5. Dynamic Design Scenarios

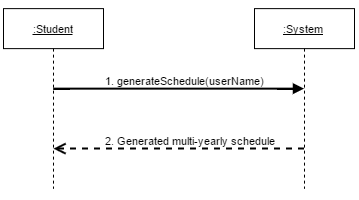
5.1 Generate Schedule

*Generate schedule*, as the name says, consists of creating a schedule for a student actor by using his/her preferences, list of taken classes and list of needed classes.

**5.1.1 Full Use Case**

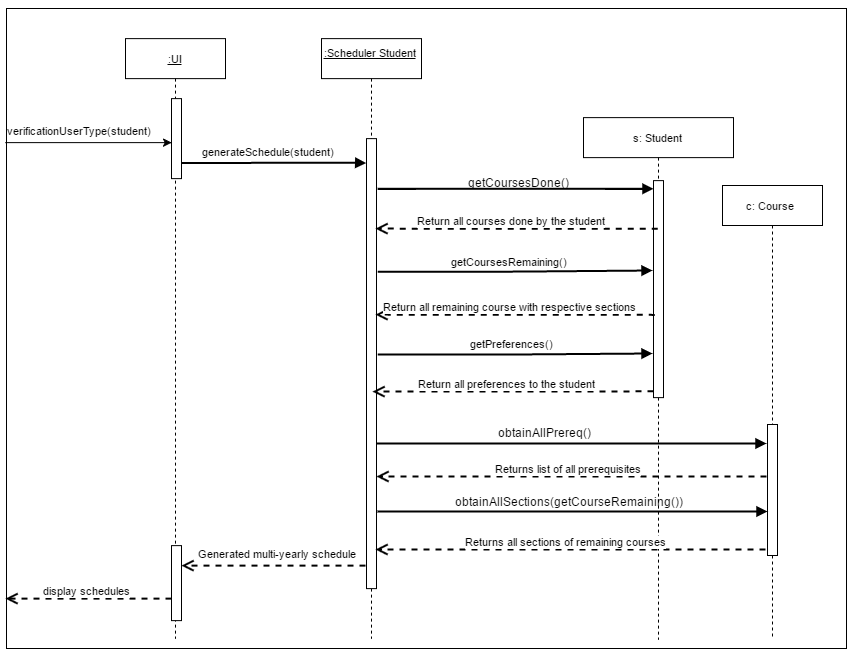
|  |  |  |  |
| --- | --- | --- | --- |
| Name: | Generate Schedule | Author | Ying-Chen Chu |
| Identifier: | UC13 | Version: | 3.0 |
| Date Created: | 2015-02-03 | Last Modified: | April 12, 2016 |
| Importance: | 5/5 | | |
| Actor(s): | Student, Administrator | | |
| Goal: | Generate a schedule | | |
| Summary: | Based on the course selection list and or without the list of preferences, the system generates schedule(s) for the selected courses. | | |
| Related use-cases: | UC8, UC9, UC12 | | |
| Preconditions | 1. User has been authenticated.  2. Schedule preferences already set. | | |
| Trigger: | User activates the "Generate Schedule" process | | |
| Basic Flow: | 1. User indicates the wish to generate a schedule.by selecting the “Build Schedule” option.  ~~2. System verifies entries in lists of taken courses, needed courses & schedule preferences saved to the system via UC8, UC9 & UC11.~~  3 System redirects user to schedule page & displays schedule. | | |
| Post-Conditions: | **Success:** Schedule with the selected courses is generated. | | |
| Minimum Guarantee: | Previous state of the system remains the same.  **Failure:** System fails to generate a schedule and an error message displays | | |
| Risk Assessment: | High | | |

**5.1.2 System Sequence Diagram**



5.1.2.1 generateSchedule*(userName)* Operating Contracts & Respective Sequence Diagram

|  |  |
| --- | --- |
| **Name** | Contract 1.1 Generate Schedule |
| **Operation** | generateSchedule(String) |
| **Cross Reference** | UC13 Generate Schedule |
| **Pre conditions** | -User is logged into the system  -User has set needed/taken courses, and preferences |
| **Post conditions** | N/A |

Sequence Diagram for contract 1.1

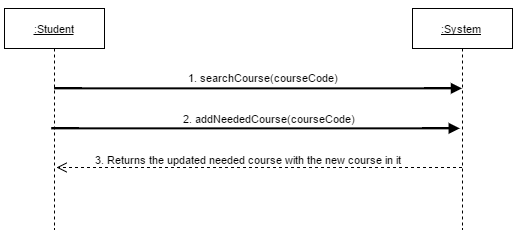
5.2 Set Needed Course

*Set needed course* consists of adding a course the student wants to the list. The course has to be first searched. If it is found, then it can be added to the list.

**5.2.1 Full Use Case**

|  |  |  |  |
| --- | --- | --- | --- |
| Name: | Set Needed Courses | Author: | Ying-Chen Chu |
| Identifier: | UC9 | Version: | 2.0 |
| Date Created: | 2015-02-03 | Last Modified: | April 12, 2016 |
| Importance: | 5/5 | | |
| Actor(s): | Student, Administrator | | |
| Goal: | Add a needed course to the schedule | | |
| Summary: | Select a course to be added to the list of courses for the schedule generator. | | |
| Related use-cases: | - | | |
| Preconditions | 1. User has been authenticated  2. Course requirements are met | | |
| Trigger: | User selects the set needed course process. | | |
| Basic Flow: | 1. User indicates the wish to add a course by selecting a course from the needed courses list.  2. User commits by selecting the “add class” option.  ~~3. System checks for eligibility and time conflicts.~~  ~~4. System generates corresponding course name/number to entry if either is missing~~.  3. System adds course to list of needed courses. | | |
| Post-Conditions: | **Success:** Course is added to the user's list of courses | | |
| Minimum Guarantee: | Previous state of the system remains unchanged.  **Failure:** System fails to process task and displays an error message. | | |
| Risk Assessment: | Low | | |

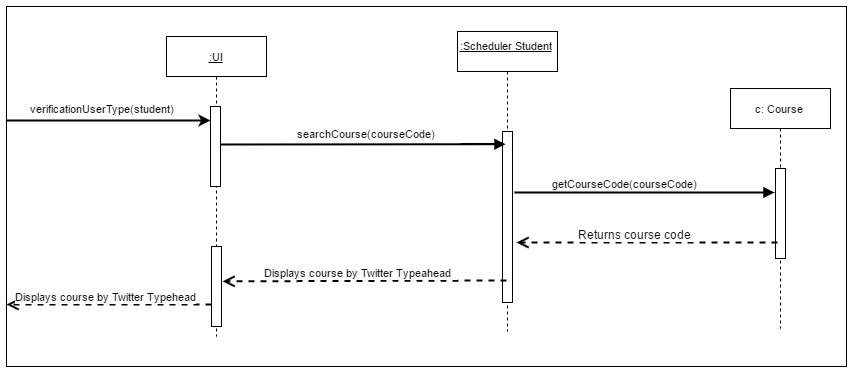
**5.2.2 System Sequence Diagram**



5.2.2.1 *searchCourse(courseCode)* Operating Contracts

|  |  |
| --- | --- |
| **Name** | Contract 2.1 Search course in database |
| **Operation** | searchCourse(String) |
| **Cross Reference** | UC9 Set Needed Courses |
| **Preconditions** | -User is logged in. |
| **Postconditions** | -Association is made between inputted course code and course code inside the system. |

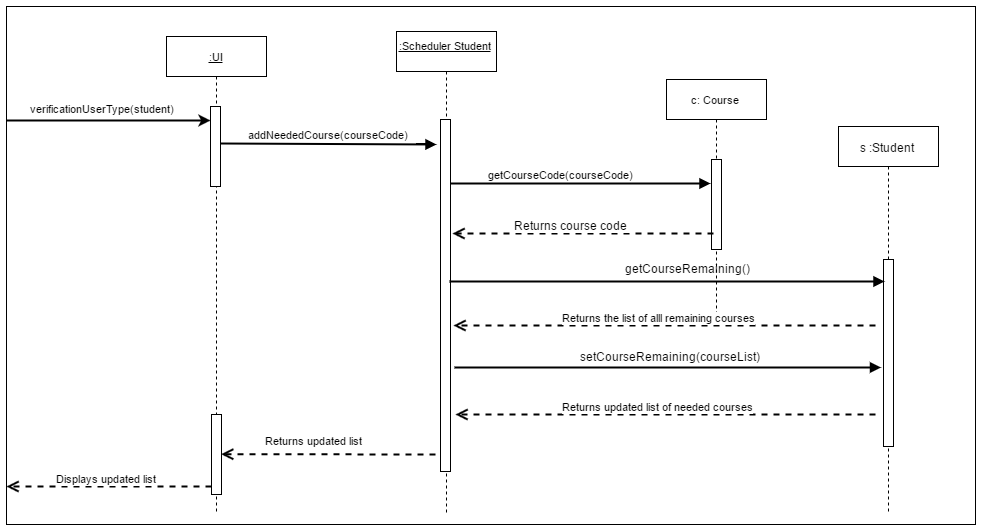
Sequence diagram for contract 2.1



5.2.2.2 *addNeededCourse(courseCode)* Operating Contracts

|  |  |
| --- | --- |
| **Name** | Contract 2.2 Add needed course |
| **Operation** | addNeededCourse(String) |
| **Cross Reference** | UC9 Set Needed Courses |
| **Pre conditions** | -User is logged-in into the system  -Inputted course exists into the system |
| **Post conditions** | -Instance of course has been created and added to the array of needed course of the student. |

Sequence diagram for contract 2.2



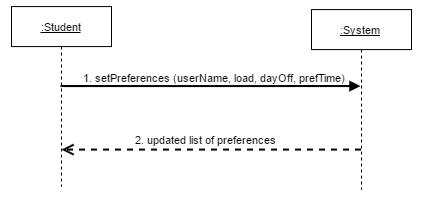
5.3 Set Preferences

*Set preferences* consists of saving the students preferences into the database, so they can be used later while generating the schedules.

**5.3.1 Full Use Case**

|  |  |  |  |
| --- | --- | --- | --- |
| Name: | Set Preferences | Author | Ying-Chen Chu |
| Identifier: | UC12 | Version: | 3.0 |
| Date Created: | 2015-02-03 | Last Modified: | 2015-02-08 |
| Importance: | 5/5 | | |
| Actor(s): | Student, Administrator | | |
| Goal: | Input the schedule preferences to the system | | |
| Summary: | Set the schedule preferences and save them to the system. | | |
| Related use-cases: | - | | |
| Preconditions | User has been authenticated | | |
| Trigger: | Previous state of the system remains unchanged. | | |
| Basic Flow: | 1. User indicates the wish to set schedule preferences.by selecting the desired “Day Off” and “Preferred Time” drop down menus.  2. System registers selections & displays schedule preferences. | | |
| Post-Conditions: | **Success:** The user’s preferences are saved to the system. | | |
| Minimum Guarantee: | Previous state of the system remains unchanged.  **Failure:** System fails to process the task, and an error messages is displayed. | | |
| Risk Assessment: | Low | | |

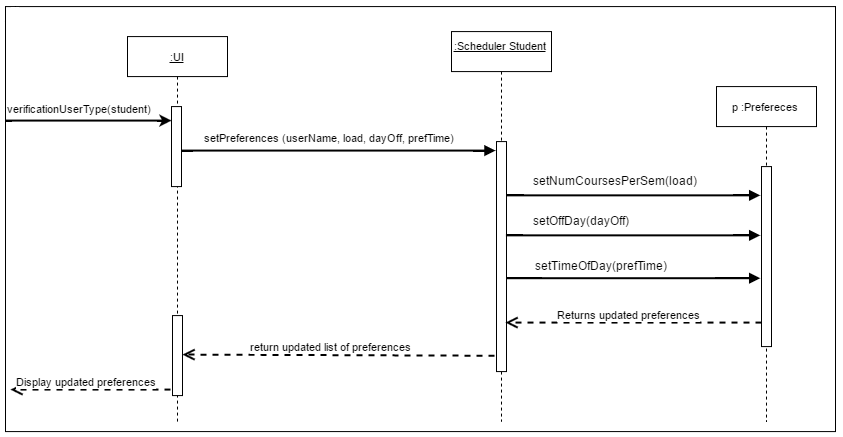
**5.3.2 System Sequence Diagram**



5.3.2.1 *setPreferences (userName, load, dayOff, periodDay)* Operating Contracts

|  |  |
| --- | --- |
| **Name** | Contract 3.1 Set Prefrences |
| **Operation** | setPreferences(String, int, String, String) |
| **Cross Reference** | UC12 Set Preferences |
| **Pre conditions** | -Student logged into the System |
| **Post Conditions** | -Attributes of preferences are modified and saved .  - Preferences are associated with the user. |

Sequence Diagram for Contract 3.1



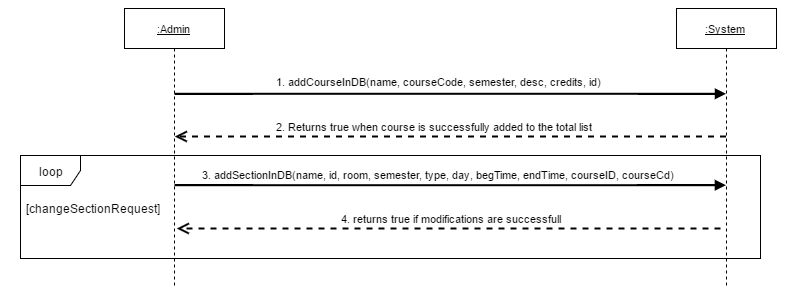
5.4 Add Course to Program

*Add course to program* consists of adding a course to the entire database of courses by the admin. By adding a course, the admin has to add the corresponding sections for lab, tutorial and lecture to this course.

**5.4.1 Full Use Case**

|  |  |  |  |
| --- | --- | --- | --- |
| Name: | Add Course to Program | Author | Adil Hssaini |
| Identifier: | UC23 | Version: | 2.0 |
| Date Created: | Feb 2, 2016 | Last Modified: | April 12, 2016 |
| Importance: | 5/5 | | |
| Actor(s): | Administrator | | |
| Goal: | To add a new course to a specific program. | | |
| Summary: | The Administrator updates the list of required courses for a specific program by adding a new course. | | |
| Related use-cases: |  | | |
| Preconditions | 1. Actor is logged on as administrator.  2. System has accessed the program menu.  3. Admin has accessed to the add course main menu | | |
| Trigger: | Administrator activates the “Add Course to Program” process. | | |
| Basic Flow: | 1. Administrator initiates the add a course process by entering and submitting the course Name, ID, Code, Semester, Description, & Number of Credits.  2. System verifies non-duplicity of information and adds supplied course information to the list of program courses.  3. Administrator enters and submits Section name, Id, Room, Semester, Type, day, Beginning Time, End Time, Course ID, Course Cd.  4. System updates course information with additional information provided. | | |
| Post-Conditions: | **Success:** Course is added successfully to the program listing. | | |
| Minimum Guarantee: | List of courses in the program stored by the system will not be affected.  **Failure:** The system fails to process the task and displays an error message. | | |
| Risk Assessment: | Low | | |

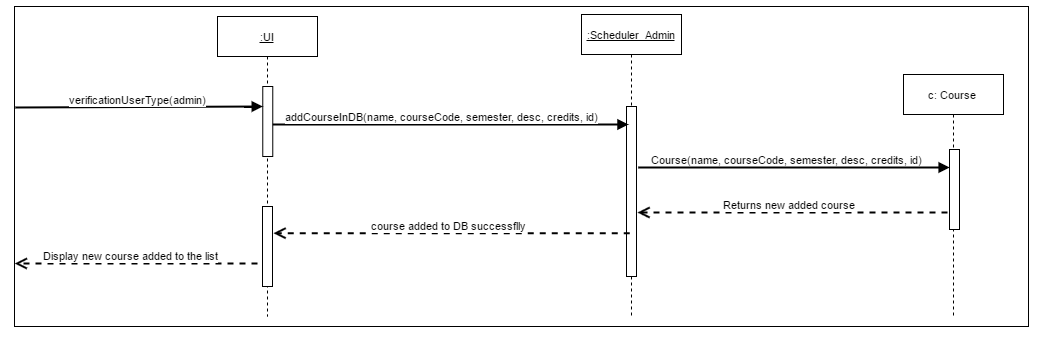
**5.4.2 System Sequence Diagram**



5.4.2.1 *addCourseInDB(name, courseCode, semester, desc, credits, id)* Operation Contracts

|  |  |
| --- | --- |
| **Name** | Contract 4.1 Add course to database |
| **Operation** | addCourseInDB(JSON) |
| **Cross Reference** | UC23 Add Course to Program |
| **Preconditions** | -Admin is logged in.  -Admin has prompted the *Add Course* section of the page. |
| **Postconditions** | -Instance of a course is created in the course database. |

Sequence Diagram for Contract 4.1



5.4.2.2 *addSectionInDB(name, id, room, semester, type, day, begTime, endTime, courseID, courseCd)* Operation Contracts

|  |  |
| --- | --- |
| **Name** | Contract 4.2 Add section in database |
| **Operation** | addSectionInDB(JSON) |
| **Cross Reference** | UC23 Add Course to Program |
| **PreConditions** | -User Logged in as administrator  -Course already exists |
| **Post Conditions** | -New instance of section created into the section database. |

Sequence Diagram for Contract 4.2

