|  |  |  |
| --- | --- | --- |
| **Use Case** | Add Classes | |
| **Description** | Add classes to a student’s schedule | |
| **Precondition** | The user is on the “Student View” screen | |
| **Success End Condition** | The classes are added to the student’s schedule. | |
| **Failed End Condition** | The classes are not added to the student’s schedule. | |
| **Actors** | Student, Advisor (Process is identical) | |
| **Used by** |  | |
| **Trigger** |  | |
| **Basic Flow** | **Step** | **Action** |
| 1 | Student clicks checkbox next to each class he/she wants to take |
| 2 | Student clicks the “Submit” button |
| 3 | App checks that the student has completed all necessary prerequisites for a class (see other info) |
|  | 4 | Repeat step 3 for each class selected |
|  | 5 | App transitions to “Schedule View” screen |
|  | 6 | App displays a table containing the added courses, a status message saying “Added,” and checks the boxes in the “Student Wants” column |
|  |  |  |
| **Alternate** | 1a | Advisor clicks checkbox next to each class he/she wants the student to take |
| 2a | Advisor clicks the “Submit” button |
| 3a | Continue at Step 3 |
|  |  |
| 6b | App displays a table containing the selected courses and a status message indicating why a class could not be added |
| **Exception** | 1a |  |
| **Other Information** | * Note step 3: The exact mechanism for doing this will change depending on whether we have a local copy of the classes or a database containing the classes * Steps 3 and 4 may be coded as a single step * Duplicate courses will be prevented by the structure of the ui; thus, it would be redundant to include here | |
| **Open Issues** | Step 3 lacks detail as we don’t know what access we will have to the course list | |
| **Version** | Version #1; DATE – 8/25/20 | |