Schedule Optimizer
Analysis Model

Submitted to:

Asst. 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/Project%20Deliverables referenced with Group 3-Schedule Optimizer-Analysis Model.

Purpose:

The purpose of this document is to analyze the different scenarios that may occur within the development of the software as well as finding the required or potential functionalities of the project.

Audience:

The audience of this paper would be UP Diliman undergraduate students.

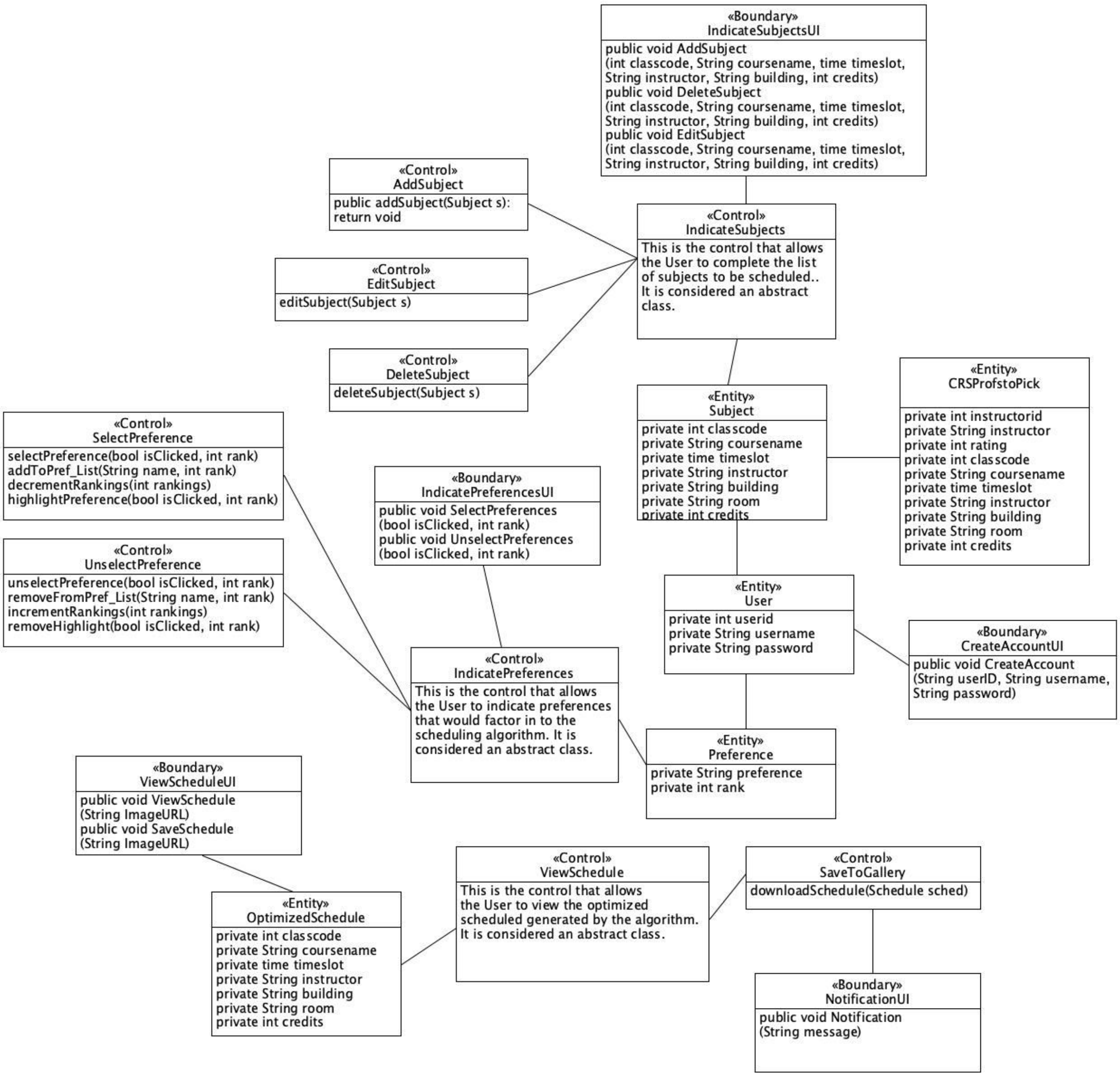
Revision Control:

Revision Date	***Person Responsible***	***Version Number***	***Modification***
09/26/19	Engelberg See	1.0	Prepared Initial Document
10/2/19	Antonio Cavan	2.0	Added System Description
10/02/19	Engelberg See	3.0	Added Entity Classes, Use-cases Name and Description
10/03/19	Antonio Cavan	4.0	Added Control Classes
10/03/19	Matthew Ramos	5.0	Added Boundary Classes
10/04/19	Engelberg See	6.0	Added Behavior Model for Use-Case Indicate Subject
10/04/19	Antonio Cavan	7.0	Added Behavior Models for Use-Case Indicate Preference
10/04/19	Antonio Cavan	7.1	Edited Control classes Add, Edit, Delete Preference to Select and Unselect Preferences.
10/04/19	Matthew Ramos	8.0	Added Behavior Model for Use-Case View Schedule, Analysis Model
10/04/19	Antonio Cavan	8.1	Edited Responsibilities and functions of boundary, control and entities related to Indicate Preference Use Case

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.

Analysis Model:


«Boundary»
IndicateSubjectsUI
public void AddSubject
(int classcode, String coursename, time timeslot, String instructor, String building, int credits)
public void DeleteSubject
(int classcode, String coursename, time timeslot, String instructor, String building, int credits)
public void EditSubject
(int classcode, String coursename, time timeslot, String instructor, String building, int credits)
«Control»
AddSubject
public addSubject(Subject s): return void
«Control»
EditSubject
editSubject(Subject s)
«Control»
DeleteSubject
deleteSubject(Subject s)
«Control»
IndicateSubjects
This is the control that allows the User to complete the list of subjects to be scheduled.. It is considered an abstract class.
«Entity»
Subject
private int classcode
private String coursename
private time timeslot
private String instructor
private String building
private String room
private int credits
«Entity»
CRSProfstoPick
private int instructorid
private String instructor
private int rating
private int classcode
private String coursename
private time timeslot
private String instructor
private String building
private String room
private int credits
«Control»
SelectPreference
selectPreference(bool isClicked, int rank)
addToPref_List(String name, int rank)
decrementRankings(int rankings)
highlightPreference(bool isClicked, int rank)
«Control»
UnselectPreference
unselectPreference(bool isClicked, int rank)
removeFromPref_List(String name, int rank)
incrementRankings(int rankings)
removeHighlight(bool isClicked, int rank)
«Boundary»
IndicatePreferencesUI
public void SelectPreferences
(bool isClicked, int rank)
public void UnselectPreferences
(bool isClicked, int rank)
«Entity»
User
private int userid
private String username
private String password
«Boundary»
CreateAccountUI
public void CreateAccount
(String userID, String username, String password)
«Control»
IndicatePreferences
This is the control that allows the User to indicate preferences that would factor in to the scheduling algorithm. It is considered an abstract class.
«Entity»
Preference
private String preference
private int rank
«Boundary»
ViewScheduleUI
public void ViewSchedule
(String ImageURL)
public void SaveSchedule
(String ImageURL)
«Entity»
OptimizedSchedule
private int classcode
private String coursename
private time timeslot
private String instructor
private String building
private String room
private int credits
«Control»
ViewSchedule
This is the control that allows the User to view the optimized scheduled generated by the algorithm. It is considered an abstract class.
«Control»
SaveToGallery
downloadSchedule(Schedule sched)
«Boundary»
NotificationUI
public void Notification
(String message)

Boundary Classes:

Class Name	**Description**
CreateAccountUI	This is the interface of the user in using the system for creating the user's account.. *Responsibilities*: public void CreateAccount (String userID, String username, String password)
IndicateSubjectsUI	This is the interface of the user in using the system for adding the user's subjects. *Responsibilities*: public void AddSubject (int classcode, String coursename, time timeslot, String instructor, String building, int credits) public void DeleteSubject (int classcode, String coursename, time timeslot, String instructor, String building, int credits) public void EditSubject (int classcode, String coursename, time timeslot, String instructor, String building, int credits)
IndicatePreferencesUI	This is the interface of the user in using the system for adding the user's preferences for the user's optimized schedule. *Responsibilities*: public void SelectPreferences (bool isClicked, int rank) public void UnselectPreferences (bool isClicked, int rank) public void incrementRankings(int rankings) public void decrementRankings(int rankings) public void removeHighlight(bool isClicked, int rank) public void highlightPreference(bool isClicked, int rank)
ViewScheduleUI	This is the interface of the user in using the system for viewing the user's optimized schedule within the app or choosing to save to their phone's gallery. *Responsibilities*: public void ViewSchedule (String ImageURL) public void SaveSchedule (String ImageURL)
NotificationUI	The is the interface presented when a problem occurs. *Responsibilities* public void Notification (String message)

Control Classes:

Class Name	**Description**
IndicateSubjects	This is the control that allows the User to complete the list of subjects to be scheduled.. It is considered an abstract class.
AddSubject	This is the control that adds a subject to the list of subjects. It extends IndicateSubjects. *Responsibilities*: public addSubject(Subject s): return void
EditSubject	This is the control that edits a subject from the list of subjects. It extends IndicateSubjects. *Responsibilities*: editSubject(Subject s)
DeleteSubject	This is the control that deletes a subject from the list of subjects. It extends IndicateSubjects. *Responsibilities*: deleteSubject(Subject s)
IndicatePreferences	This is the control that allows the User to indicate preferences that would factor in to the scheduling algorithm. It is considered an abstract class.
SelectPreference	This is the control that adds a preference to the list of preferences. It extends IndicatePreferences. *Responsibilities*: selectPreference(bool isClicked, int rank)
UnselectPreference	This is the control that removes a preference from the list of preferences It extends IndicatePreferences. *Responsibilities*: unselectPreference(bool isClicked, int rank)
ViewSchedule	This is the control that allows the User to view the optimized scheduled generated by the algorithm. It is considered an abstract class.
SaveToGallery	This is the control that allows the User to save a screenshot of the generated schedule to their gallery. It extends ViewSchedule. *Responsibilities*: downloadSchedule(Schedule sched)

Entity Classes:

Class Name	**Description**
User	This is the entity class User, which contains the data about the user *Attributes*:

	private int userid private String username private String password
Subject	This is the entity class Subject, which contains the data about the subject *Attributes*: private int classcode private String coursename private time timeslot private String instructor private String building private String room private int credits *Functions*: public AddSubject (String coursename) public EditSubject (String coursename) public DeleteSubject (String coursename)
PreferenceList	This is the entity class Preference, which contains the data about the preference. *Attributes*: private String preference private int rank *Functions*: public removeFromPref_List(String name, int rank) public addToPref_List(String name, int rank)
CRSProfstoPick	This is the entity class CRSProfstoPick, which contains the data about the Professors and the classes that he or she teaches. *Attributes*: private int instructorid private String instructor private int rating private int classcode private String coursename private time timeslot private String instructor private String building private String room private int credits

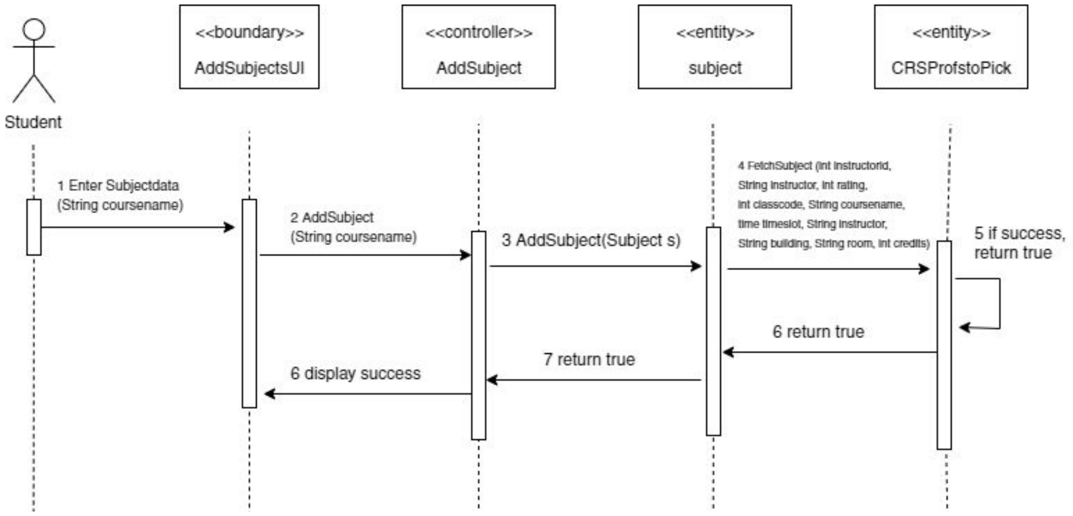
	Functions: public FetchSubject (int instructorid, String instructor, int rating, int classcode, String coursename, time timeslot, String instructor, String building, String room, int credits)
OptimizedSchedule	This is the entity class OptimizedSchedule, which contains the data about the optimized schedule *Attributes*: private int classcode private String coursename private time timeslot private String instructor private String building private String room private int credits

Behavioral Model:

***Use-Case Name*:** Indicate Subject

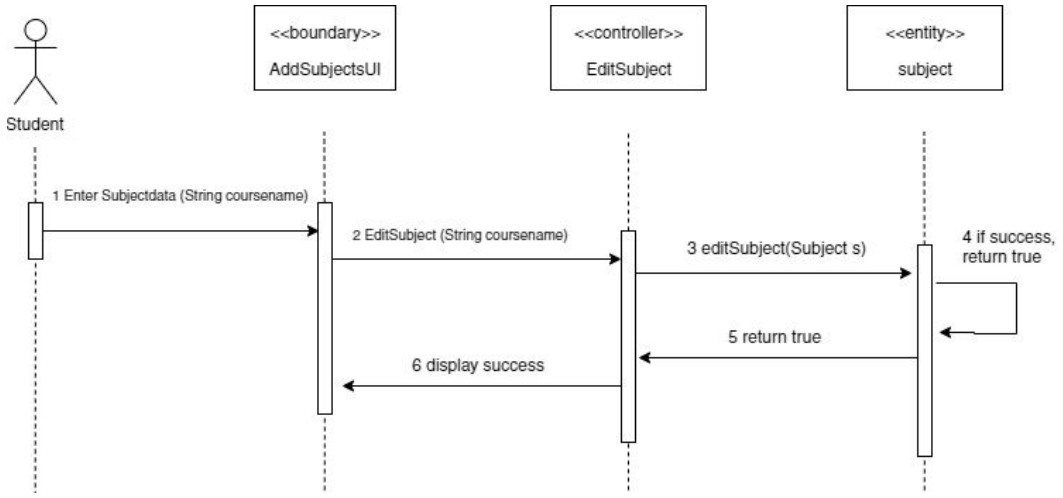
Description: Indicate Subject is a use case in which the user would select, edit or delete the subjects that would be used for the succeeding use cases. It would contain a list of subjects being offered in UP Diliman wherein he or she would input the different subjects that he or she would want. The user could also modify the hierarchy of subjects that they have chosen to prioritize which subjects are the main points of their schedule.

Scenario 1: User wishes to indicate the subjects that he or she would want to see on his or her schedule. (Basic Flow)

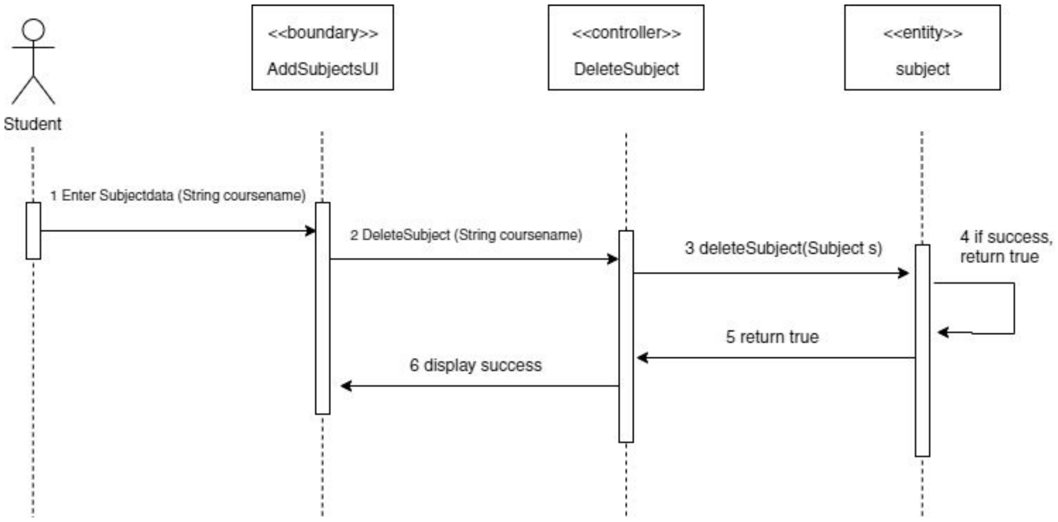


Student
<<boundary>>
AddSubjectsUI
<<controller>>
AddSubject
<<entity>>
subject
<<entity>>
CRSProfstoPick
1 Enter Subjectdata (String coursename)
2 AddSubject (String coursename)
3 AddSubject(Subject s)
4 FetchSubject (int instructorid, String instructor, int rating, int classcode, String coursename, time timeslot, String instructor, String building, String room, int credits)
5 if success, return true
6 return true
7 return true
6 display success

Scenario 2: User wishes to edit the hierarchy of the subjects that they had selected. (Alternative Flow)

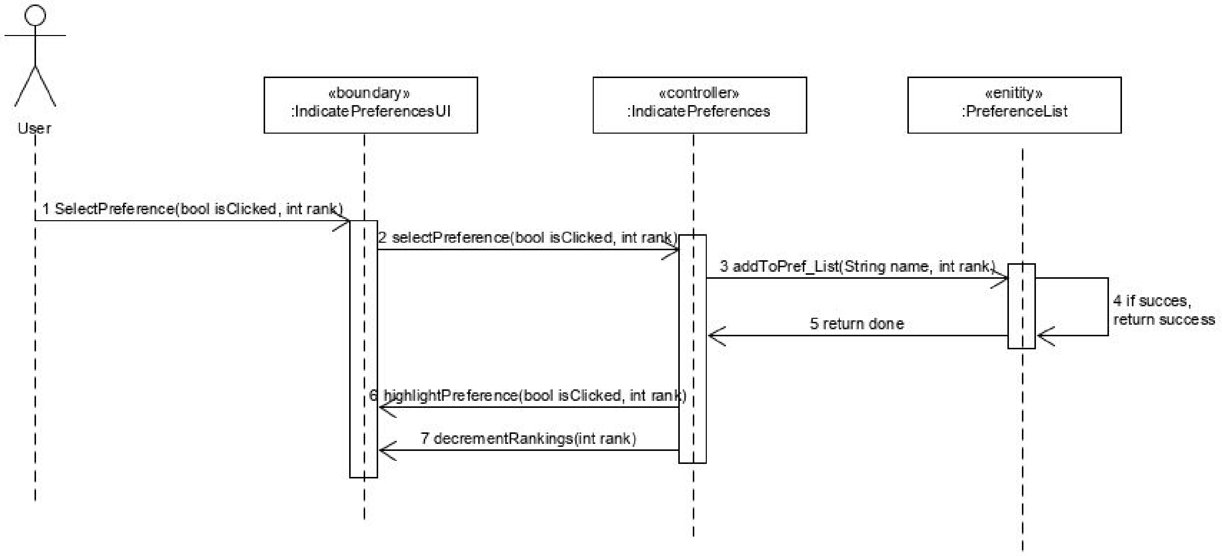


Scenario 3: User wishes to delete a subject that they had misclicked. (Alternative Flow)

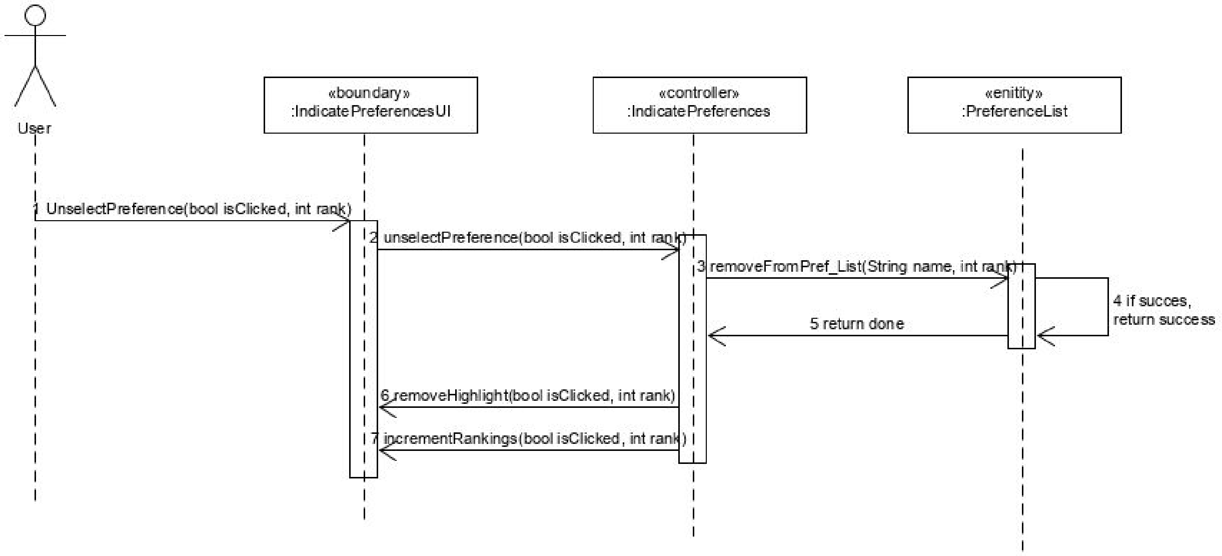

<<boundary>>
AddSubjectsUI
<<controller>>
DeleteSubject
<<entity>>
subject
Student
1 Enter Subjectdata (String coursename)
2 DeleteSubject (String coursename)
3 deleteSubject(Subject s)
4 if success, return true
5 return true
6 display success

***Use-Case Name*:**	Indicate Preference
Description:	This function of the Schedule Optimizer is to set the necessary variables that would factor in the scheduling algorithm. Indicate Preference is a prerequisite in creating an optimized schedule, i.e the software will not be able to process a request if the preferences are left blank. The User may select three preferences ranked from first, second and third. These ranking reflects the priority of the preference and will factor in the scheduling algorithm. While in the Indicate Preference Case, the user may select or unselect preferences.

Scenario 1: User selects three preferences with respective priorities. (Basic Flow)


User
«boundary»
:IndicatePreferencesUI
«controller»
:IndicatePreferences
«enitity»
:PreferenceList
1 SelectPreference(bool isClicked, int rank)
2 selectPreference(bool isClicked, int rank)
3 addToPref_List(String name, int rank)
4 if succes,
return success
5 return done
6 highlightPreference(bool isClicked, int rank)
7 decrementRankings(int rank)

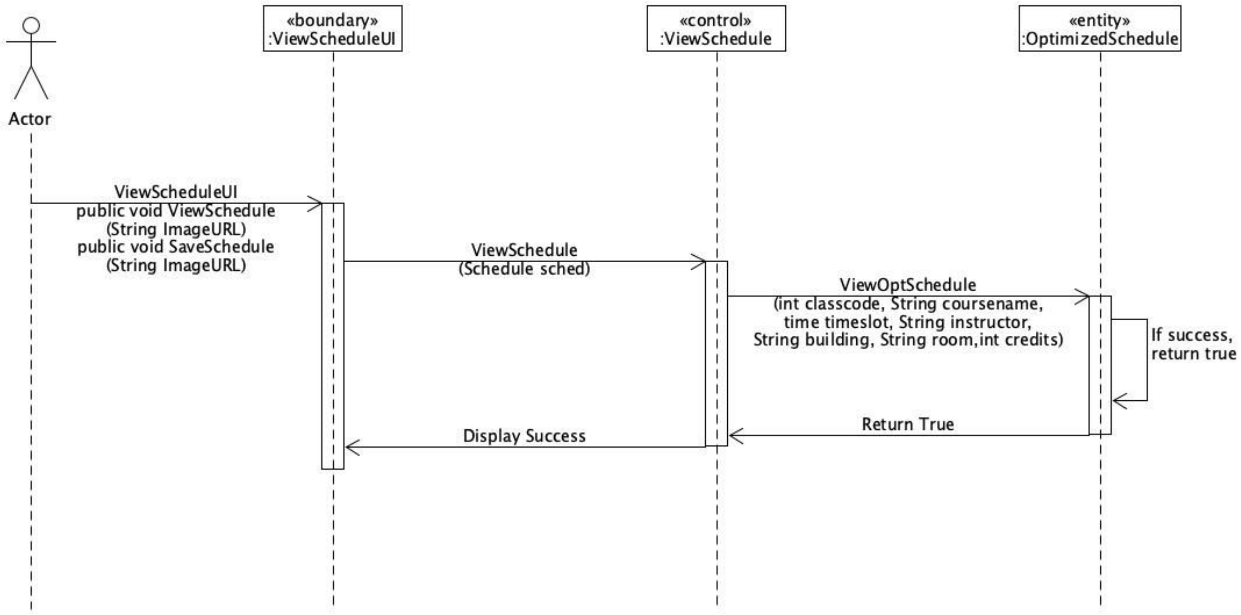
Scenario 3: User wishes to unselect a Preference. (Alternative Flow)


User
«boundary»
:IndicatePreferencesUI
«controller»
:IndicatePreferences
«enitity»
:PreferenceList
1 UnselectPreference(bool isClicked, int rank)
2 unselectPreference(bool isClicked, int rank)
3 removeFromPref_List(String name, int rank)
4 if succes,
return success
5 return done
6 removeHighlight(bool isClicked, int rank)
7 incrementRankings(bool isClicked, int rank)

***Use-Case Name*:** View Schedules

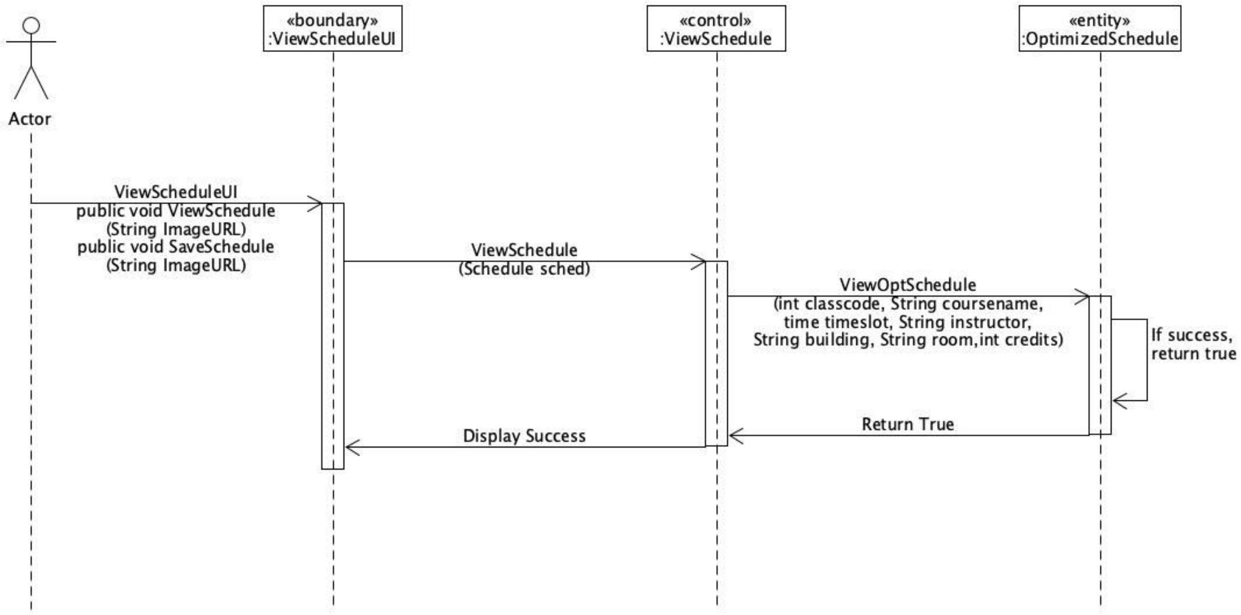
Description: The role of this use case would be to allow students to view their schedule within the app or save their schedule to the gallery of their device. The layout of the saved image will be exactly what would be seen in the app. This is important as it gives the students to share their schedule to their friends, back it up in case of device failure or loss, or use as a wallpaper as this is something that most students do.

Scenario 1: User wishes to view their schedule within the app. (Basic Flow)
uc



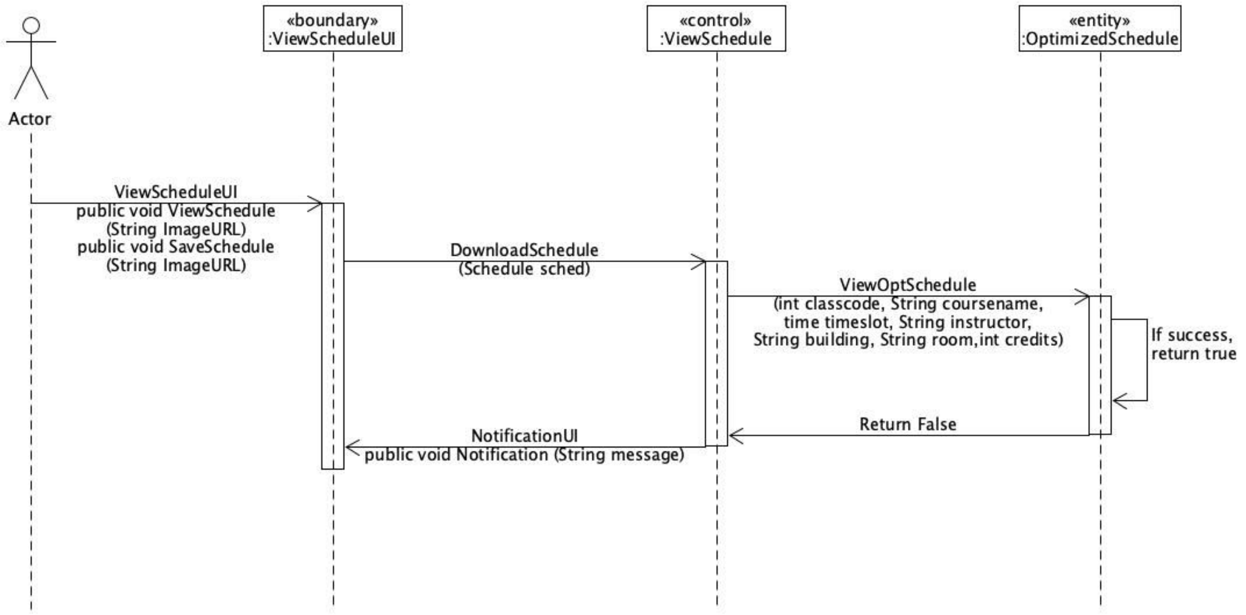
Scenario 2: User wishes to save their schedule to the gallery on their phone. (Alternative Flow)

[Sequence Diagram of Scenario 2]



Scenario 3: User wishes to save their schedule to the gallery on their phone but there is insufficient storage. (Alternative Flow)

[Sequence Diagram of Scenario 3]


Actor
«boundary»
:ViewScheduleUI
«control»
:ViewSchedule
«entity»
:OptimizedSchedule
ViewScheduleUI
public void ViewSchedule
(String ImageURL)
public void SaveSchedule
(String ImageURL)
DownloadSchedule
(Schedule sched)
ViewOptSchedule
(int classcode, String coursename,
time timeslot, String instructor,
String building, String room,int credits)
If success,
return true
Return False
NotificationUI
public void Notification (String message)