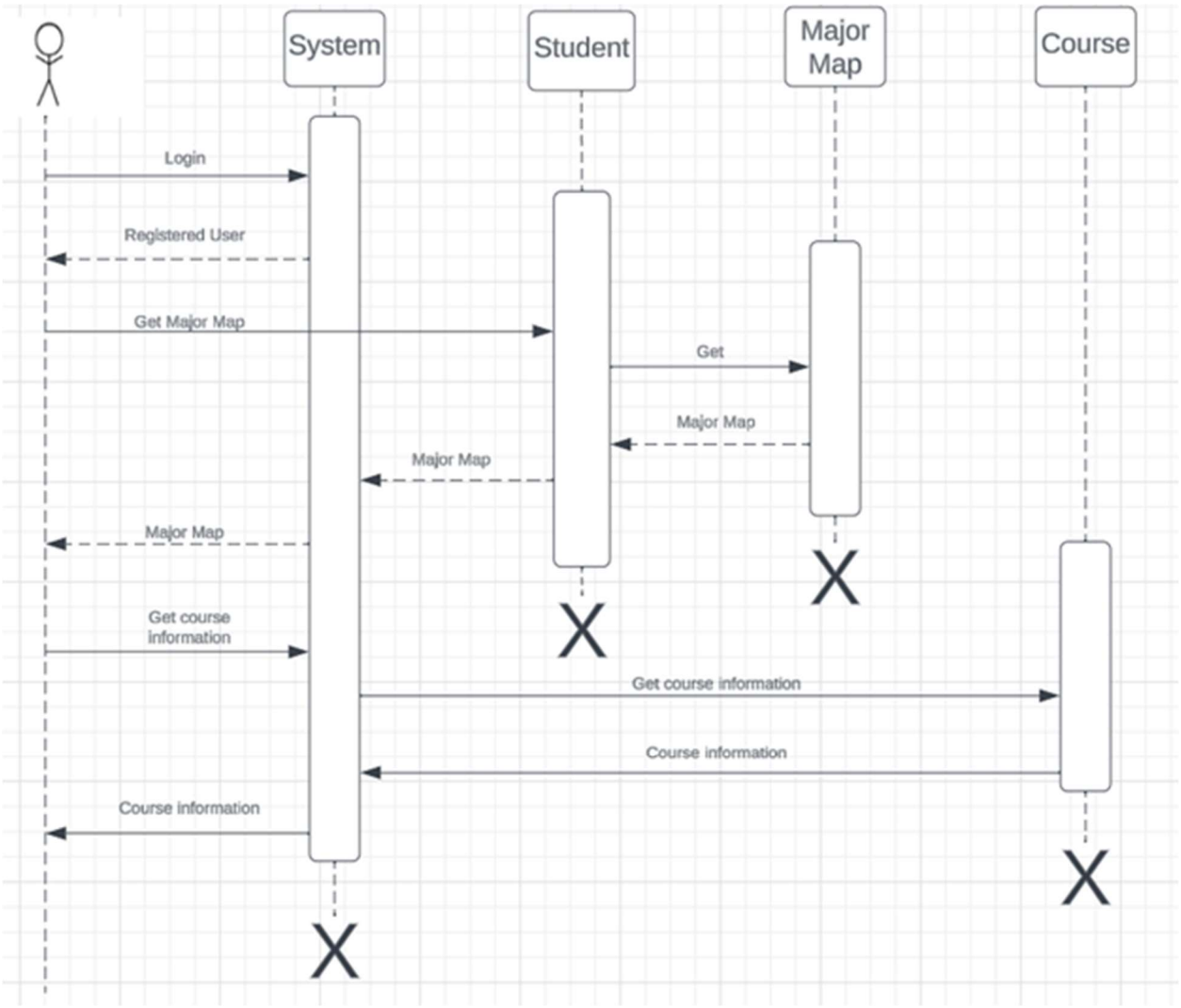
Semester Plan:
Purpose: The semester plan creates an eight-semester plan based on their major and already completed classes. The progress of the semester plan is also shown based on completed courses as the student completes their degree.

Enum
RequireType:
Purpose: Designates between if a course is either a prerequisite and/or corerequisite.

Scenarios

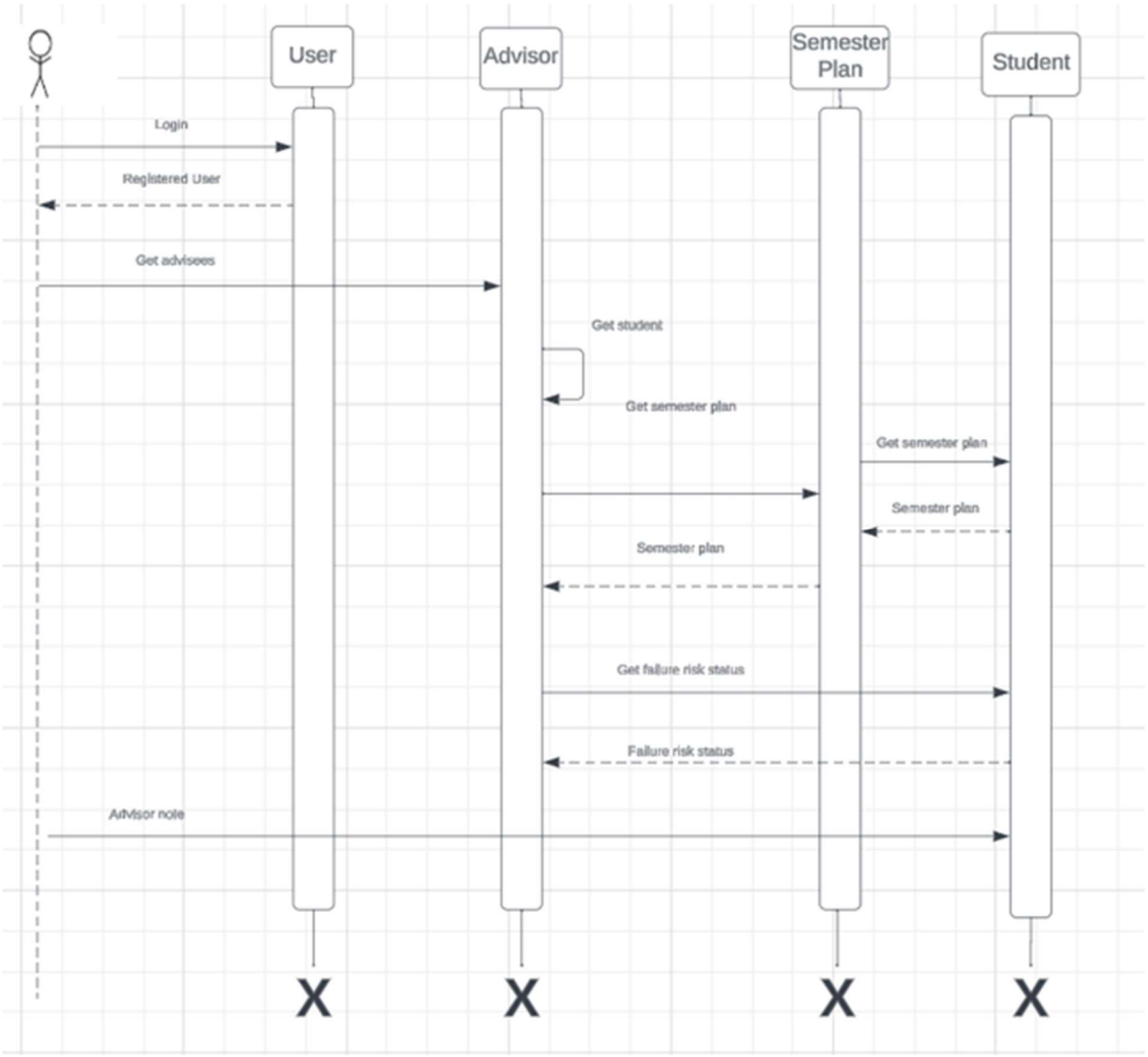
Scenario 1

- **Scenario Name:** Student Login
- **Scenario Description:** A student logs into the system, views their major map listing completed courses, identifies a missing requirement, and looks up course details.
- **Sequence Diagram:**



Scenario 2

- **Scenario Name:** Advisor Login
- **Scenario Description:** An advisor logs into the system, accesses their list of advisees, generates an Semester plan for a specific advisee, identifies scholarship risks, and adds a note to the student's account.
- **Sequence Diagram:**


User
Advisor
Semester Plan
Student
Login
Registered User
Get advisees
Get student
Get semester plan
Get semester plan
Semester plan
Semester plan
Get failure risk status
Failure risk status
Advisor note