|  |
| --- |
| UC1: Launching the application |
| Description: |
| Starting the application |
|  |
| Preconditions: |
| There must be a program executable file for the program on the computer |
|  |
| Postconditions: |
| 1. The application is running 2. Login screen appears. |
|  |
| Basic Course of Action: |
| 1. The user wants to open the application 2. Finds the executable file of the application 3. Clicks on the executable file. 4. The application opens with login screen. |

|  |
| --- |
| UC2: Logging in using student identity number |
| Description: |
| Logging in to the system. |
|  |
| Preconditions: |
| 1. The application is running 2. The user has not logged in yet |
| Postconditions: |
| 1. The user is logged in 2. The current view changes to the view of list of courses |
|  |
| Basic Course of Action: |
| 1. The user enters their student ID. 2. The user clicks the login button 3. The user info is checked using the database    1. If the user has already logged in,       1. Prompt an error message indicating that the user has already logged in.       2. Clear the textbox for entering another student identity number    2. If the has not logged in before,       1. Get the respective data from the database using the provided student identity number       2. Show the fetched information on the top right-hand side of the screen, indicating the student name, surname, department and id.       3. Changing the view to show all offered courses |

|  |
| --- |
| UC3: Listing all offered courses |
| Description: |
| Listing all courses available for the current semester |
|  |
| Preconditions: |
| 1. The user has already logged in to the system. |
|  |
| Postconditions: |
| 1. All the available courses offered at the current semester are listed. |
|  |
| Basic Course of Action: |
| 1. Application attempts to establish a connection to the database    1. If connection cannot be established,       1. Prompt an error message indicating that connection cannot be established       2. Trying to re-connect every 10 seconds.    2. If successfully connected to the DB,       1. Application sends a query to database requesting the list of offered courses for the current semester.       2. Application receives the list of offered courses       3. Application reveals the received list of offered courses in such a way that every course has its own checkbox and is selectable by the user.       4. Course selection view is opened.   Exceptions:  \* . If connection is lost,   1. Prompt an error message indicating that connection is lost 2. Trying to re-connect every 10 seconds. |

|  |
| --- |
| UC4: Selecting the desired courses |
| Description: |
| Deciding on the desired courses and selecting them |
|  |
| Preconditions: |
| 1. The list of offered courses has already been listed. |
|  |
| Postconditions: |
| 1. The desired courses are selected. 2. Priority selection view is opened. |
|  |
| Basic Course of Action: |
| 1. The user decides on what courses to take for the current semester. 2. The user finds the desired courses from the available courses list. 3. The user checks the checkboxes of the found desired courses 4. The user clicks the “Done” button at the end of the page 5. The user’s selected courses are saved locally. 6. Priority selection view is opened. |

|  |
| --- |
| UC5: Designating priority for each selected course |
| Description: |
| Deciding on the priority for the selected courses |
|  |
| Preconditions: |
| 1. The desired courses have been selected. |
|  |
| Postconditions: |
| 1. The priorities for each course are selected. 2. The scheduling algorithm starts running. |
|  |
| Basic Course of Action: |
| 1. Application defines a priority range from 1 to the number of selected courses. (i.e. If the user has previously selected 5 courses, then it is only allowed to choose a priority number between 1 and 5, where 1 indicates the highest priority) 2. For each course, the user must choose a unique priority number from the nearby drop-down list. (i.e. Two or more courses cannot have the same priority number.) 3. The user clicks the “Done” button at the end of the page 4. The user’s selected course priorities are saved locally. 5. The scheduling algorithm starts running. |

|  |
| --- |
| UC6: Creating a non-conflicting schedule(s) |
| Description: |
| The scheduling algorithm creates a non-conflicting course schedule(s) |
|  |
| Preconditions: |
| 1. Priorities for each course have been selected. |
|  |
| Postconditions: |
| 1. A non-conflicting schedule has been generated. 2. The generated schedules view is opened. |
|  |
| Basic Course of Action: |
| 1. The application generates a non-conflicting schedule(s) using the locally saved selected courses and their priorities. 2. The generated schedules are saved locally. 3. The generated schedules are opened in list view. |

|  |
| --- |
| UC7: Displaying the generated schedules in list view |
| Description: |
| The application displays the generated schedules in a list view. |
|  |
| Preconditions: |
| 1. The non-conflicting schedules have been generated. |
|  |
| Postconditions: |
| 1. The application displays the generated schedules in list view. 2. The application offers to display a schedule in weekly view. |
|  |
| Basic Course of Action: |
| 1. The received outputs of the scheduling algorithm are converted into list view for each non-conflicting schedule. 2. The lists are spread around the screen as separate tiles. |

|  |
| --- |
| UC8: Choosing a schedule to be displayed in weekly view |
| Description: |
| User chooses a schedule to be displayed in weekly view with a pop-up. |
|  |
| Preconditions: |
| 1. The list view of the schedule has been generated. |
|  |
| Postconditions: |
| 1. A pop-up window shows the chosen schedule in a weekly view. |
|  |
| Basic Course of Action: |
| 1. The user picks a schedule from the tiles representing each non-conflicting schedule 2. The user clicks on the desired schedule. 3. A pop-up window appears. 4. Application displays the picked schedule in a weekly view within the pop-up window. |

|  |
| --- |
| UC9: Printing the selected schedule |
| Description: |
|  |
|  |
| Preconditions: |
|  |
|  |
| Postconditions: |
|  |
|  |
| Basic Course of Action: |
|  |

|  |
| --- |
| UC10: Sharing the selected schedule |
| Description: |
|  |
|  |
| Preconditions: |
|  |
|  |
| Postconditions: |
|  |
|  |
| Basic Course of Action: |
|  |

|  |
| --- |
| UC11: Closing the view of the selected schedule |
| Description: |
|  |
|  |
| Preconditions: |
|  |
|  |
| Postconditions: |
|  |
|  |
| Basic Course of Action: |
|  |

|  |
| --- |
| UC12: Modifying the priorities of courses |
| Description: |
|  |
|  |
| Preconditions: |
|  |
|  |
| Postconditions: |
|  |
|  |
| Basic Course of Action: |
|  |

|  |
| --- |
| UC13: Dropping the selected courses |
| Description: |
|  |
|  |
| Preconditions: |
|  |
|  |
| Postconditions: |
|  |
|  |
| Basic Course of Action: |
|  |

|  |
| --- |
| UC14: Clearing all selected courses |
| Description: |
|  |
|  |
| Preconditions: |
|  |
|  |
| Postconditions: |
|  |
|  |
| Basic Course of Action: |
|  |

|  |
| --- |
| UC15: Closing the program |
| Description: |
|  |
|  |
| Preconditions: |
|  |
|  |
| Postconditions: |
|  |
|  |
| Basic Course of Action: |
|  |

|  |
| --- |
| UC16: Terminating the algorithm before completion ??????? |
| Description: |
|  |
|  |
| Preconditions: |
|  |
|  |
| Postconditions: |
|  |
|  |
| Basic Course of Action: |
|  |