**Software Requirements Specification**  
**CPE 656  Software Engineering Studio I**  
**Team B**  
**December 3, 2012**  
  
**Nick Lathem**

**Chris Smeal**

**Adrian Flannery**

**Natalie Kelly**

**1.  System Overview**  
  
**Brief Description: Graduation Planning System is a tool that will assist in the guidance of a successful plan for students in their perspective Program of Study. GPS, will help the student in choosing the “best” option for their busy schedules. This tool in its tenure will alleviate not only the student’s stress but the Advisors responsibilities; ultimately creating a more successful graduation rate and many satisfied customers.**

**-Initial Set Up (pictured above)**

**2.  Referenced Documents**  
  
**STD**  
  
**3.  Definitions and Acronyms**  
  
**GPS-Graduation Planning System**

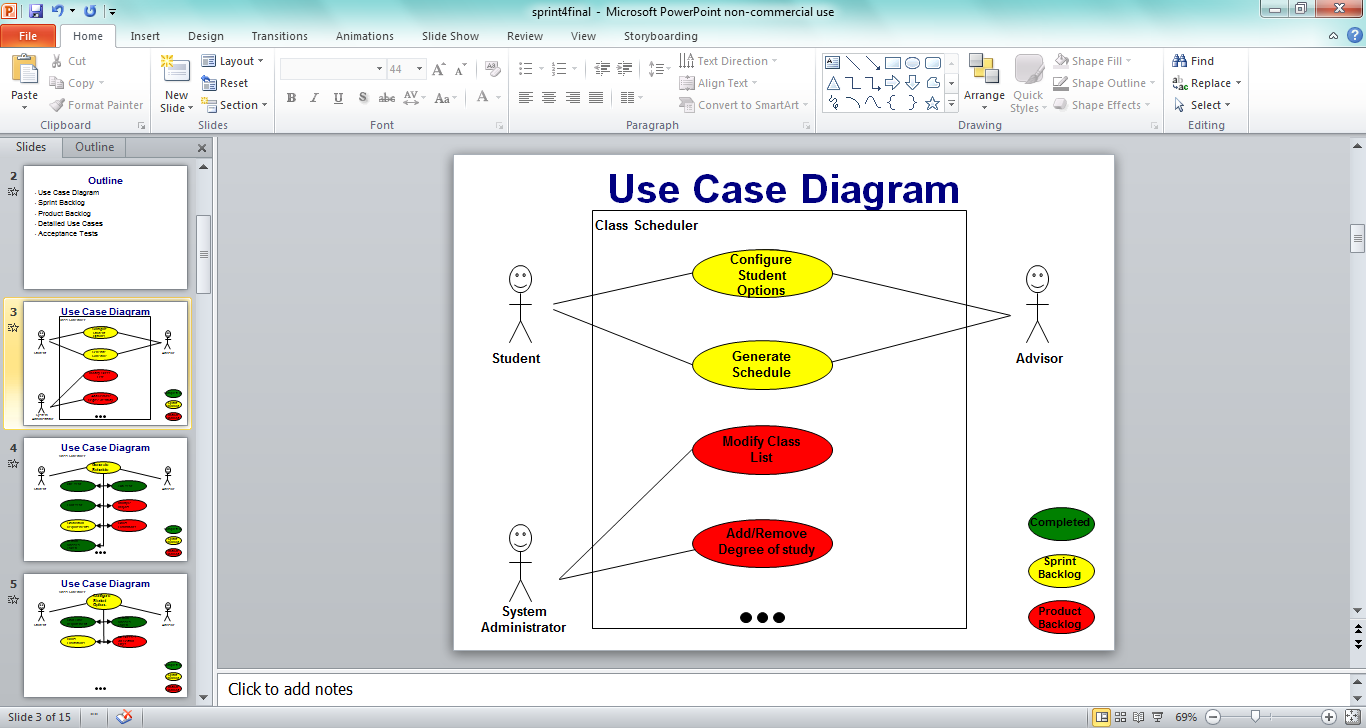
**POS-Plan of Study**  
  
**4.  Functional Requirements**  
  
**4.1  Required Roles**

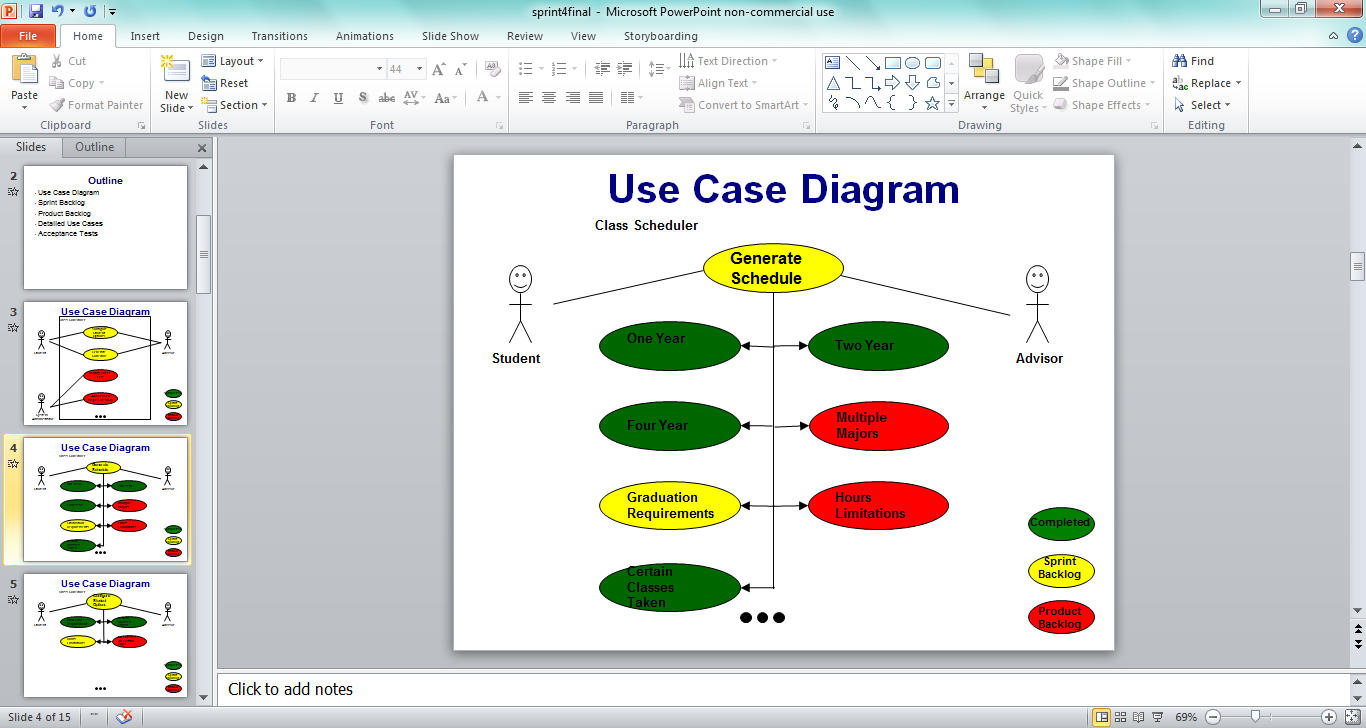
**Students**

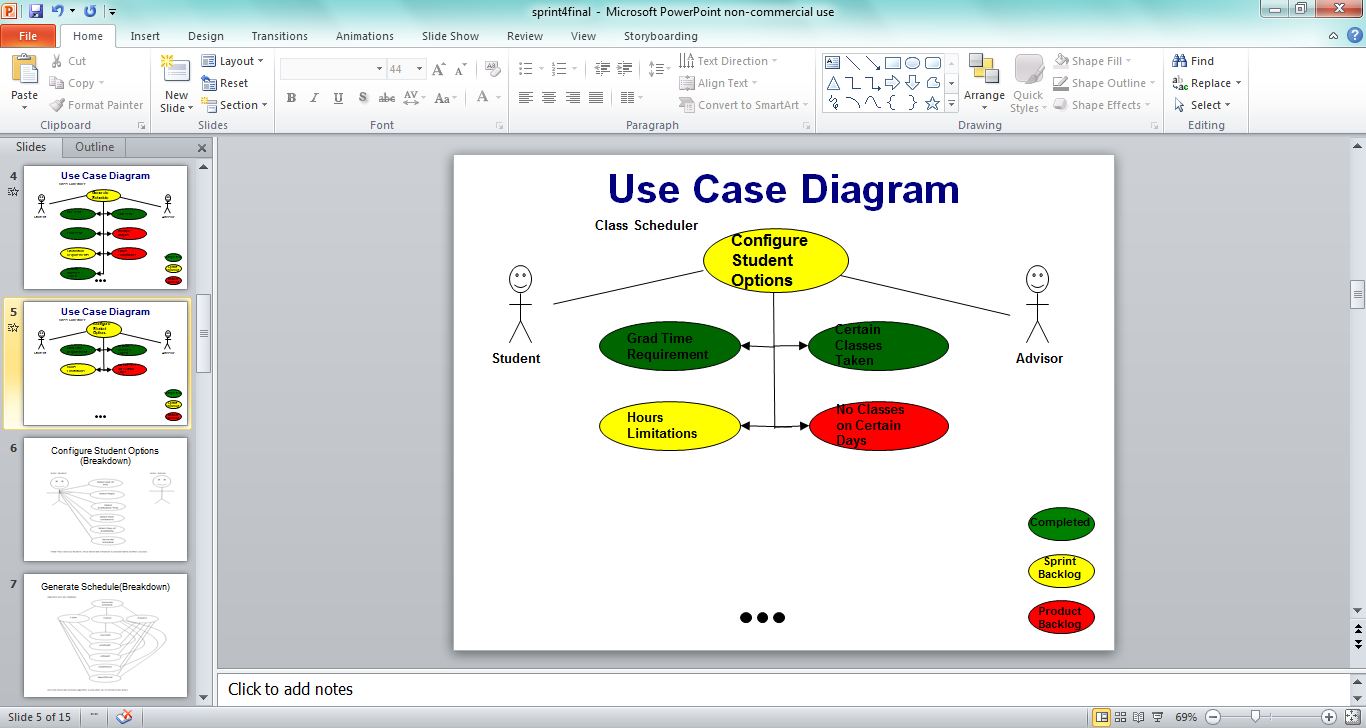
**Advisors**

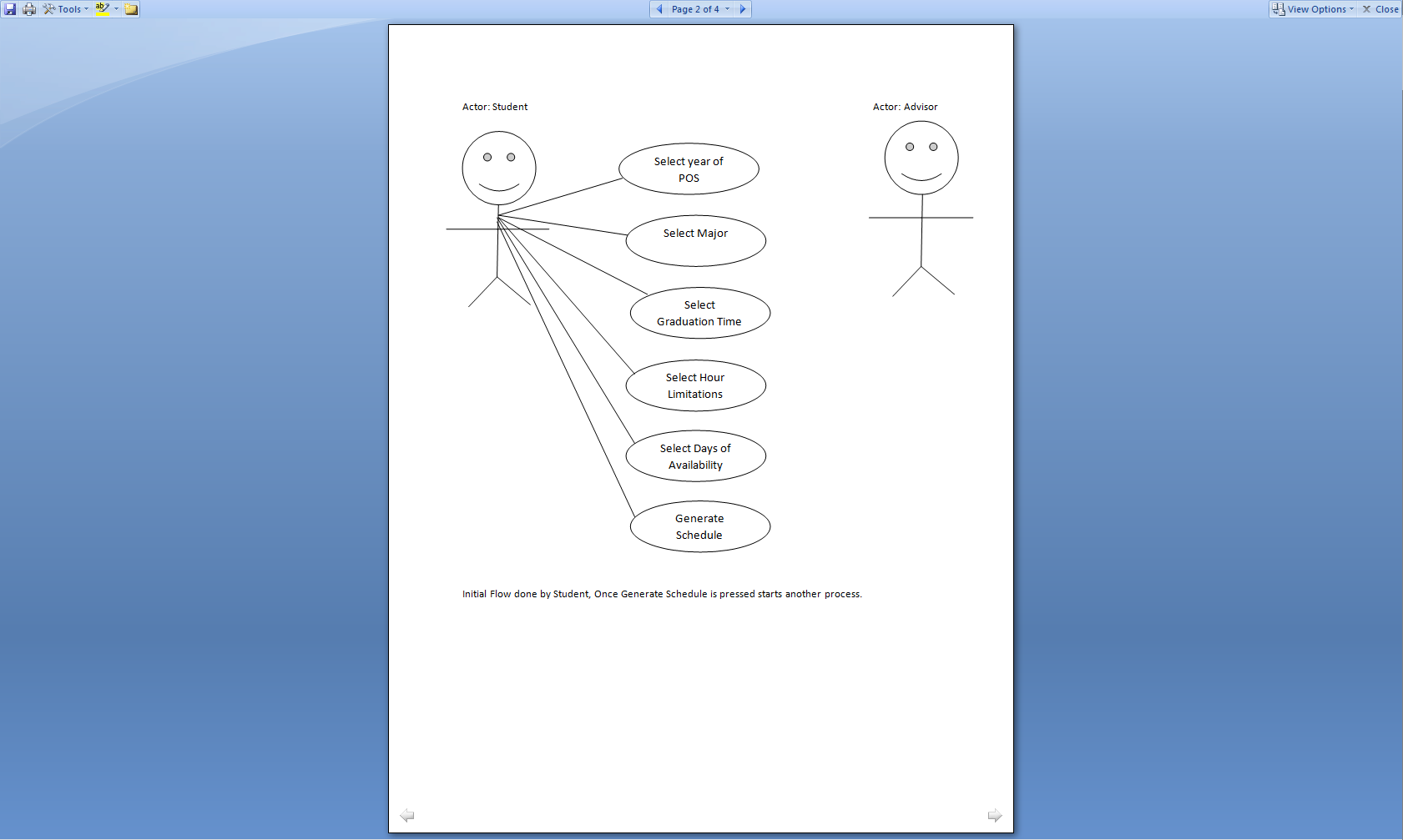
**System Admin**

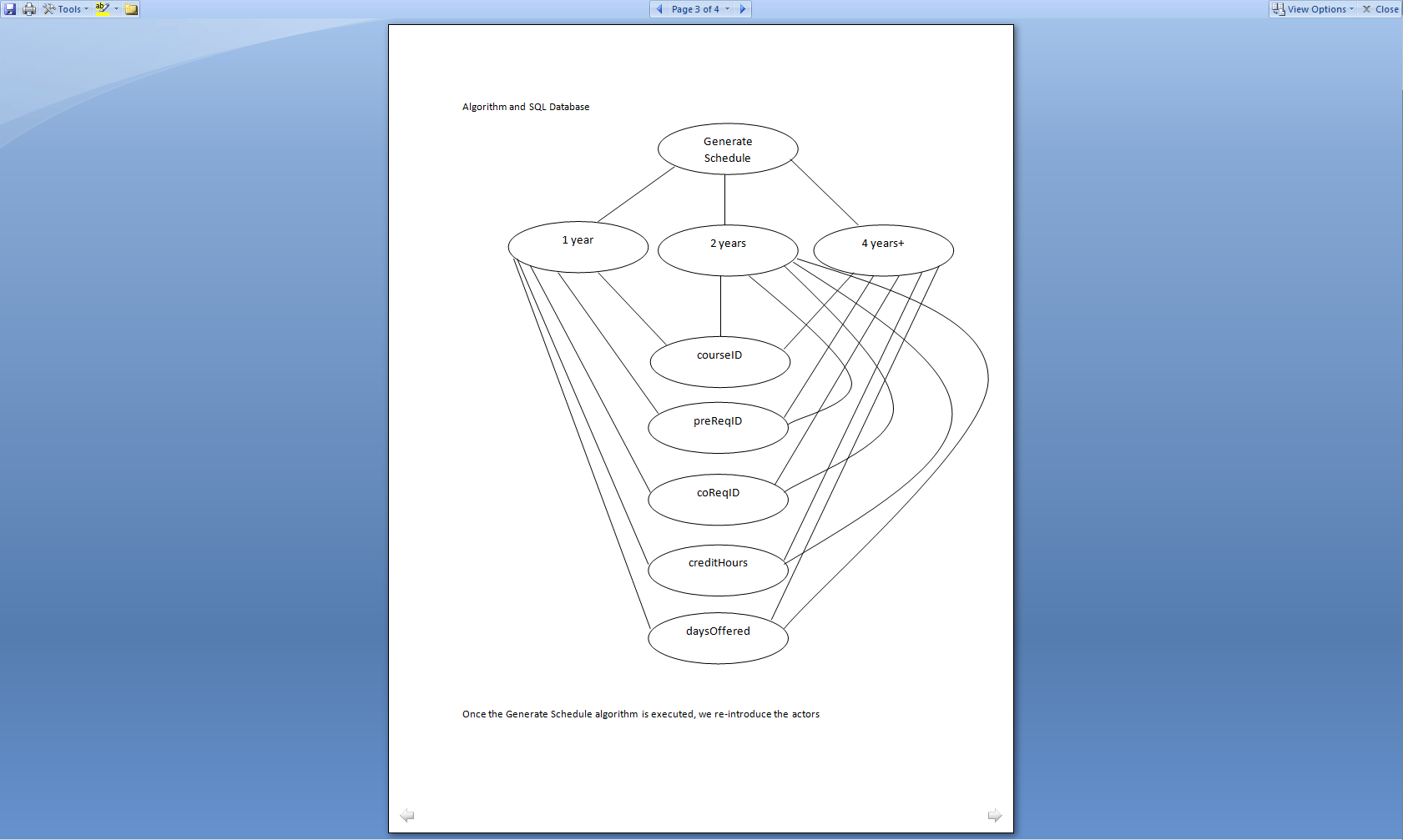
**4.2  Use Case Diagram/Breakdown of Use Cases**

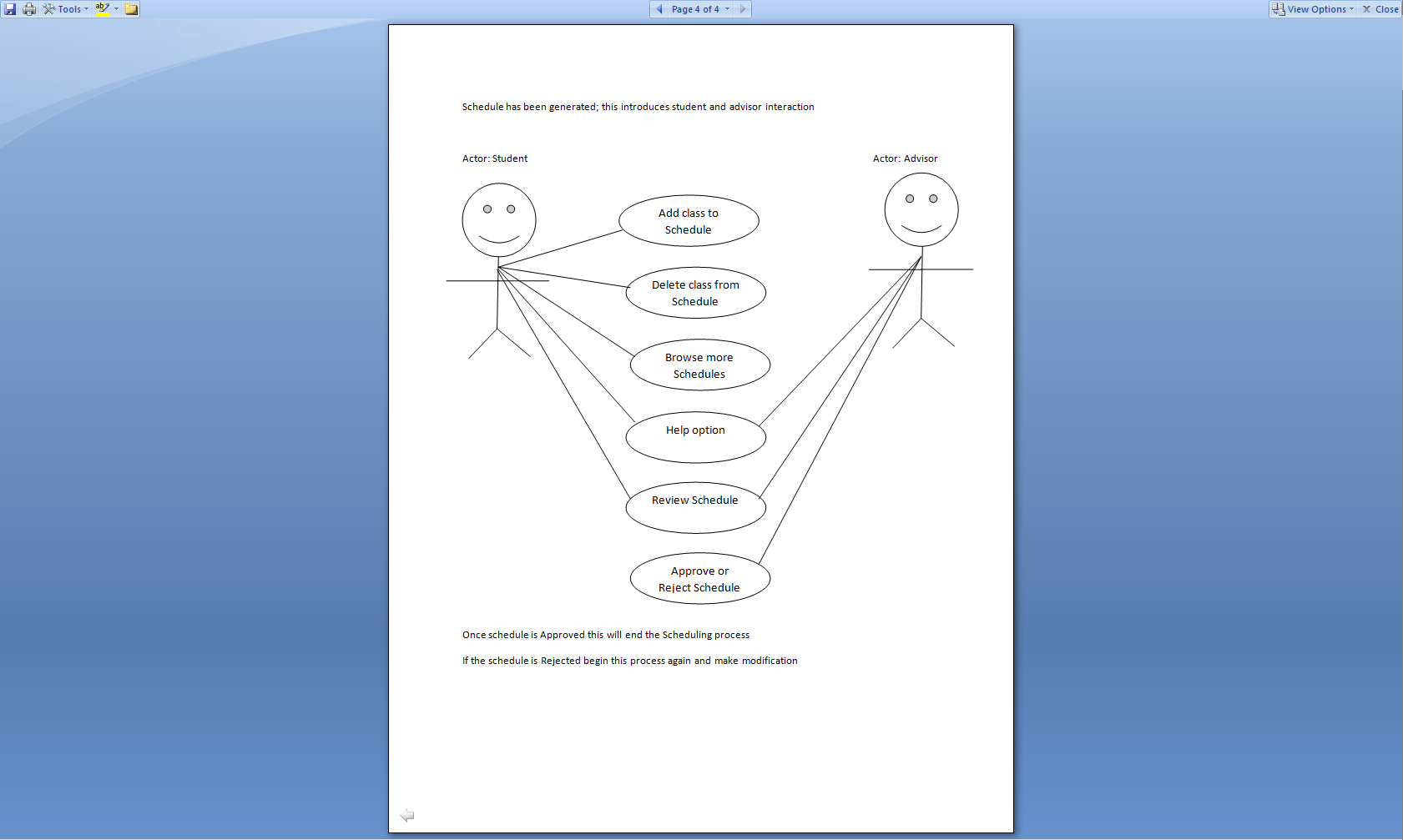


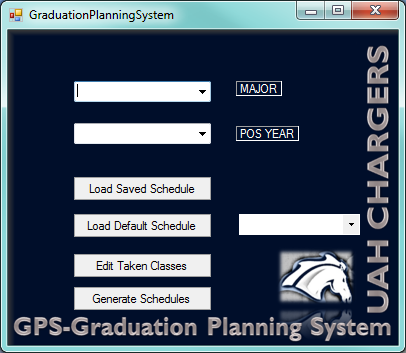




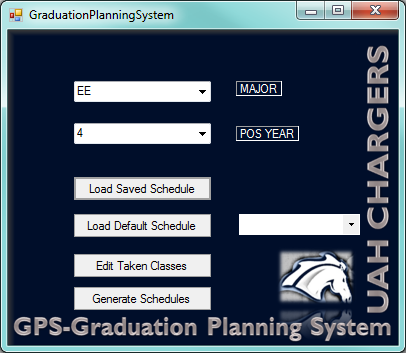




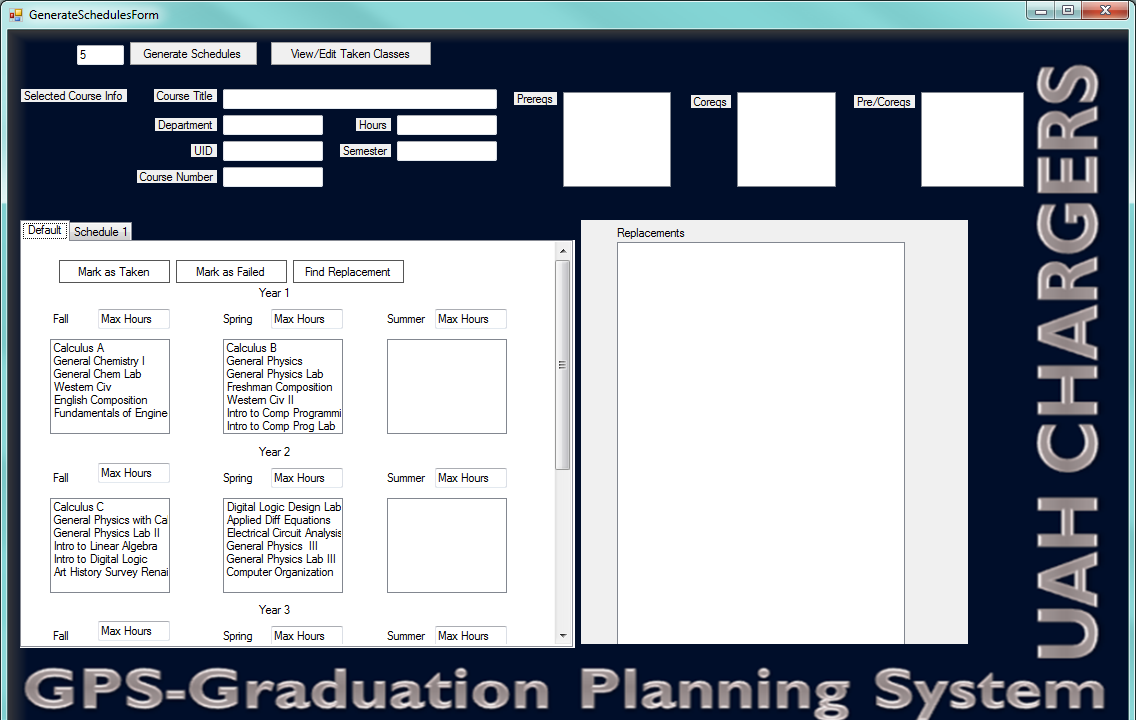
  
**4.3 Story Board**



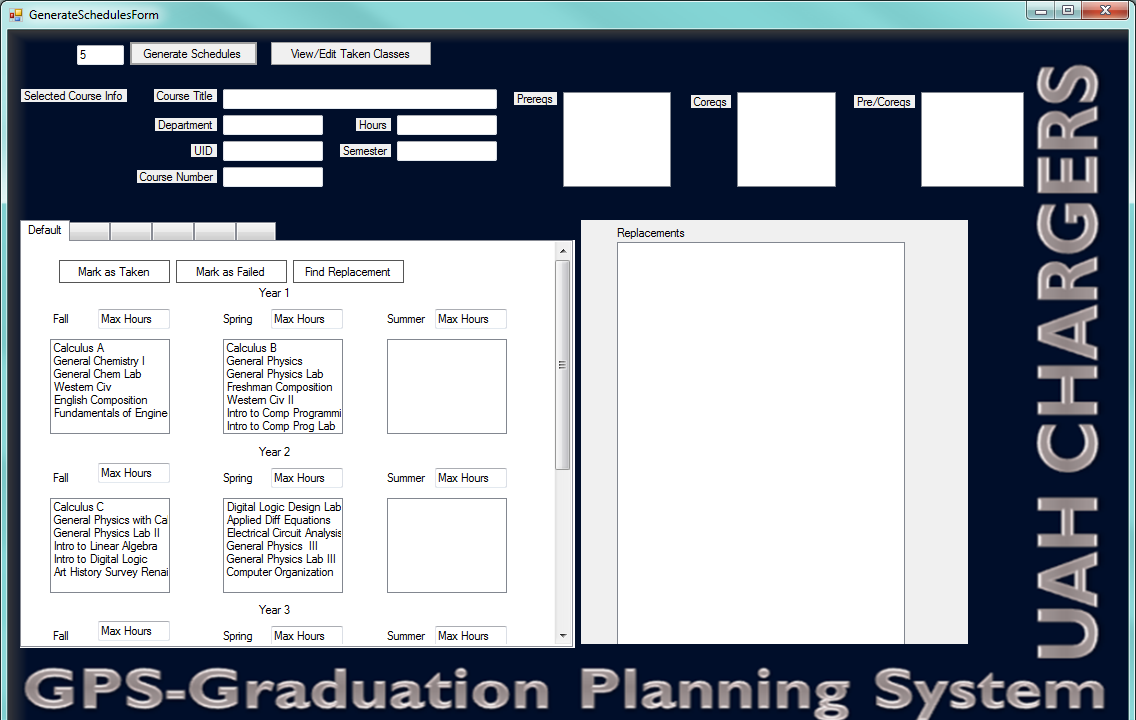
**Pictured above is the initial screen. Here the user has the ability to “Select” their major, “Select” their Plan of Study Year, Load a Saved Schedule previously saved by the user, Load the Default Schedule which has been created as a template for “new” users, Edit Taken Classes and Generate Schedules based on your set preference. Pictured below the user has selected the “EE” major which is Electrical Engineering and has selected the POS Year as a value of “4”.**



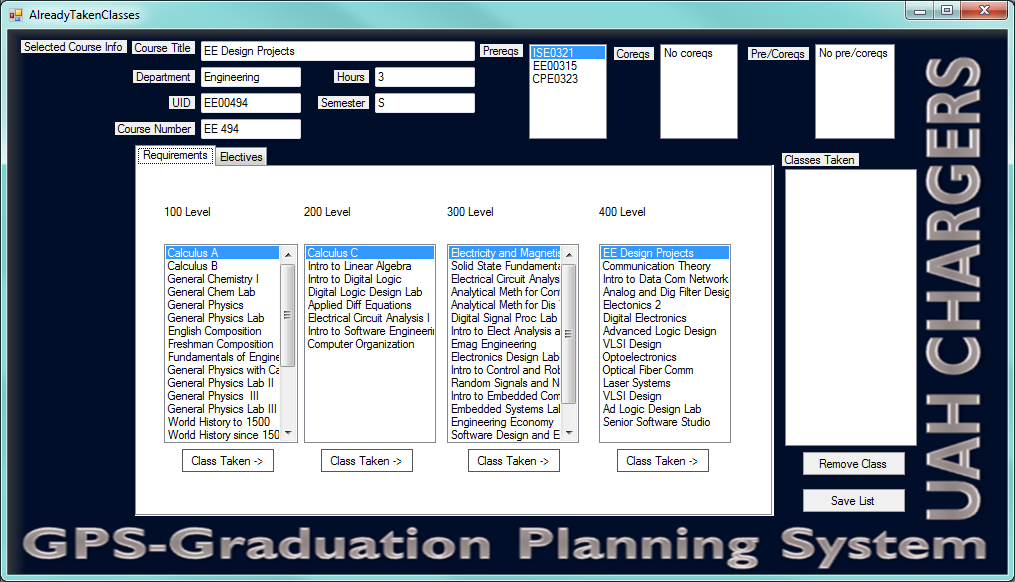
**After “Clicking” the “Generate Schedules” button; the screen below will be displayed. Here the user can generate a variable amount of schedules based on the parameters selected from the initial screen. Here all the information from the DB has been loaded in this form. The user can mark the classes as taken, failed or find a replacement for a particular class if available.**

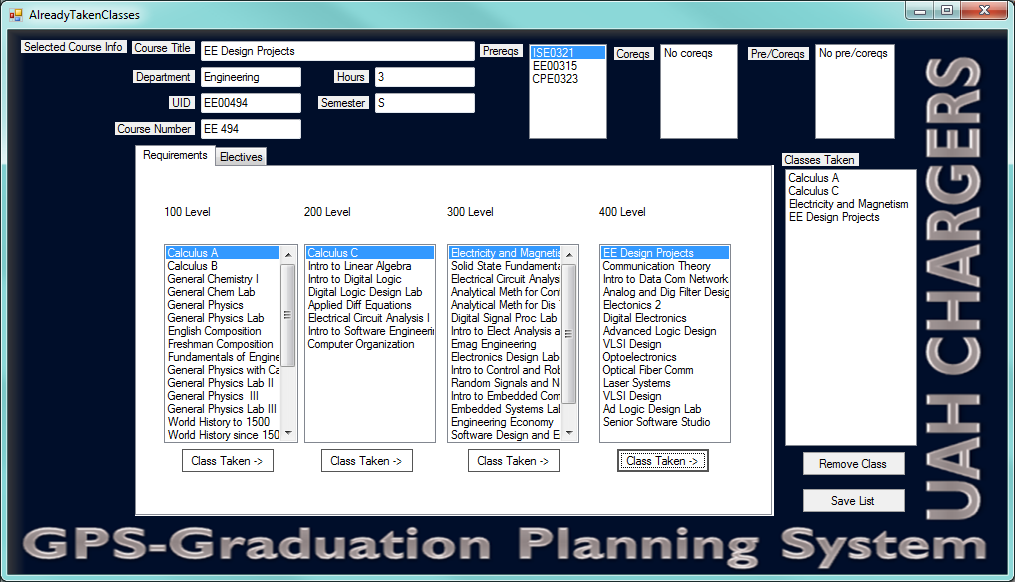


**If the user wants to generate multiple schedules then this will produce multiple tabs with different schedule combinations, the user may choose the best option. Also the user can “Mark” a class as Taken, Failed, or Find a Replacement**

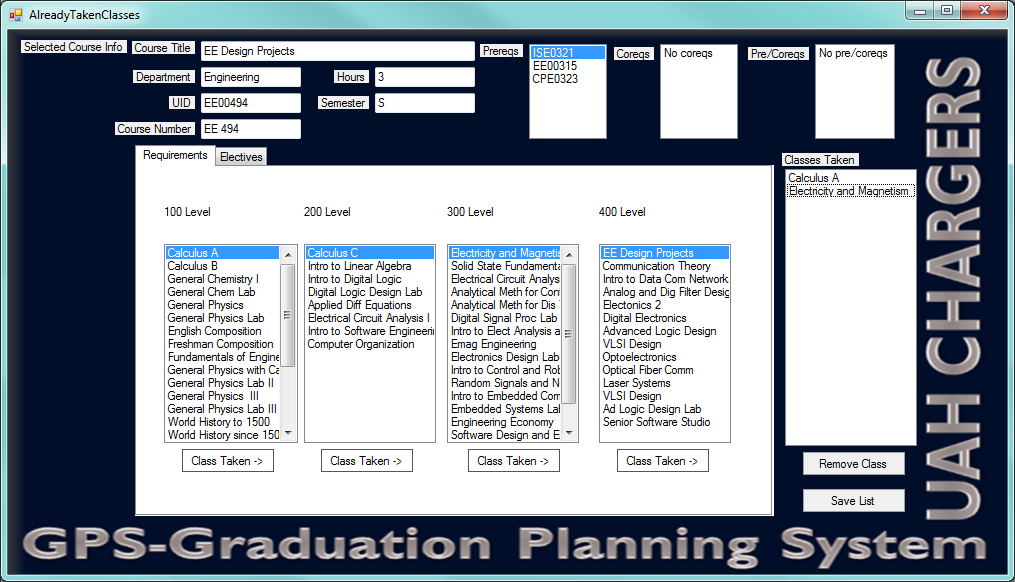


**If the user “selects” the Edit Taken Classes option the screen below will be displayed. Here the user will pull the information from the DB and select which classes they have already taken. If a class was mistakenly added to the list the user has the ability to “Remove” the class from the list or “Save” the list.**





**Above user has added some classes to the “Classes Taken” list box.**



**Above the user has clicked the “Remove Class” button to delete two of the four courses added by the previous action**

**🡪 Add E-mail communication or some kind of way to communicate with the student and advisor/data controller (Banner administrator) (planned)**

**4.4 Use Case Descriptions**

**Use Case Number:** UC001

**Use Case Name:** Configure Student Options

**Actors:** Student

**Preconditions:** At least one property has been listed.

**Postconditions:** None

**Brief Description:** Buyer browses through property listings categorized by price.

**Basic Flow of Events:**

1. Buyer requests to review property listings.
2. System responds with a selection of price search criteria.
3. Buyer inputs the following search parameters and initiates the search: **lower price limit, upper price limit**

4. System responds with list of properties whose prices are greater than or equal to the lower price limit but less

than or equal to the upper price limit.

**Alternative Flow #1:** Omitted **lower price limit** only

3. Buyer inputs **upper price limit** and initiates the search.

4. System assumes **lower price limit** = $0.00

5. System responds with list of properties whose prices are greater than or equal to $0.00 and less than or equal

to the **upper price limit**.

**Alternative Flow #2:** Omitted **upper price limit** only

3. Buyer inputs **lower price limit** and initiates the search.

4. System responds with list of properties whose prices are greater than or equal to **lower price limit**.

**Alternative Flow #3:** Omitted both **lower price limit** and **upper price limit**

3. Buyer initiates the search.

4. System responds with list of all properties.

**Use Case Number:** UC002

**Use Case Name:** Generate Schedule

**Actors:** Student and Advisor

**Preconditions:** At least one property has been listed.

**Postconditions:** None

**Brief Description:** Buyer browses through property listings categorized by price.

**Basic Flow of Events:**

1. Buyer requests to review property listings.
2. System responds with a selection of price search criteria.
3. Buyer inputs the following search parameters and initiates the search: **lower price limit, upper price limit**

4. System responds with list of properties whose prices are greater than or equal to the lower price limit but less

than or equal to the upper price limit.

**Alternative Flow #1:** Omitted **lower price limit** only

3. Buyer inputs **upper price limit** and initiates the search.

4. System assumes **lower price limit** = $0.00

5. System responds with list of properties whose prices are greater than or equal to $0.00 and less than or equal

to the **upper price limit**.

**Alternative Flow #2:** Omitted **upper price limit** only

3. Buyer inputs **lower price limit** and initiates the search.

4. System responds with list of properties whose prices are greater than or equal to **lower price limit**.

**Alternative Flow #3:** Omitted both **lower price limit** and **upper price limit**

3. Buyer initiates the search.

4. System responds with list of all properties.

**Use Case Number:** UC003

**Use Case Name:** Modify Class List

**Actors:** Student and Advisor

**Preconditions:** At least one property has been listed.

**Postconditions:** None

**Brief Description:** Buyer browses through property listings categorized by price.

**Basic Flow of Events:**

1. Buyer requests to review property listings.
2. System responds with a selection of price search criteria.
3. Buyer inputs the following search parameters and initiates the search: **lower price limit, upper price limit**

4. System responds with list of properties whose prices are greater than or equal to the lower price limit but less

than or equal to the upper price limit.

**Alternative Flow #1:** Omitted **lower price limit** only

3. Buyer inputs **upper price limit** and initiates the search.

4. System assumes **lower price limit** = $0.00

5. System responds with list of properties whose prices are greater than or equal to $0.00 and less than or equal

to the **upper price limit**.

**Alternative Flow #2:** Omitted **upper price limit** only

3. Buyer inputs **lower price limit** and initiates the search.

4. System responds with list of properties whose prices are greater than or equal to **lower price limit**.

**Alternative Flow #3:** Omitted both **lower price limit** and **upper price limit**

3. Buyer initiates the search.

4. System responds with list of all properties.

**Use Case Number:** UC004

**Use Case Name:** Add/Remove Degree of Study

**Actors:** Student and Advisor

**Preconditions:** At least one property has been listed.

**Postconditions:** None

**Brief Description:** Buyer browses through property listings categorized by price.

**Basic Flow of Events:**

1. Buyer requests to review property listings.
2. System responds with a selection of price search criteria.
3. Buyer inputs the following search parameters and initiates the search: **lower price limit, upper price limit**

4. System responds with list of properties whose prices are greater than or equal to the lower price limit but less

than or equal to the upper price limit.

**Alternative Flow #1:** Omitted **lower price limit** only

3. Buyer inputs **upper price limit** and initiates the search.

4. System assumes **lower price limit** = $0.00

5. System responds with list of properties whose prices are greater than or equal to $0.00 and less than or equal

to the **upper price limit**.

**Alternative Flow #2:** Omitted **upper price limit** only

3. Buyer inputs **lower price limit** and initiates the search.

4. System responds with list of properties whose prices are greater than or equal to **lower price limit**.

**Alternative Flow #3:** Omitted both **lower price limit** and **upper price limit**

3. Buyer initiates the search.

4. System responds with list of all properties.

**For Use Cases still on the product backlog as of the end of Fall 2012, be sure to provide, at minimum, the Sample Screenshot, Use Case Number, Use Case Name, and a Brief Description**.  
  
**4.5.  Supplemental Requirements**  
**N/A**

**7.  Computer Resource Requirements**

CRR0001-Windows Operating System only\*

\*Further research on multiplatform distribution  
  
**8.  Qualification Provisions**

|  | ID | Description |
| --- | --- | --- |
| Requirement | UC 001 | Configure Student Options |
| Initial conditions | Clean. | Start a new instance of the program. Then begin the test. |
| Additional Assumptions | None | NA |
| **Test Procedures** |  |  |
| **Test Step** | **Operator actions** | **Result** |
| 1 | Start GPS | GPS starts. |
| 2 | Change Major to EE | Major changes successfully |
| 3 | Change Year to 2012 | Year changed to 2012 |
| 4 | Select the “Edit Classes Taken” button | A pop up with a list of classes to select as taken is brought up. |
| 5 | Mark several classes as taken. | The marked classes are visually displayed as taken. |
| 6 | Click the “Save” button. | The pop up disappears. |
| 7 | Click the “Generate Schedules” button. | A schedule meeting all the criteria above is generated. |

|  | ID | Description |
| --- | --- | --- |
| Requirement | UC 002 | Generate Schedules |
| Initial conditions | Clean. | Start a new instance of the program. Then begin the test. |
| Additional Assumptions | None | NA |
| **Test Procedures** |  |  |
| **Test Step** | **Operator actions** | **Result** |
| 1 | Start GPS | GPS starts. |
| 2 | Change Major to EE | Major changes successfully |
| 3 | Change Year to 2012 | Year changed to 2012 |
| 4 | Change the “Load Default Schedule” drop down to “Major Only” | The “Load Default Schedule” option is changed. |
| 5 | Click the “Load Default Schedules” button. | A new window is brought up with a schedule meeting the above criteria. |
| 6 | Change the “Number of schedules generated” to 5 |  |
| 7 | Click the “Generate Schedules” button. | 5 schedules are generate meeting the above criteria. |
| 8 | Change several of the Semesters “Max hours” to zero and other assorted values. |  |
| 9 | Click the “Generate Schedules button. | 5 new schedules are generated meeting the above criteria. |
| 10 | Change the “Number of schedules generated” to 20 |  |
| 11 | Click the “Generate Schedules” button. | 20 new schedules are generated meeting the above criteria. |

|  | ID | Description |
| --- | --- | --- |
| Requirement | UC 003 | Generate Schedules |
| Initial conditions | Clean. | Start a new instance of the program. Then begin the test. |
| Additional Assumptions | None | NA |
| **Test Procedures** |  |  |
| **Test Step** | **Operator actions** | **Result** |
| 1 | Start GPS | GPS starts. |
| 2 | Change Major to EE | Major changes successfully |
| 3 | Change Year to 2012 | Year changed to 2012 |
| 4 | Change the “Load Default Schedule” drop down to “Default Schedule” | The “Load Default Schedule” option is changed. |
| 5 | Click the “Load Default Schedules” button. | A new window is brought up with a schedule meeting the above criteria. |
| 6 | Drag and drop a number of classes to different semesters. | Classes are allowed to move to viable spots, but not where they lack needed pre-reqs. |
| 7 | Select a class and click the “Find Replacement” button. Repeat several times. | Classes with viable replacements display such replacements. |
| 8 | Select a class and press the “Mark as failed” button. Repeat several times, then click the “Refresh” button. | The classes are marked as failed. After the “Refresh” button is pressed, a new schedule is generated bearing these changes. |
| 9 | Click the “Generate Schedules” button. | A number of new schedules appear in the window. |

|  | ID | Description |
| --- | --- | --- |
| Requirement | UC 004 | Add/Remove Program of Study |
| Initial conditions | Clean. | Start a new instance of the program. Then begin the test. |
| Additional Assumptions | None | NA |
| **Test Procedures** |  |  |
| **Test Step** | **Operator actions** | **Result** |
| 1 | Start GPS Admin Tool | GPS Admin Tool starts. |
| 2 | Click the “Add POS” button. | A wizard allowing for the creation of a new POS is created. |
| 3 | Make changes to this POS. |  |
| 4 | Click the save button. | The new POS is added. |
| 5 | Click the “Edit existing POS” button. | An existing POS is brought up for edit. |
| 6 | Click the “Save” button. | Changes are saved to the POS. |

**Appendix A: Use Case Status**

****