# CAPSTONE PROJECT INTERMEDIATE REPORT I

Submitted to

Computer Science Department

College of Science

Kuwait University

**Advisor**

Dr. Maha Al-Alabduljalil

**Group Members**

Amna Ali 2131110038

Asmaa Sultan 2151117051

Monday, February 17th, 2020

# Scheduley

## For Fast Track Course Scheduling


**1 Table of Contents**

## 2 Table of Figures

## 3 Table of Tables

# 1 Introduction

Universities are seen as the interface to a country's future civilization. Through them, a new generation of promising workers, problem-solvers, and achievers are born. For such a critical entity, it is mandatory to provide the students all possible resources that encourage learning, research, and success in their study fields. However, the issue of planning their academic years, which courses to take and when is best to take them are very stressful decisions and had been an obstacle for many students at Kuwait University.

In this project, we decided to automate a solution for this issue for the Computer Science department's students and instructors at Kuwait University. This report will help explain the progress made so far, in addition to the future plan to reach the goal delivered by this project.

For any solution, the problem must first be defined in thorough. To do so, we applied requirement gathering methods to help understand the different factors involved, that might be hidden or unclear to us. Getting survey responses from Computer Science students with different academic progress, GPAs and experiences was a success. Furthermore, interviewing instructors and decision makers in the department added a stretch to the problem definition.

For the project, the system is preloaded with all the mandatory courses for the students to graduate from CS at KU following its major sheet guidelines. This will be saved internally as an acyclic directed graph [1] with multiple roots. Vertices are the courses and edges are the dependency across these courses. Since the CS major sheet has mandatory, elective and free list of courses, some courses appear to be isolated while others are connected to the graph. The system assumes three types of users, students, instructors and administrators. For students, the page requires an as input the *academic progress* which are the list of courses passed along with their grades. After that, the system analyzes the input and suggests a list of courses to be taken next semester referred to as, the *advised plan*. The algorithm views each course as an unvisited vertex until it has been taken by the student. Following the decision to use a graph, a search for an appropriate traversal method was needed. The method needs to optimize the traversal path with respect to time needed to graduate and is unique to each student's circumstances.

This system is implemented as a web-application using Django web framework. This document summarizes the work done so far, and how much is covered from the plan to reaching the product.

# 2 System Requirements

In this section, the team applied requirements engineering gathering techniques that helped set feasible goals for the project.

## 2.1 Requirements Elicitation Approach

Requirement engineering [2] mechanisms are meant to help understand and analyze the requirements meant to be provided by the system in addition to negotiate and validate those requirements to specify them unambiguously. Specifically, the steps included are as follows:

- **Inception**: asking questions to establish a basic understanding of the problem to offer solutions.
- **Elicitation**: to bring forth requirements that are most important.
- **Elaboration**: to create different user scenarios to better describe data, functional, and behavioral requirements.
- **Negotiation**: to decide which requirements are mandatory and which might be prone to later modifications.
- **Specification**: a prototype is made to design web interface requirements and how are carried out.
- **Validation**: to answer a set of questions such as: Is each requirement consistent with the overall objective of the system? Are the requirements bounded and unambiguous?

Sources of requirements gathering included the following:

- **Brainstorming**: team members and their supervisor gather frequently to propose different student scenarios and discuss possible solutions to be embedded gradually to the system prototype. This helped specify how each user interaction should be enabled as an input and what's the expected output.
- **Focus Groups**: students of classes 2012, 2013, 2014, 2015, and 2016 were asked to answer the questions while also providing input they believe is necessary. This range seemed the best fit as those students do have enough knowledge and experience to provide, and because of their delayed progress -if any- is what the team hopes to guide future students against. Each offered a different opinion and during the elicitation phase, the team decided what is achievable.
- **Interviews**: other stakeholders such as instructors, the general advisor, and other decision-makers in the computer science department were interviewed to get different perspectives, understand what the optimal solution is, and what the system is needed to offer.

## 2.2 Requirements Specification

This section will describe the services that the system will provide. There are two types of requirements which are functional and non-functional requirements.

### 2.2.1 Functional requirements

Requirements that have been obtained by the above methods will be listed below. Each requirement will have an ID, a description, their acceptance criteria, and a priority.

| ***ID*** | ***US1*** |
|---|---|
| ***Content:*** | As a student, I'd like to view the courses I should take next semester so that I can graduate on time. |
| ***Acceptance Criteria:*** | Given that the student passed the prerequisite courses<br>When the student views his plan<br>Then the plan should suggest what courses he or she should take for the rest of their academic progress. |
| ***Priority:*** | High |

| ***ID*** | ***US2*** |
|---|---|
| ***Content:*** | As a website user, I'd like to login the system with my university ID. |
| ***Acceptance Criteria:*** | Given the user had an assigned ID by the university<br>When the user wants login<br>Then the system should allow them to login with the ID. |
| ***Priority:*** | High |

| ***ID*** | ***US3*** |
|---|---|
| | As a website user, I'd like to modify my profile information. |

| | |
|---|---|
| ***Content:*** | |
| ***Acceptance Criteria:*** | Given the user is logged in<br>When the user wants to modify his profile information<br>Then the system should allow them to do so |
| ***Priority:*** | High |

| ***ID*** | ***US4*** |
|---|---|
| ***Content:*** | As an admin, I'd like to create the accounts myself so there won't be any users outside the department registered in the system. |
| ***Acceptance Criteria:*** | Given that the user is a student or instructor in the Computer Science department<br>When the user wants to create the account<br>Then the system should allow the admin to create one for them. |
| ***Priority:*** | High |

| ***ID*** | ***US5*** |
|---|---|
| ***Content:*** | As an admin, I'd like to delete the accounts of people who no longer work or study in the Computer Science Department. |
| ***Acceptance Criteria:*** | Given that the user no longer needs to use the system<br>When the admin wants to delete their account<br>Then the system should allow the admin to delete the accounts. |
| ***Priority:*** | Low |

| ***ID*** | ***US6*** |
|---|---|
| | As a website user, I'd like the interface design to be simple to understand |

| | |
|---|---|
| ***Content:*** | |
| ***Acceptance Criteria:*** | Given the user wants to use the functions offered by the website<br>When the user uses the website<br>Then the website should be simple to use. |
| ***Priority:*** | Low |

| ***ID*** | ***US7*** |
|---|---|
| ***Content:*** | As a website user, I'd like the interface design to be simple to understand |
| ***Acceptance Criteria:*** | Given the user wants to use the functions offered by the website<br>When the user uses the website<br>Then the website should be simple to use. |
| ***Priority:*** | Low |

| ***ID*** | ***US8*** |
|---|---|
| ***Content:*** | As a website user, I'd like the interface design to be simple to understand |
| ***Acceptance Criteria:*** | Given the user wants to use the functions offered by the website<br>When the user uses the website<br>Then the website should be simple to use. |
| ***Priority:*** | Low |

| ***ID*** | ***US9*** |
|---|---|
| ***Content:*** | As a student, I'd like to view how my progress is reflected in the major sheet. |

| | |
|---|---|
| ***Acceptance Criteria:*** | Given the student entered semesters he or she have passed<br>When the student views his major sheet<br>Then the website should view his updated major sheet. |
| ***Priority:*** | Low |

| ***ID*** | ***US10*** |
|---|---|
| ***Content:*** | As a student, I'd like to view how my progress is reflected in the major sheet. |
| ***Acceptance Criteria:*** | Given the student entered semesters he or she have passed<br>When the student views his major sheet<br>Then the website should view his updated major sheet. |
| ***Priority:*** | Low |

| ***ID*** | ***US11*** |
|---|---|
| ***Content:*** | As a student, I'd like to view how my progress is reflected in the major sheet. |
| ***Acceptance Criteria:*** | Given the student entered semesters he or she have passed<br>When the student views his major sheet<br>Then the website should view his updated major sheet. |
| ***Priority:*** | Low |

| ***ID*** | |
|---|---|
| | ***US12*** |
| ***Content:*** | As a student, I'd like to view how my progress is reflected in the major sheet. |
| ***Acceptance Criteria:*** | Given the student entered semesters he or she have passed<br>When the student views his major sheet<br>Then the website should view his updated major sheet. |
| ***Priority:*** | High |

| ***ID*** | ***US13*** |
|---|---|
| ***Content:*** | As a student, I'd like to add courses that I can register for to my wishlist. |
| ***Acceptance Criteria:*** | Given the student passed the prerequisite courses<br>When the student wants to add to his or her wish-list<br>Then they should be able to add courses they can take to the wish-list. |
| ***Priority:*** | Low |

| ***ID*** | ***US14*** |
|---|---|
| ***Content:*** | As a student, I'd like to add courses that I can register for to my wishlist. |
| ***Acceptance Criteria:*** | Given the student passed the prerequisite courses<br>When the student wants to add to his or her wish-list<br>Then they should be able to add courses they can take to the wish-list. |
| ***Priority:*** | Low |

| ***ID*** | ***US15*** |
|---|---|
| ***Content:*** | As a student, I'd like to add courses that I can register for to my wishlist. |
| ***Acceptance Criteria:*** | Given the student passed the prerequisite courses<br>When the student wants to add to his or her wish-list<br>Then they should be able to add courses they can take to the wish-list. |
| ***Priority:*** | Low |

| ***ID*** | ***US16*** |
|---|---|
| ***Content:*** | As a student, I'd like to add courses that I can register for to my wishlist. |
| ***Acceptance Criteria:*** | Given the student passed the prerequisite courses<br>When the student wants to add to his or her wish-list<br>Then they should be able to add courses they can take to the wish-list. |
| ***Priority:*** | Low |

| ***ID*** | ***US17*** |
|---|---|
| ***Content:*** | As a student, I'd like to add elective courses that I can register for to my wish-list. |
| ***Acceptance Criteria:*** | Given the student passed the prerequisite courses<br>When the student wants to add to his or her wish-list<br>Then they should be able to add courses they can take to the wish-list. |
| ***Priority:*** | Low |

***US18***

| ***ID*** | |
|---|---|
| ***Content:*** | As an admin, I'd like to add the course information and prerequisite course. |
| ***Acceptance Criteria:*** | Given the courses are needed for planning of the Computer Science department<br>When the wants to add the course to the system to help with the advised plan for students<br>Then the system should allow them to do so. |
| ***Priority:*** | High |

| ***ID*** | ***US19*** |
|---|---|
| ***Content:*** | As an admin, I'd like to modify the course information and prerequisite course. |
| ***Acceptance Criteria:*** | Given the courses are needed for planning of the Computer Science department<br>When the wants to modify the course to the system to help with the advised plan for students<br>Then the system should allow them to do so. |
| ***Priority:*** | Low |

| ***ID*** | ***US19*** |
|---|---|
| ***Content:*** | As an admin, I'd like to modify the course information and prerequisite course. |
| ***Acceptance Criteria:*** | Given the courses are needed for planning of the Computer Science department<br>When the wants to modify the course to the system to help with the advised plan for students<br>Then the system should allow them to do so. |

| | |
|---|---|
| ***Priority:*** | Low |

| ***ID*** | ***US19*** |
|---|---|
| ***Content:*** | As an instructor, I'd like to add courses I want to teach for the next semester to my wish-list. |
| ***Acceptance Criteria:*** | Given the courses are in the Computer Science department<br>When the instructor wants to teach a course for the next semester<br>Then the system should allow them to do so. |
| ***Priority:*** | High |

| ***ID*** | ***US20*** |
|---|---|
| ***Content:*** | As an instructor, I'd like to delete a course from my wish-list in case I change my mind |
| ***Acceptance Criteria:*** | Given the courses are in the Computer Science department<br>When the instructor wants to remove it from their wish-list<br>Then the system should allow them to do so. |
| ***Priority:*** | Low |

| ***ID*** | ***US20*** |
|---|---|
| ***Content:*** | As an instructor, I'd like to add to my profile the two non-preferred time I don't want to teach in for the next semester. |
| ***Acceptance Criteria:*** | Given the instructor is in the Computer Science department<br>When the instructor wants to add his or her preferences<br>Then the system should allow them to do so. |

| | |
|---|---|
| ***ID*** | |
| ***Priority:*** | Medium |

| | |
|---|---|
| | ***US21*** |
| ***Content:*** | As a student, I'd like to for the plan to help me keep a reasonable GPA. |
| ***Acceptance Criteria:*** | Given the student is in the Computer Science department<br>When the student enters their academic progress<br>Then the system should help them calculate their GPA and tell them if it's reasonable or not. |
| ***Priority:*** | Medium |

| | |
|---|---|
| ***ID*** | ***US22*** |
| ***Content:*** | As a student, I'd like for the plan to help me keep a reasonable GPA. |
| ***Acceptance Criteria:*** | Given the student is in the Computer Science department<br>When the student wants to view his or her plan<br>Then the system should help them understand the grade they should get. |
| ***Priority:*** | Medium |

| | |
|---|---|
| ***ID*** | ***US23*** |
| ***Content:*** | As a student, I'd like to add courses I am taking in the current semester as pending, so I can update my progress immediately. |
| ***Acceptance Criteria:*** | Given the student is in the Computer Science department<br>When the student wants to add courses that they are taking to the system<br>Then the system should allow them to do so. |
| ***Priority:*** | High |

| ***ID*** | ***US24*** |
|---|---|
| ***Content:*** | As a student, I'd like to change the status of pending courses to completed or uncompleted after I get my grades. |
| ***Acceptance Criteria:*** | Given the student is in the Computer Science department<br>When the student wants to update their pending courses<br>Then the system should allow them to do so. |
| ***Priority:*** | Medium |

| ***ID*** | ***US25*** |
|---|---|
| ***Content:*** | As a student, I'd like to change the status of pending courses to completed or uncompleted after I get my grades. |
| ***Acceptance Criteria:*** | Given the student is in the Computer Science department<br>When the student wants to update their pending courses<br>Then the system should allow them to do so. |
| ***Priority:*** | Medium |

| ***ID*** | ***US26*** |
|---|---|
| ***Content:*** | As an admin, I'd like if students aren't allowed to change their progress in the middle of the semester. |
| ***Acceptance Criteria:*** | Given the student are taking a Computer Science course<br>When the students try to change the status of the course<br>Then the admin should set it to be allowed or not. |
| ***Priority:*** | Medium |

### 2.2.2 Non-Functional Requirements

Other requirements such as non-functional requirements such as:

- Reliability: The system should be available all the time.
- Scalable: system can hold a variable number of users.
- Maintainability: ease of repairing application bugs and adapting the web application to changed or new requirements
- Appearance: The user interface should be easy to use.
- Flexibility: The system should be flexible, so it is easy to add new features in the future.
- Security: The system should provide security for the user information.

### 2.2.3 UML Diagrams [4]

The following diagrams will help to clarify more about the system.

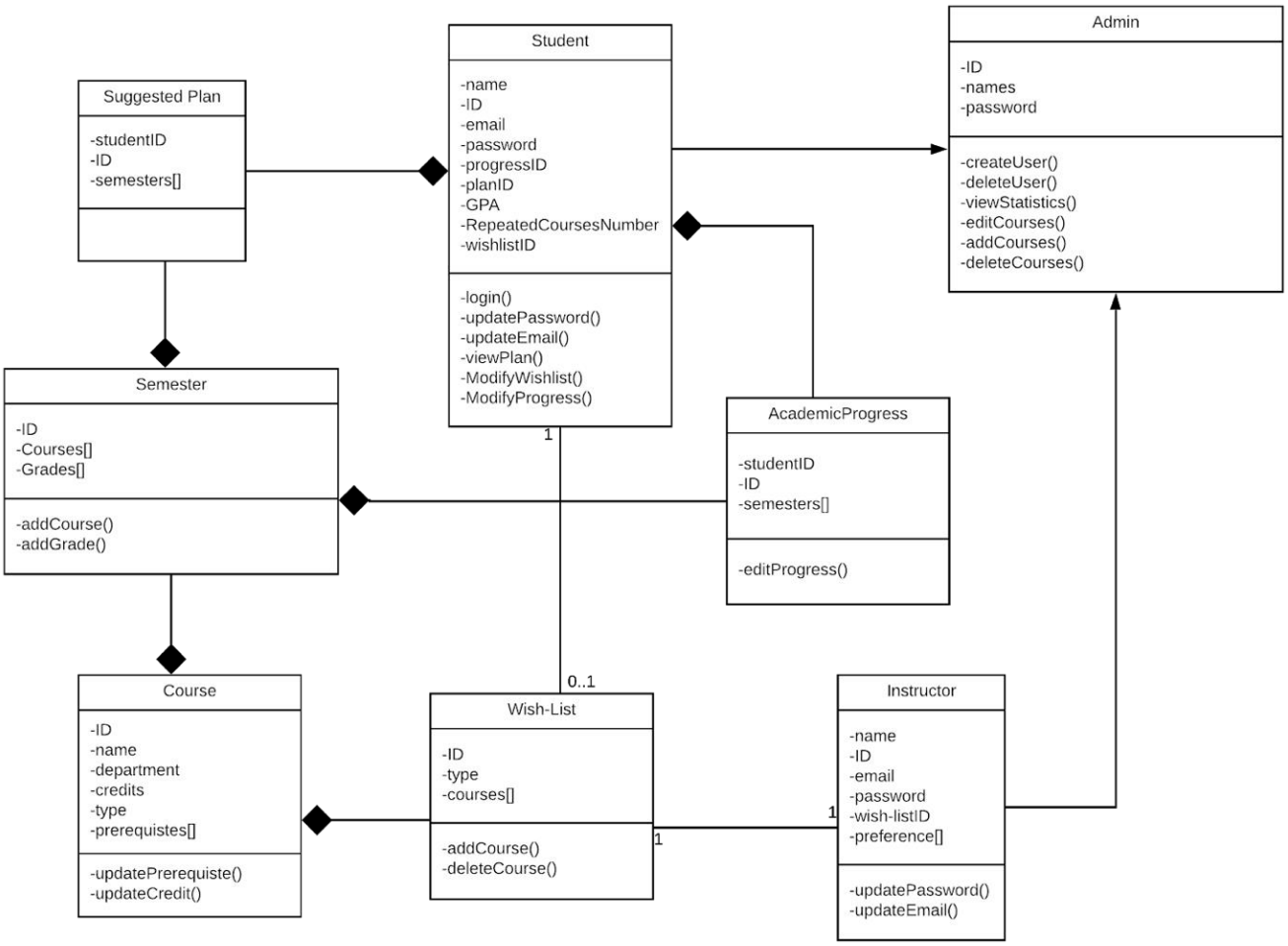


*Figure 1: Class diagram of the system*


*Figure 2:Use-case diagram for system major features*

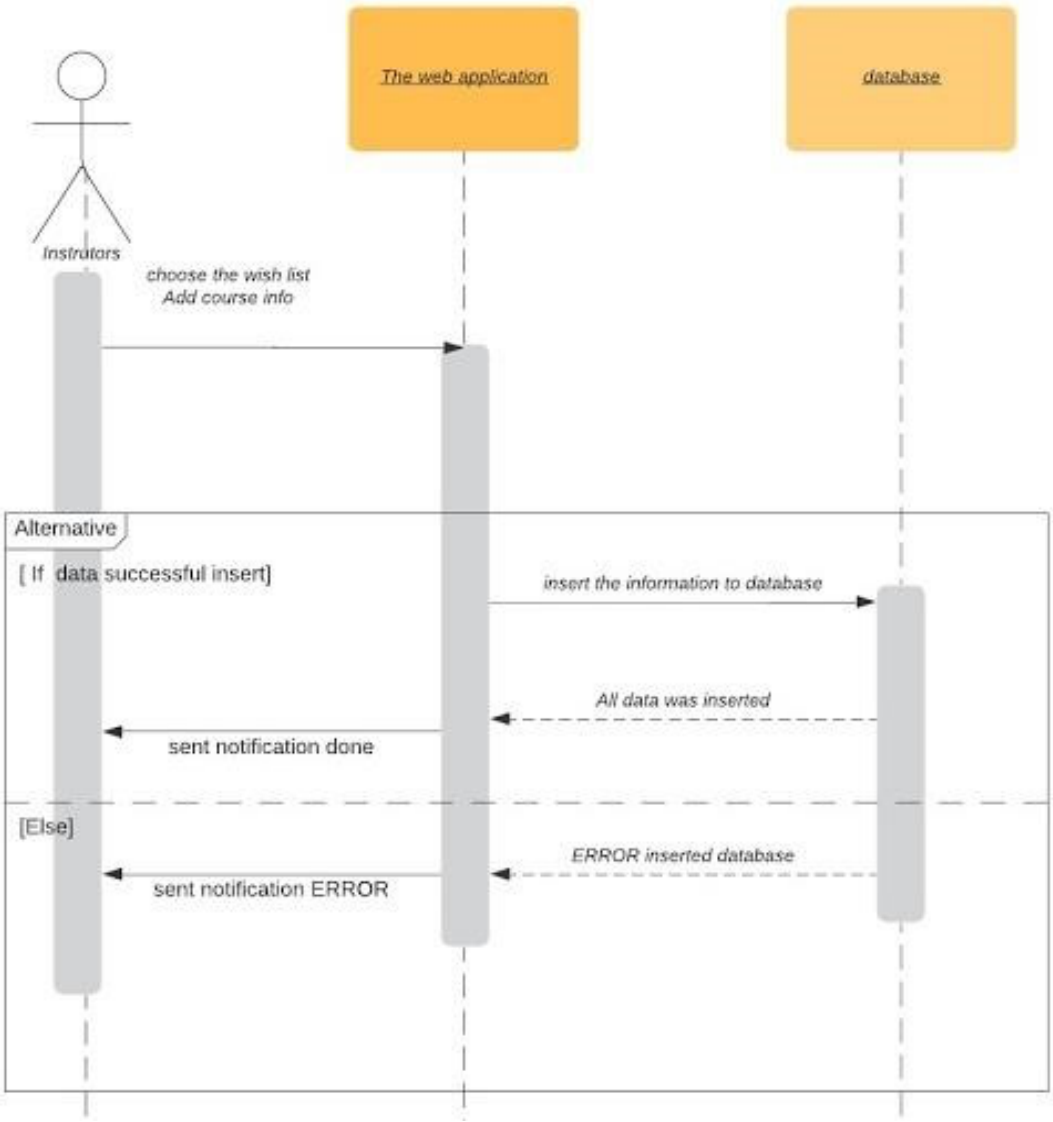


*Figure 3:Sequence diagram for an instructor adding a course to their wish-list*

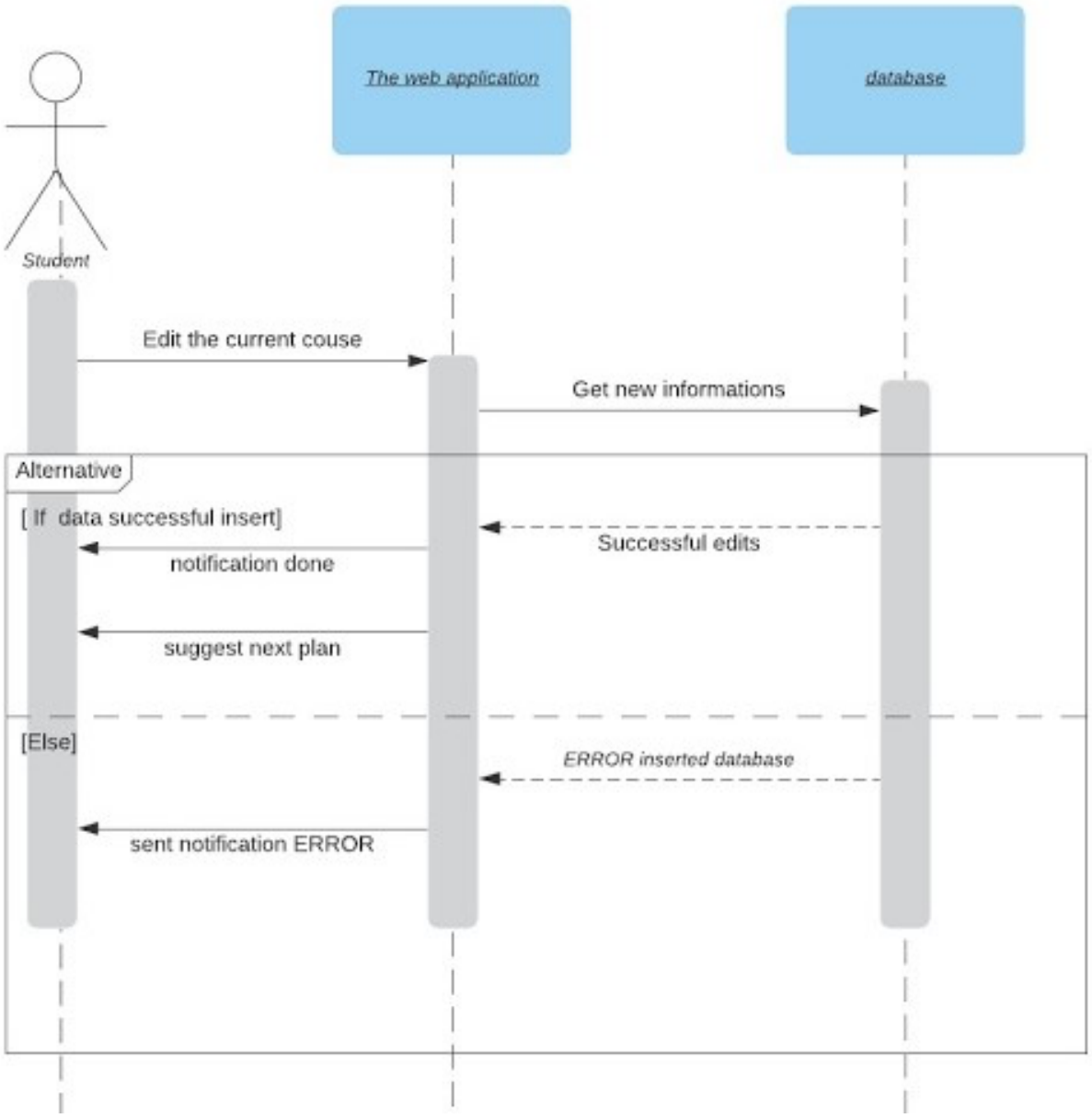


*Figure 4:Sequence diagram for a student to view advised plan next course*

# 3 System Architecture and Design

The system architecture representing the web-app follows the MTV design pattern (Model, Template, View) as it is supported by the Django framework.

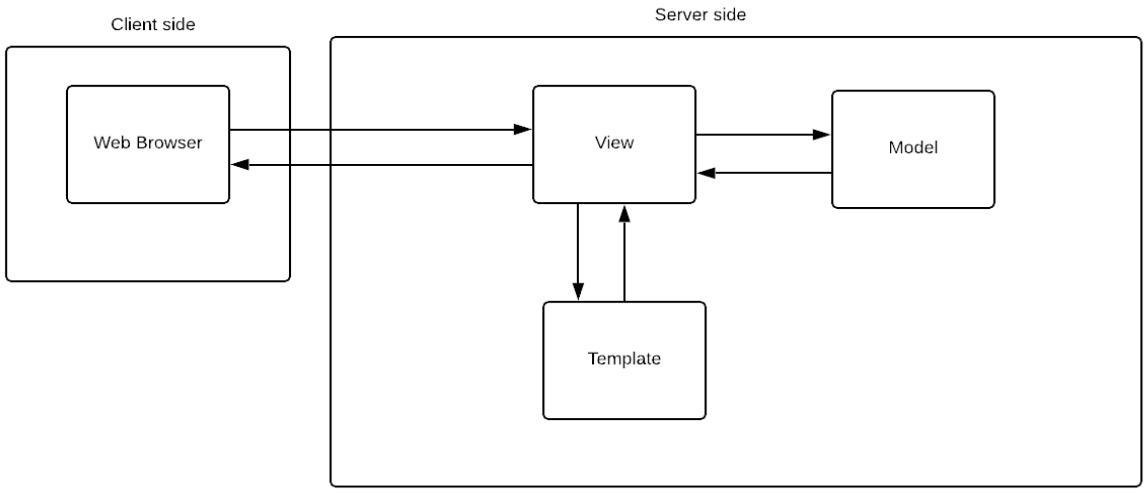


*Figure 5:Conceptual model of the system*

# 4 Developed Artifacts

## 4.1 Algorithm

The algorithm to handle the input of the system (visualized as the figure below) was developed after studying human decisions from interviews along with considering each inputs' unique characteristics (such as English department courses that's a different root, and no edges to connect to the CS department courses) and their relative importance to the overall progress.

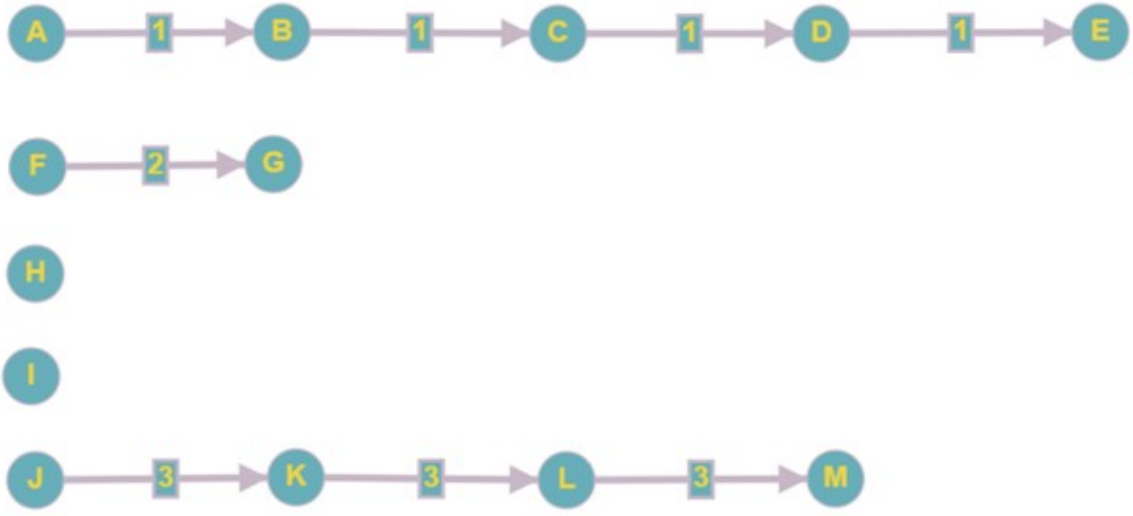


*Figure 6: Acyclic directed graph*

The type of vertices in the system:

1. Department compulsory courses

2. Department elective courses
3. College compulsory courses
4. College elective courses
5. University compulsory courses
6. University elective courses

Enforcing an order of importance for the course types along with keeping track of other considerations (i.e. passed prerequisites, max number of credits per semester) proved to be the best way to personalize the optimal plan and translated electronically.
The algorithm also followed other considerations elicited from interviews such as:

- Student preference of delaying university elective courses
- Student preference of delaying college elective courses

## 4.2 Users and Authorization

Different types of user roles are implemented in this system to include the real-time different types of audience members in the university's environment. With different types of user functions and permission.

| User role: | Authorization: |
|---|---|
| Student | The student can view the plan. And feed the system the input it needs to provide the optimal plan. While managing his or her profile, major sheet, wish list. |
| Instructor | The instructor can view statistics of courses such as the courses in demand. While adding to his or her wish list the courses they want to teach. With a special case of an instructor who can view all instructor choices. |
| Admin | Admin handles adding students by ID numbers, creating instructor accounts, and initializing the major sheet, setting dependencies of courses, and setting the wish-list and courses registration times. |

*Table 1:Table of user authorizations.*

## 4.3 Prototype

In the designing of interface, the team followed Schneiderman golden rules such as

- Consistency
- Enable users to short-cuts
- Offer informative feedback
- Reduce short-term memory load

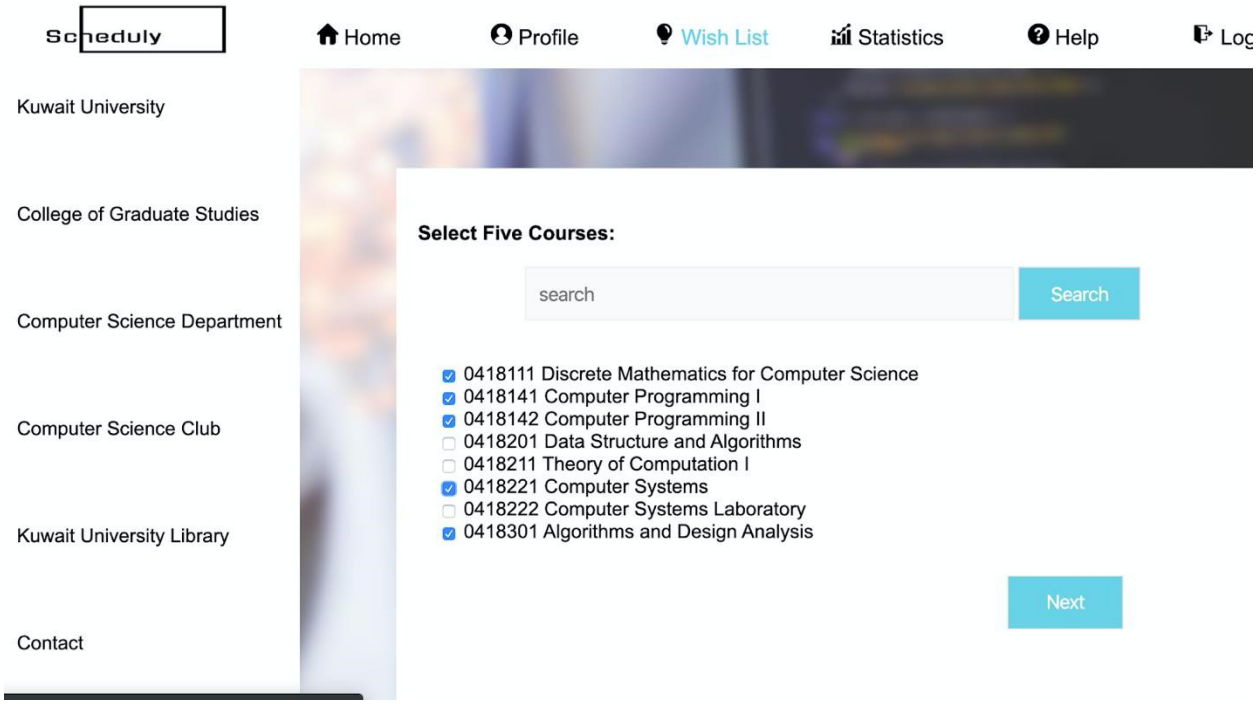


*Figure 7: Wish list page for an instructor*

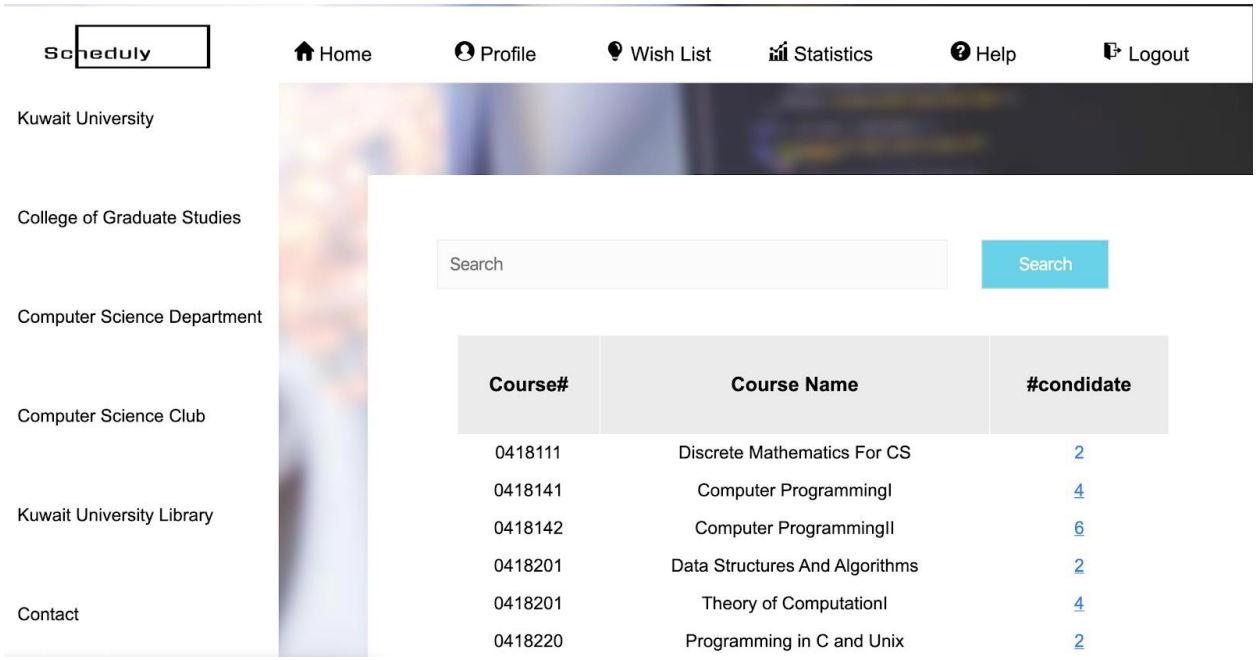


*Figure 8: The statistics page for instructors.*

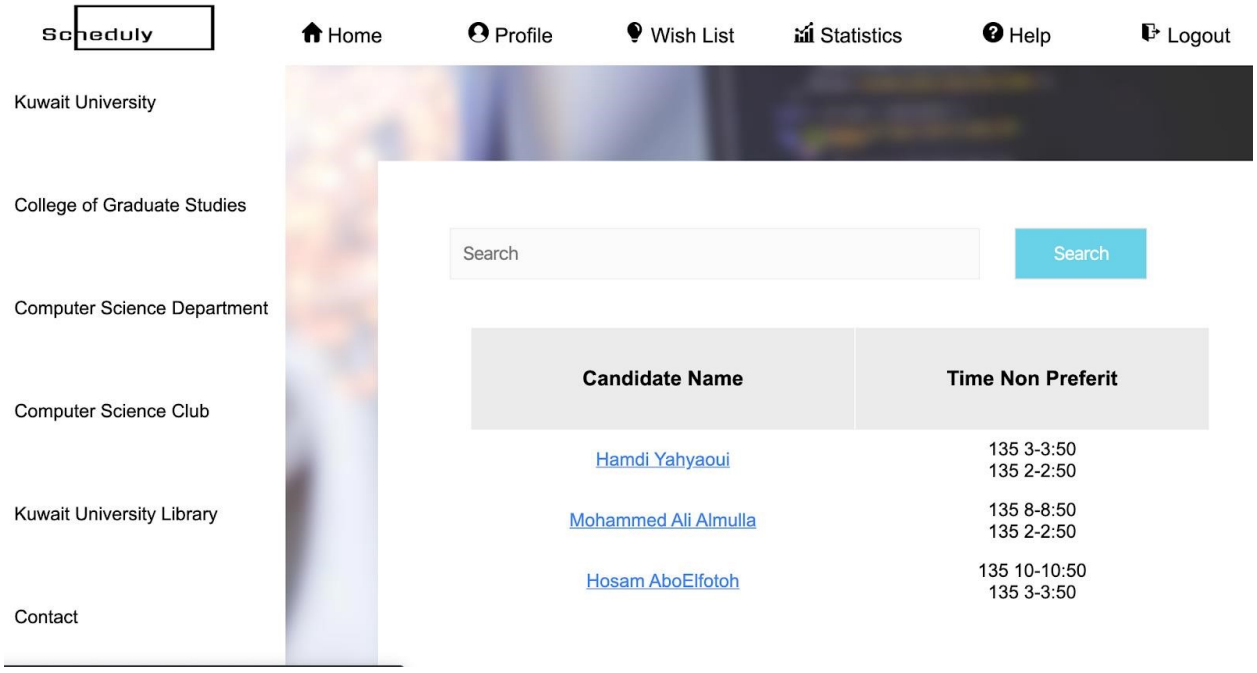


*Figure 9: Page to show non-preferred times of a special instructor.*

# 5 Current issues

## 5.1 Design Challenges

Design challenges were faced during the past phase included:

### 5.1.1 Invalid advised plan

A special case in the Computer Science department is that the major sheet has changed in the recent year. A design decision of handling only the new major sheet was made.

### 5.1.2 Deployment on the web

Choosing which hosting web platform to deploy the source artifacts according to this project. The team decided to use Amazon Web Services to host the system since it has python supported virtual machines and PostgreSQL.

### 5.1.3 Adapting to a new technology

Following the decision of changing the technology of implementation from the proposal, adapting to the new to the members technology Django framework was for team members.

### 5.1.4 Developing different versions

Having two team members and locally hosted development environments caused confusion and waste of effort.

## 5.2 Risk Management

### 5.2.1 Incompatible Hardware

This previous risk which was mentioned in the proposal has been managed by the way Django handles web-requests.

### 5.2.2 Security

This previous risk which was mentioned in the proposal has been reduced as Django has less security vulnerabilities (e.g. SQL injection, Cross-site scripting) than the previous choice PHP did.

### 5.2.3 Member withdrawal

The withdrawal of a team member resulted in the plan of work to change and assigned differently than intended while trying to keep the quality of the product from decreasing.

## 5.3 Newly Discovered Disks

### 5.3.1 Unable to Follow Previous Plan

The team members needed a different and more optimized plan to the team's new dynamic.

### 5.3.2 Loss of One More Team Member

The system will not be delivered if any more team members withdraw.

### 5.3.3 Database Damage

May lose the database before deploying on cloud servers for some reason such as disk crashes.

# 6 Plan of the Next Stage

## 6.1 Plan Modifications

Changes of previous plan included:

- Loss of a team member and more tasks divided.
- Different technology used and different pace of programming.
- Some requirements change such as:
    - The student advised plan cannot be modified by the student. o Introducing course registration and wish-list time.
    - The instructor is asked to enter two non-preferred time slots instead of a preferred one.

## 6.2 Updated Plan

The new plan of the project was updated to the following:

| **Task:** | **Status:** | **Task Owner:** |
|---|---|---|
| Requirement Gathering | Done | Asmaa, Amna |
| Requirement Analysis and validation | Done | Asmaa, Amna |
| User's Permissions | Done | Asmaa, Amna |
| Interface Design | Done | Amna |
| Interface Implementation | Done | Amna |
| Database Design | Done | Asmaa |
| Database Implementation | Not finished | Asmaa ,Amna |
| Main Algorithm Design | Done | Asmaa, Amna |
| Main Algorithm Implementation | Not finished | Asmaa |

| | | |
|---|---|---|
| Deployment on AWS | Not finished | Asmaa, Amna |
| AWS Configuration | Not finished | Amna |
| Populating | Not finished | Asmaa |
| Testing | Not finished | Asmaa, Amna |

*Table 2:Table for tasks for the project.*

# 7 Summary

Following major changes such as change in team members, change of technology used, and newly discovered requirements. The project plan changed as well as minor details in system features. But this project's main motive, that is to help the students and instructors of the Computer Science department have more fulfilled and smooth university experience, hasn't changed. Complete implementation is planned by the next phase along with deployment on the internet

# 8 References


1. Chakrabarti, A., Ghosh, P., McGregor, A. and Vorotnikova, S., 2020. Vertex ordering problems in directed graph streams. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (pp. 1786-1802). Society for Industrial and Applied Mathematics.
2. Elijah, J., Mishra, A., Udo, E.M.C., Abdulganiyu, A. and Musa, A., 2017. Survey on Requirement Elicitation Techniques: It's Effect on Software Engineering.
3. Bräuer, J., Plösch, R., Saft, M. and Körner, C., 2018. Measuring object-oriented design principles: The results of focus group-based research. *Journal of Systems and Software*, *140*, pp.74-90.
4. Software Engineering: A Practitioner's Approach, 8/e by Roger S. Pressman and Bruce R. Maxim
5. Correia, R. and Adachi, E., 2019, September. Detecting Design Violations in Djangobased Web Applications. In *Proceedings of the XIII Brazilian Symposium on Software Components, Architectures, and Reuse* (pp. 33-42).