Schedule Optimizer

Use Case Model

Submitted to:

Prof. Ma. Rowena C. Solamo
Faculty Member
Department of Computer Science
College of Engineering
University of the Philippines, Diliman

Submitted by:
Cavan, Antonio D.
Ramos, John Matthew G.
See, Engelberg Jeremy T.

In partial fulfillment of Academic Requirements
for the course
CS 191 Software Engineering I
of the
1st Semester, AY 2019-2020

Unique Reference:

The documents are stored in the https://github.com/DarkLuminosity/Schedule-Optimizer/tree/master/02-Requirements%20Engineering referenced with Group 3-Schedule Optimizer - Use Case Model.pdf

Document Purpose:

In order to have a thorough Requirements Analysis, the group will utilize the Use Case Model which will be explicitly discussed in this document.

Target Audience:

The target audience of this document would be UP Diliman undergraduate students.

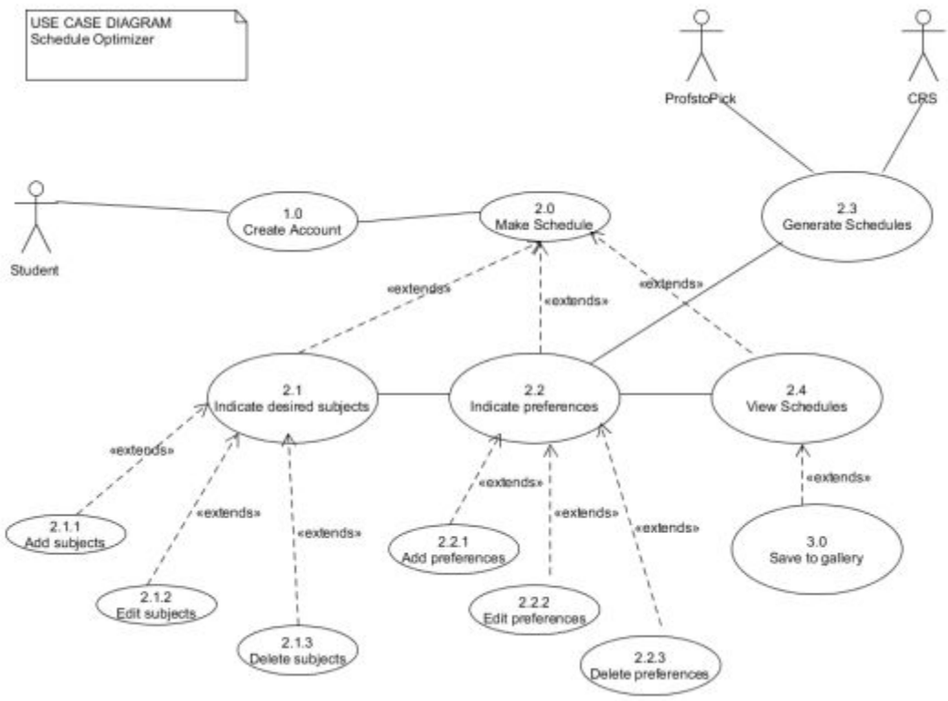
Revision Control:

Revision Date	*Person Responsible*	*Version Number*	*Contribution and Modification*
09/11/19	Engelberg See	1.0	Created Initial Document
09/16/19	Engelberg See	2.0	Added Use-Case Diagram, List of Actors and List of Use-Cases, Target Audience and Description
09/16/19	Antonio Cavan	3.0	Added Document Purpose and Description of system. Edited Use Case Description Table (i.e. Edit Subjects, Delete subjects, Edit Preferences, Delete Preferences, View schedules, Save to Gallery)

***System Name*:** Schedule Optimizer

***Description*:** The Schedule Optimizer is an application that creates a schedule of UP classes that is optimized according to the student's preference. There are three actors that will interact with the system, mainly: Student, ProfstoPick, and CRS. The Student will be the main consumer of the application as he or she will provide the classes to be sorted and the preferences that would factor in the computation. Furthermore, the student will be able to keep track of the schedules taken for each semester. The ProfstoPick will be providing numerical data of professor ratings. Lastly, the CRS will be providing the list of classes from which the system will be picking from.

***Use-Case Diagram*:**



List of Actors:

Actors	*Description*
Student	The student is the main student of the system. It plays largely in the role of indicating their classes and preferences to create schedules based on them.
ProfstoPick	ProfstoPick contains the ratings of the different profs teaching in UP Diliman. It plays the role of generating schedules based on the ratings of the profs.
CRS	CRS contains the list of available classes that will be offered in UP Diliman at a specific time. It plays the role of generating schedules based on the availability of classes.

List of Use-cases:

Use-Case	*Description*
Use-Case 1.0 Create Account	The students will create an account in order to utilize the system efficiently. This would include saving the username and password to keep track of the different students in the system. This would also enable the student to store the previously and presently made schedules.
Use-Case 2.0 Make Schedule	This is the main functionality of the system. This would create the schedule based on the student's schedules and preferences. It would then depend on the two actors, ProfstoPick and CRS, to generate the schedule afterwards.
Use-Case 2.1 Indicate desired Subjects	This functionality would allow students to indicate their desired subjects. It would have a list of subjects in which the students have to choose. It has options such as adding, deleting as well as editing the subjects.
Use-Case 2.1.1 Add Subjects	This functionality would allow students to add subjects that they need based on their schedules. It would contain the list of subjects being offered in UP Diliman. By adding the subjects, the student has indicated that these will be used in the creation of their schedule
Use-Case 2.1.2 Edit Subjects	This functionality would allow students to change the list of classes they initially chose. Editing the subjects gives the system more flexibility. Since the system automatically arranges the classes from highest hierarchy to lowest, students may want to change the subject on a certain slot without affecting the hierarchy of those below it.
Use-Case 2.1.3 Delete Subjects	This functionality would allow students to delete subjects that they do not desire in the case that they had added them earlier. This option would select from the list of added subjects that the student has made earlier. By deleting the unwanted subjects, subjects at the lower level will rise up the hierarchy.
Use-Case 2.2 Indicate Preferences	This functionality would allow students to indicate the preferences they want in the creation of their schedule. It would include preferences such as choosing from having the rating of the professors, time, etc. It also has options where the student could add, edit or delete their preferences.
Use-Case 2.2.1 Add Preferences	This functionality would allow students to add the preferences they want in the creation of their schedule. It would include the options of choosing from the ratings of the professors, time, etc. By adding the preferences, the student would have indicated what they want the most in the creation of their schedule.
Use-Case 2.2.2 Edit Preferences	This functionality would allow students to change specific preferences that previously chose. This adds flexibility to the system as well. By editing the preferences, students would be able to change factors (e.g. Pedagogy to Easiness) or change their priority level without redoing the entire process.
Use-Case 2.2.3 Delete Preferences	This functionality would allow students to delete the preferences that they do not want. This would include the list of preferences that they had selected. By deleting preferences, the system will automatically move up preferences below the hierarchy.
Use-Case 2.3 Generate Schedules	This functionality would allow the system to generate schedules based on

	the student's subjects and preferences. There would be an algorithm that would take those factors into account, making an optimized schedule from their preferences. The data that will be used in this use-case would be acquired from CRS and ProfstoPick.
Use-Case 2.4 View Schedules	This functionality would allow students to view the schedules that the system had generated. It includes the total scores of the schedule in terms of the preferences which would be the student in choosing their subjects. This would lead to the student having an optimized schedule from the classes and preferences that they had specified..
Use-Case 3.0 Save to Gallery	This functionality would allow students to save their chosen schedule to the gallery of the device wherein the system is used. The layout of the image will be exactly what would be seen in the app. This is important as it gives the students to share their schedule to their friends or back it up in case of device failure or loss.