[1.Project Description 2](#_Toc500131300)

[2. Requirements 3](#_Toc500131301)

[2.1 Requirements for the student 3](#_Toc500131302)

[2.2 Requirements for the Faculty Advisor 4](#_Toc500131303)

[2.3 Requirements for the Academic Advisor 5](#_Toc500131304)

[2.4 Requirements for the Senior Secretary 7](#_Toc500131305)

[3. USECASES 8](#_Toc500131306)

[3.1 Use Case for Student 10](#_Toc500131307)

[3.2. Use Case for Faculty Advisor 12](#_Toc500131308)

[3.3 Use Case for Academic Advisor 15](#_Toc500131309)

[3.4 Use Case for Senior Secretary: 20](#_Toc500131310)

[4.Task Analysis 22](#_Toc500131311)

[4.1Task Analysis for Student 23](#_Toc500131312)

[4.2Task Analysis for Faculty Advisor 24](#_Toc500131313)

[4.3Task Analysis for Academic Advisor 25](#_Toc500131314)

[4.4Task Analysis for Senior Secretary 26](#_Toc500131315)

[Task Analysis for Generate Draft 55](#_Toc500131316)

[5.Architecture 86](#_Toc500131317)

[6. Activity Diagrams 88](#_Toc500131318)

[6.1 Activity diagram for the Student 88](#_Toc500131319)

[6.2 Activity diagram for Faculty Advisor 90](#_Toc500131320)

[6.3 Activity diagram for Academic Advisor 92](#_Toc500131321)

[6.4 Activity diagram for a Senior Secretary 95](#_Toc500131322)

[7.Class Diagrams 97](#_Toc500131323)

[7.1 Class Diagram for academicadmin Database 97](#_Toc500131324)

[7.2 Class Diagram for CandidatePlanOfStudy Database 98](#_Toc500131325)

[7.3 Class Diagram for CourseCatalog Database 99](#_Toc500131326)

[7.4 Class Diagram for eservices Database 100](#_Toc500131327)

[8.Project Guidelines 101](#_Toc500131328)

[9.Design Rationale 105](#_Toc500131329)

[11.QUIS Report 106](#_Toc500131330)

[11.1 QUIS for User Interface 108](#_Toc500131331)

[11.2 QUIS for System Functionality 110](#_Toc500131332)

[11.3 QUIS for System Performance 113](#_Toc500131333)

[11.4 QUIS for System Usability 115](#_Toc500131334)

[12. Feedback obtained from the users: 118](#_Toc500131335)

# 1.Project Description

Candidate Plan of Study (CPS) is a student degree plan that addresses all the courses that the student needs to earn a degree. CPS plan begins when a student is enrolled in the university. CPS form is used by the students, their assigned Faculty Advisors, Academic Advisors and Senior Secretary to ensure the students to take the courses that are required by their program.

At present, CPS filing process is manual where the students visit their Faculty Advisor and request for draft CPS. Then the Faculty Advisor will fill the electives in the CPS form, which has both Faculty Advisor and student sign. A copy of draft CPS is given to the student and at the same time, forwards that draft CPS to the Academic Advisor for approval. The student may come to update his CPS form in the next semesters till he gets the final CPS approved. In this manual process, the student needs to meet the Advisors in person whenever he needs to update his CPS. The goal of this project is to develop an online web application where the Students, Faculty Advisors, Academic Advisors and Senior Secretary may easily access the CPS and everything will be done online which finally makes all tasks easier and time consuming.

An academic advisor can create a new catalog for an academic year, generate the draft CPS to a student by assigning foundation courses if any, changing approved draft CPS to final CPS, Archive CPS, view the faculty details, student details. The faculty advisor can view the details of all the assigned students, edit the draft CPS form by adding the electives that student wants to take, assigning grades, sign it and save it as an approved draft. Senior Secretary performs audits like for Teaching Assistant, Capstone, Research Assistant and also able to do Archive the final audit CPS. Students can only view, download, and print CPS through their login into the system.

# 2. Requirements

The main purpose of this document is to gather all the requirements and outline them to understand what our website should do. This includes functional requirements, technical requirements and security requirements for the student, Faculty Advisor, Academic Advisor and Senior Secretary.

## 2.1 Requirements for the student

**Functional Requirements for the student:**

1. A Student shall have access to log in using username and password.
2. A Student should be able to view his/her CPS.
3. A Student should be able to print his/her CPS.
4. A Student should be able to download his/her CPS.

**Technical Requirements for the student:**

* 1. The application should provide an option to enter login Id and password.
  2. Login details should be validated from e-services database.
  3. Student should be directed to their homepage displaying information like name, Id, faculty advisor name, CPS.

**3.1** The application should save the CPS of the student in PDF format.

**Security Requirements for the student:**

* + 1. Password field should not display its value.
    2. Value cannot be copied.

**2.1.1** Does not allow a student to edit his/her CPS form.

## 2.2 Requirements for the Faculty Advisor

**Functional Requirements for the Faculty Advisor**:

1. Faculty Advisor shall have access to login using username and password.
2. Faculty Advisor should be able to view CPS of all the assigned students.
3. Faculty Advisor should be able to modify a student CPS by assigning electives.
4. Faculty Advisor should be able to print CPS of a student.
5. Faculty Advisor will change the draft CPS to approved draft CPS.

**Technical Requirements for the Faculty Advisor:**

* 1. The application should provide option to enter login Id and password.
  2. Login details should be validated from e-services database.
  3. The application will direct the faculty to his/her home page by displaying their name, details of assigned students.
  4. There should be text box to search for a student with a student Id or with his/her name.
  5. The search for a student should be validated from “e-services” database.
  6. The application should print the CPS of the student in the PDF format.
  7. The application should access the ‘Candidate Plan of Study’ database to select an elective in a particular major.

**Security Requirements for the Faculty Advisor:**

* + 1. Password field should not display its value.
    2. Value cannot be copied.
    3. Does not allow faculty advisor to edit Core courses of the student.

## 2.3 Requirements for the Academic Advisor

**Business Requirements for the Academic Advisor:**

1. Academic advisor should have access to log in using username and password.
2. Academic advisor should be able to view the assigned faculty.
3. Academic advisor should be able to view the students.
4. Academic advisor should create a new catalog for the academic year.
5. The draft CPS for a student should be generated.
6. Academic advisor should audit the draft CPS.
7. Academic advisor should archive the final audit CPS.

**Technical Requirements for the Academic Advisor:**

* 1. The application should provide an option to enter login Id and password.
  2. Login details should be validated from e-services database.
  3. Application will direct the academic advisor to their homepage.
  4. There should be a text box to search for a faculty with a faculty Id or with his/her name.
  5. The search for a faculty should be validated from “e-services” database.
  6. Hyperlink ‘show students’ displays students under the faculty, details of student and CPS.
  7. There should be a text box to search for a student with a student Id or with his/her name.
  8. The search for a student should be validated from ‘e-services’ database.
  9. It should be able to view CPS of the students and their details.
  10. The user should select a major and a year to change the catalogue.
  11. The user can able to view the previous year catalogue or create a new catalogue for the new academic year.
  12. The courses are fetched from “catalogue” table of “Candidate Plan of Study” database.
  13. The user shall able to assign the course as Foundation or Core or Elective by selecting a dropdown list.
  14. It should display the course Id, course names in a selected major, electives, foundations, core courses.
  15. The user should be able to save the catalogue to the new academic year in the ‘candidateplanofstudy’ database.
  16. The user should select a year and a major from the drop-down list.
  17. Selected major and year are validated from “e-services” database and should display all students and their details.
  18. “View CPS” should display the student details, assigned foundation courses in checkboxes and Core courses.
  19. The foundation courses and Core courses are fetched from “catalogue” table of “Candidate Plan of Study” database.
  20. The user selects the foundation courses and save it to “student draft” table of “Candidate Plan of Study’ database.
  21. There should be a text box to search for a student with a student Id or with his/her name and select a particular audit from a drop down.
  22. The “Submit” button should save the CPS of the student to “Candidate plan of Study” database if the condition satisfies for the selected audit.
  23. There should be a text box to search for a student with a student Id or with his/her name and another text box to specify the reason to archive the student CPS.
  24. The search for a student should be validated from “e-services” database.
  25. The ‘Archive’ should save the CPS of the student into ‘Archive’ table.

**Security Requirements for the Academic Advisor:**

* + 1. Password field should not display its value.
    2. Value cannot be copied.

## 2.4 Requirements for the Senior Secretary

**Functional Requirements for the Senior Secretary:**

1. Secretary shall have access to log in using username and password.
2. Secretary can able to view the students.
3. Secretary should be able to do an audit of the draft CPS.
4. Secretary can archive the final audit CPS.

**Technical Requirements for the Senior Secretary:**

* 1. The application should provide an option to enter login Id and password.
  2. Login details should be validated from e-services database.
  3. Application will direct the Secretary to their homepage.
  4. There should be a text box to search for a student with a student Id or with his/her name.
  5. The search for a student should be validated from ‘e-services’ database.
  6. It should be able to view CPS of the students and their details.
  7. There should be a text box to search for a student with a student Id or with his/her name and select a particular audit from a drop down.
  8. If the condition according to an audit satisfies, then the student CPS and student details should save into ‘Candidate Plan of Study’ database.
  9. There should be a text box to search for a student with a student Id or with his/her name and another text box to specify the reason to archive the student CPS.
  10. The search for a student should be validated from ‘e-services’ database.
  11. The ‘Archive’ should save the CPS of the student into ‘Archive’ table.

**Security Requirements for the Senior Secretary:**

* + 1. Password field should not display its value.
    2. Value cannot be copied.

# 3. USECASES

**INTRODUCTION:**

**Purpose:**

The main purpose of use case document is to show the interactions between actors and the system. It is the description of how users will perform tasks on your website. Use cases are generally used during the analysis phase of a project to identify the system functionality.

**Scope:**

This use case document shows the entire functionality of online Candidate Plan of Study (CPS). This describes who are the users, how the user interacts with the online CPS system. This online CPS system allows users to view, download, print, generate, and authorize CPS online.

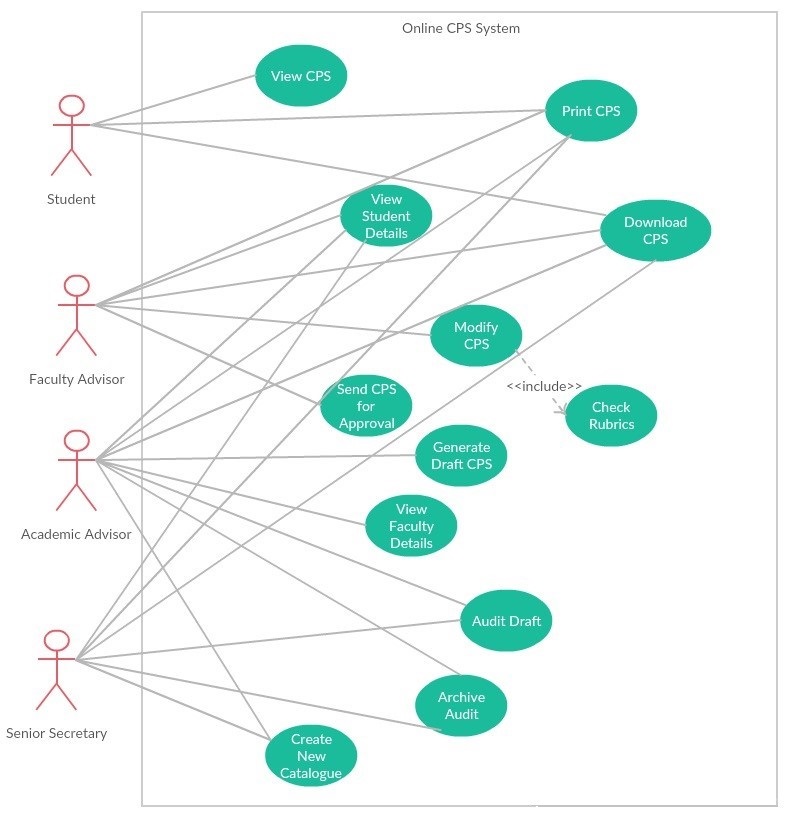
**Definitions, Acronyms, Abbreviations:**

1. **Candidate Plan of Study:** CPS is a degree plan that lists all the courses required to obtain a degree.
2. **Student:** Student is an actor who gets admission into the university degree plan.
3. **Faculty Advisor:** Faculty advisor is an actor who is assigned to the students by academic advisor after students getting admitted into the university.
4. **Academic Advisor:** Academic advisor is an actor who helps the students in pursuing their degree by adapting certain degree plans.
5. **Senior Secretary:** Senior Secretary is an actor who accepts the student requests for capstone, for Teaching Assistance, Research Assistance and checks for their eligibility.

**Actors of Online CPS:**

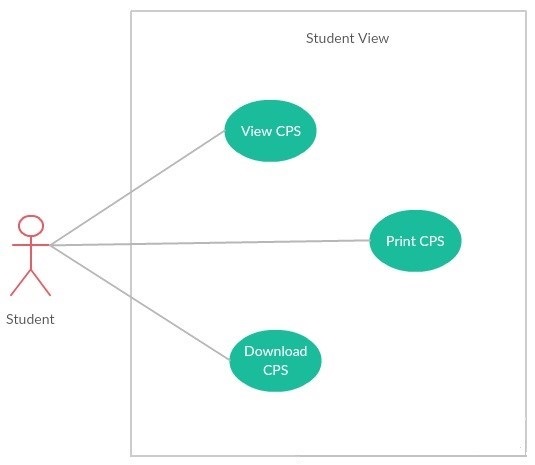
1. **Student:** The primary role of student is to view the CPS, download it and print.
2. **Faculty Advisor:** The primary role of faculty advisor is to assign electives and to modify the student CPS if required.
3. **Academic Advisor:** The primary role of academic advisor is to generate a draft CPS and creating a new catalogue for every academic year.
4. **Senior Secretary:** The primary role of Senior Secretary is to audit draft CPS and archive CPS.

**Cloud-Level Use Case Diagram for Online CPS System:**



**Figure 1: Cloud-Level Use Case for Online CPS System**

## 3.1 Use Case for Student

****

**Figure 2: Use Case Diagram for Student**

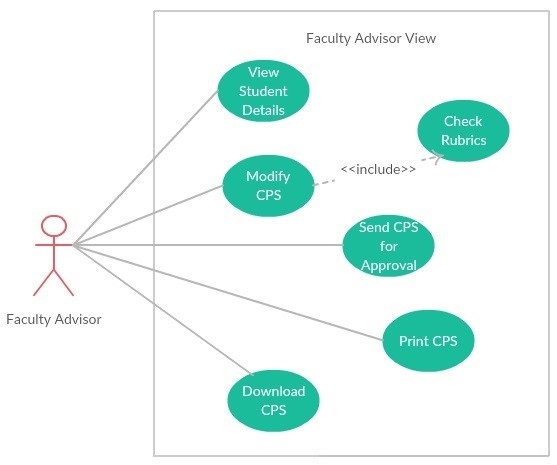
**For students,**

|  |  |
| --- | --- |
| Name | View CPS |
| Brief Description | Student can login and view his/her CPS. |
| Basic Flow | 1. Student will login in to the application. 2. Click view CPS. 3. The draft or final CPS will be displayed. |
| Pre-condition | Student has to login to view his/her CPS. |
| Post condition | Student can download the CPS after viewing it. |
| Special requirements | The application should allow the authorized students only to login and view their CPS. |

|  |  |
| --- | --- |
| Name | Download CPS |
| Brief Description | Student can login and download the CPS. |
| Basic Flow | 1. Student will login in to the application. 2. Click view CPS. 3. Download the CPS if needed. |
| Alternate Flow | If the students do not want to download their CPS, they can just view their CPS without downloading it. |
| Pre-condition | Student has to login to view his/her CPS. |
| Post condition | Students can logout after downloading their CPS. |

|  |  |
| --- | --- |
| Name | Print CPS |
| Brief Description | Student can login and print the CPS. |
| Basic Flow | 1. Student will login in to the application. 2. Click view CPS. 3. Print the CPS if needed. |
| Alternate Flow | If the students do not want to print their CPS, they can just view their CPS. |
| Pre-condition | Student has to login and view their CPS. |
| Post condition | Students can logout after printing their CPS. |

## 3.2. Use Case for Faculty Advisor



**Figure 3: Use Case Diagram for Faculty Advisor**

**For Faculty advisor,**

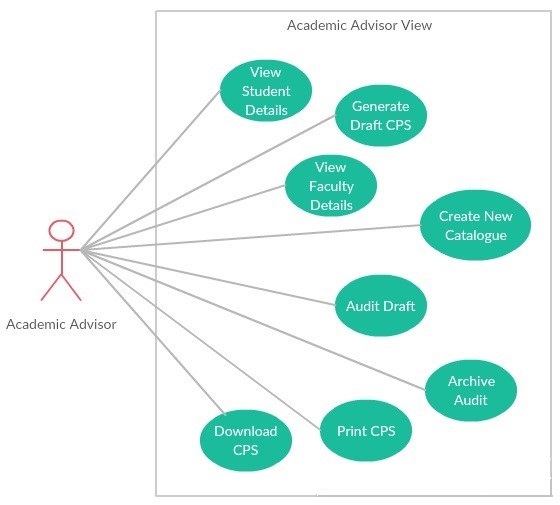
|  |  |
| --- | --- |
| Name | View student details |
| Brief Description | Faculty can login and view the student details assigned to him. |
| Basic Flow | 1. Faculty will login in to the application. 2. Click view student details. |
| Pre-condition | Faculty has to login to view his/her student details. |
| Post condition | Faculty can view the student CPS. |

|  |  |
| --- | --- |
| Name | View CPS |
| Brief Description | Faculty can login and view the student CPS. |
| Basic Flow | 1. Faculty will login in to the application. 2. Click on view student details which displays all the students assigned to him with their student ID. 3. Click on view CPS button to see the CPS of that particular student. |
| Pre-condition | Faculty must view the student details to click on view CPS of the particular student. |
| Post condition | Faculty can modify the student CPS by adding electives to the CPS form. |

|  |  |
| --- | --- |
| Name | Modify CPS |
| Brief Description | Faculty advisor can modify the student CPS after viewing it. |
| Basic Flow | 1. Faculty advisor will login to the application. 2. View the student details and click on view CPS of particular student. 3. Faculty advisor will view the CPS and modify it if there are any changes or if he adds any elective courses. |
| Pre-condition | Faculty advisor must view the CPS to modify it. |
| Post condition | After modifying the CPS, he can send the CPS to academic advisor for his approval if needed. |

|  |  |
| --- | --- |
| Name | Send CPS for approval |
| Brief Description | Faculty advisor send the CPS to academic advisor after modifying it for his/her approval. |
| Basic Flow | 1. Faculty advisor will login to the application. 2. After viewing the CPS of particular student, he will modify it. 3. After modifying the CPS, he will send the CPS for the academic advisor approval and wait for his approval. |
| Pre-condition | Faculty advisor will modify the CPS before sending it to academic advisor for approval. |
| Post condition | Faculty advisor waits for the academic advisor approval. |

## 3.3 Use Case for Academic Advisor



**Figure 4: Use Case Diagram for Academic Advisor**

**For Academic advisor,**

|  |  |
| --- | --- |
| Name | View student details |
| Brief Description | Academic advisor can view the student details after login into the application. |
| Basic Flow | 1. Academic advisor will login to the application. 2. Click on view student details button. 3. He can view the student list with their first name, last name and student Id. |
| Pre-condition | Academic advisor has to login to view the student details. |
| Post condition | 1. Academic advisor can view the student details and may change the faculty advisor assigned to a student if requested. 2. Academic advisor can view the student CPS. |

|  |  |
| --- | --- |
| Name | Modify CPS |
| Brief Description | Academic advisor can modify the final CPS if required. |
| Basic Flow | 1. Academic advisor will login to the application. 2. Click on view student details. 3. Click on view CPS of the particular student. 4. Final CPS with elective courses filled will be displayed. 5. Academic advisor can modify the final CPS if required. |
| Alternate Flow | If there is nothing to change in the final CPS, the academic advisor will leave it without modifying. |
| Pre-condition | Academic advisor has to view the final CPS before modifying it. |
| Post condition | After modifying the final CPS, academic advisor can logout. |

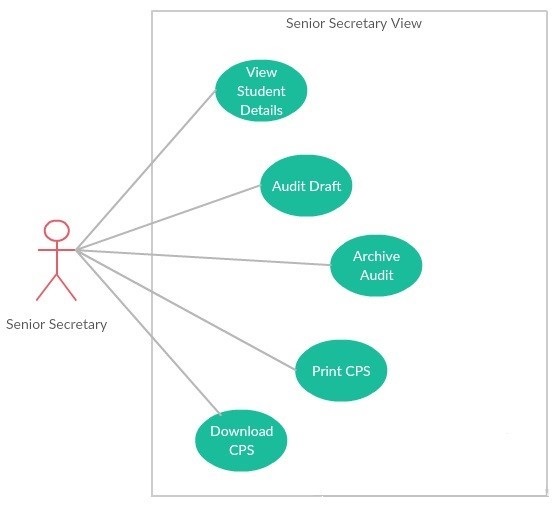
|  |  |
| --- | --- |
| Name | Generate draft CPS |
| Brief Description | Academic advisor generates a draft CPS for each student with core and foundation courses filled in it. |
| Basic Flow | 1. Academic advisor will login to the application. 2. Click on Generate draft CPS button. 3. Academic advisor selects the major and fill the core courses to that particular course. 4. Academic advisor fills the foundation courses to the students if any. |
| Pre-condition | Academic advisor has to fill the core and foundation courses of the students to generate a draft CPS. |
| Post condition | After generating the draft CPS, students can view their generated draft CPS. |

|  |  |
| --- | --- |
| Name | View Faculty details |
| Brief Description | Academic advisor can view the faculty details after login to the application. |
| Basic Flow | 1. Academic advisor will login to the application. 2. Click on view Faculty details button. 3. The academic advisor can view all the faculty details and the students assigned to that particular faculty. |
| Pre-condition | Academic advisor has to login to view the faculty details. |
| Post condition | After viewing the faculty details, the academic advisor may change the faculty advisor of particular student if requested. |

|  |  |
| --- | --- |
| Name | Create new catalogue |
| Brief Description | Academic advisor creates a new catalogue. |
| Basic Flow | 1. Academic advisor will login to the application. 2. Click on new catalogue button. 3. Now, click on view catalogue which displays the previous academic year course catalogue. 4. Then edit the courses and click on save button and select the academic year to which this catalogue has to be saved. |
| Alternate Flow | Can create a new catalogue by clicking on new catalogue button, fill the courses as required (as core, electives, foundation, capstone, and thesis) and save it to the same academic year or save it to the next academic year. |
| Pre-condition | While creating a new catalogue for the first time, academic advisor will create it based on the course catalogue XL sheet. |
| Post condition | Based on the new catalogue created, core courses will be automatically assigned to the students and will be displayed in their CPS. |

|  |  |
| --- | --- |
| Name | Audit draft |
| Brief Description | Academic advisor will audit the CPS for which the student has requested. |
| Basic Flow | 1. Academic advisor will login to the application. 2. Click on Audit draft button. 3. Academic advisor will select a dropdown from Teaching Assistant, Research Assistant, capstone enrolment for which the student has requested. 4. Based on university rules, academic advisor will audit for the eligibility of requested students. |
| Pre-condition | Academic advisor will login to the application and select a dropdown based on student request. |
| Post condition | After auditing it, a message will be displayed as CPS audited. |
| Name | Archive audit |
| Brief Description | Academic advisor archives the CPS. |
| Basic Flow | 1. Academic advisor will login to the application. 2. Click on Archive audit button. 3. Search for a student by using student Id or first name or last name. 4. Click on archive and enter a valid reason for archiving. 5. The application saves the archived CPS. |
| Alternate Flow | Academic advisor can view the archived CPS by clicking on view archived button. |
| Pre-condition | Academic advisor has to login to the application and search for a student to archive his CPS. |
| Post condition | After archiving, a message will be displayed as CPS archived. |

## 3.4 Use Case for Senior Secretary:

****

**Figure 5: Use Case Diagram for Senior Secretary**

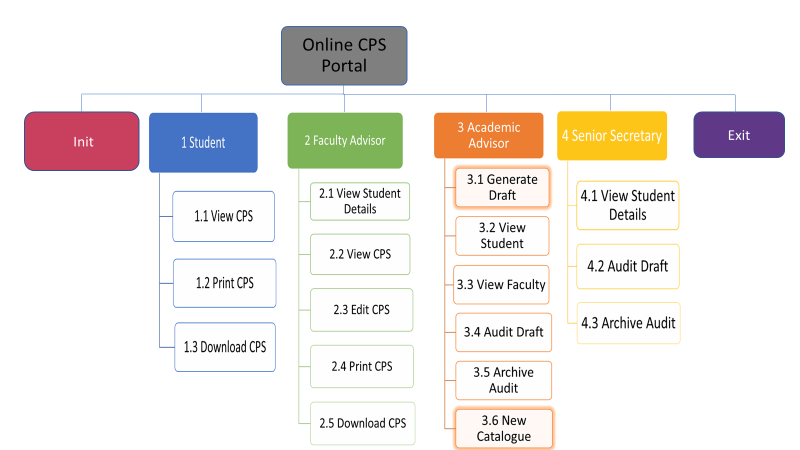
**For Senior Secretary,**

|  |  |
| --- | --- |
| Name | View student details |
| Brief Description | Senior secretary can view the student details after their login into the application. |
| Basic Flow | 1. Senior secretary will login to the application. 2. Senior secretary can search for a student by using student id or first name or last name. 3. Senior secretary can view the student details and his CPS. |
| Pre-condition | Senior secretary has to login to view the student details. |
| Post condition | After viewing, Senior secretary can audit the student CPS or he can also archive the CPS. |

|  |  |
| --- | --- |
| Name | Audit draft |
| Brief Description | Senior Secretary will look for the student requests of Teaching Assistant, Research Assistant, and capstone enrolment and check their eligibility according to the university rules. |
| Basic Flow | 1. Senior Secretary will login to the application. 2. Senior Secretary will select an option from dropdown like Teaching Assistant, Research Assistant, and capstone enrolment for which the student has requested. 3. Based on the university rules, senior secretary will audit for the eligibility of students for what they have requested. |
| Pre-condition | Senior Secretary has to select a dropdown for which the student has requested. |
| Post condition | After auditing, a message will be displayed to senior secretary whether the student is eligible for what he has requested or not. |

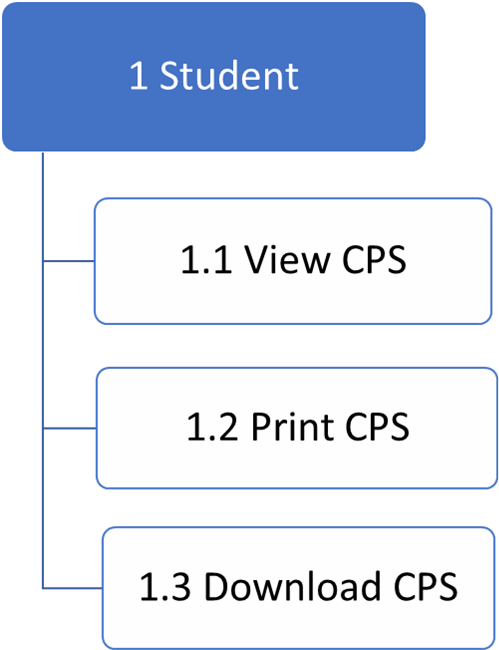
|  |  |
| --- | --- |
| Name | Archive audit |
| Brief Description | Senior Secretary will archive the student CPS. |
| Basic Flow | 1. Senior Secretary will login to the application. 2. Senior Secretary will search for a student and archive his CPS. 3. Senior Secretary will enter a valid reason for archiving the student CPS. |
| Alternate Flow | Senior Secretary can also view the archived CPS of the students by clicking view archived button. |
| Pre-condition | Senior Secretary has to search for a student whose CPS is to be archived. |
| Post condition | After archiving, a message will be displayed as CPS archived. |

# 4.Task Analysis



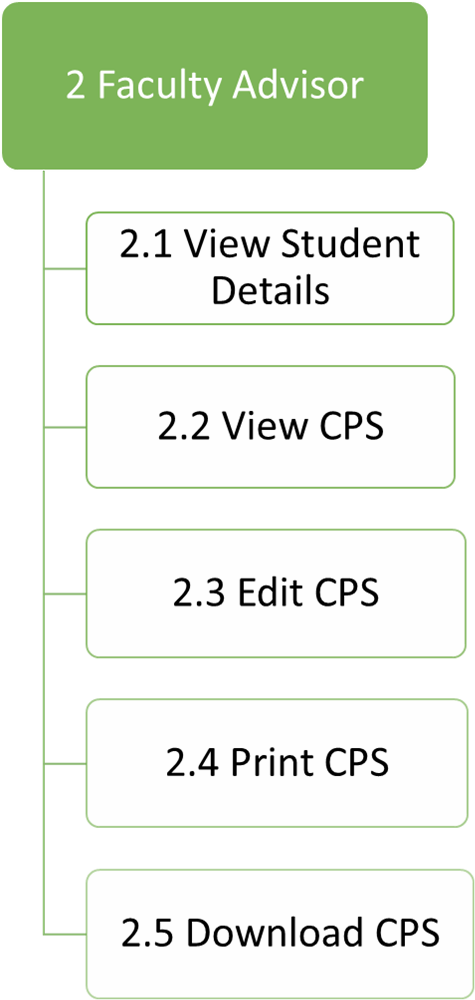
**Figure 6: Task Analysis for Online CPS System**

## 4.1Task Analysis for Student



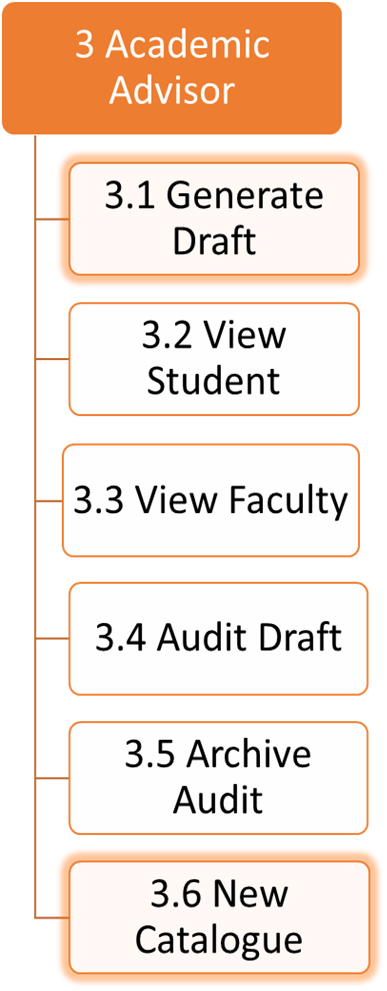
**Figure 7: Task Analysis for Student**

## 4.2Task Analysis for Faculty Advisor



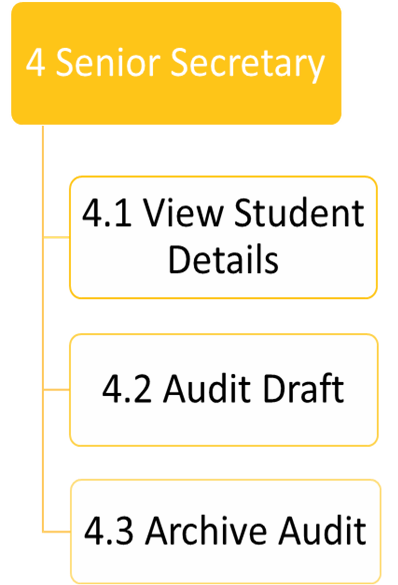
**Figure 8: Task Analysis for Faculty Advisor**

## 4.3Task Analysis for Academic Advisor

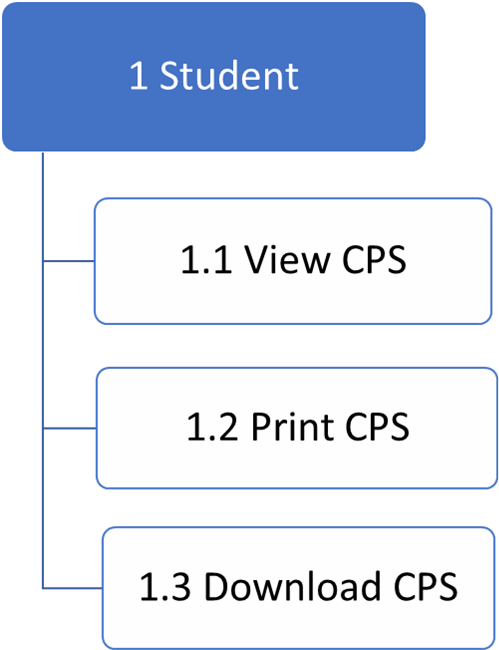


**Figure 9: Task Analysis for Academic Advisor**

## 4.4Task Analysis for Senior Secretary



**Figure 10: Task Analysis for Senior Secretary**



1.1 View CPS

1. **What is the goal of this task?**

The goal of this task is to allow the Student to View the CPS which is helpful in changing and printing the CPS.

1. **What are sub-tasks defining this task?**

There are no sub-tasks that define this task.

1. **Is this task a subset of another task?**

No, this task is not a subset of another task.

1. **What non-user interface functions does this task require?**

Student will provide with draft CPS and final CPS if available.

1. **What kind of input or actions does this task require from the user?**

The inputs that are required to perform this task from the user are student need to login into the system and can View CPS by clicking on View CPS hyperlink.

1. **What kind of output or results occurs by virtue of performing this task?**

The Student can successfully View the CPS.

1. **What automatic actions does this task expect from the system?**

The system should display the Student CPS requested matches with the database.

1. **What special characteristics of this task should we record?**

This task doesn’t require any special characteristics to record.

1. **In this sub tree, is there a task that must come before?**

No, there is no sub-task which comes immediately before.

1. **In this sub tree, is there a task for which this one is required to be immediately preceding? Is there any specific sequence in which the task must be performed?**

No, as this is independent task there is no specific sequence to be performed.

1. **Which if any primary entities (things) are involved in this sub-task? (Note: Are these listed as primary entities from the use case analysis?)**

The primary entities that are involved in this task are CPS form and student details which includes name, ID, major, and academic year.

1. **How can this task fail?**

This task fails when Faculty Advisor or Academic Advisor updated the CPS but Student cannot be able to View updated CPS.

1. **How frequently is this task performed?**

This task is performed when Student wants to View CPS.

1. **How open is this task, especially in terms of its sequence or inputs?**

This task is constrained such that Student can just View the CPS which are accessible up to that date.

1. **What if any are the specific usability expectations (e.g. ease of use, ease of learning) for this task and how do we anticipate determining if we have satisfied the expectations?**

The system should be user-friendly, so that Student should be able to View the CPS with the ease of use.

1.2 Print CPS

1. **What is the goal of this task?**

The goal of this task is to allow the Student to Print the CPS which will be useful in presenting the CPS for various applications, for example, Research Assistant, Teaching Assistant which is generated by Faculty Advisor and Academic Advisor.

1. **What are sub-tasks defining this task?**

There are no sub-tasks that define this task.

1. **Is this task a subset of another task?**

No, this task is not subset of another task.

1. **What non-user interface functions does this task require?**

This task requires displaying the Student CPS form.

1. **What kind of input or actions does this task require from the user?**

The inputs that are required to perform this task from the user are student need to login into the system and can Print CPS after clicking on View CPS hyperlink.

1. **What kind of output or results occurs by virtue of performing this task?**

After successfully login and viewing the CPS, Student can be able to Print the CPS form.

1. **What automatic actions does this task expect from the system?**

The system should Print the CPS when Student views the CPS.

1. **What special characteristics of this task should we record?**

This task doesn’t require any special characteristics to record.

1. **In this sub tree, is there a task that must come before?**

Yes, there is a task called View CPS that must come before this particular task.

1. **In this sub tree, is there a task for which this one is required to be immediately preceding? Is there any specific sequence in which the task must be performed?**

No, as this is independent task there is no specific sequence to be performed.

1. **Which if any primary entities (things) are involved in this sub-task? (Note: Are these listed as primary entities from the use case analysis?)**

The primary entities that are involved in this task are CPS form and student details which includes name, ID, major, and academic year.

1. **How can this task fail?**

This task fails when Faculty Advisor or Academic Advisor updated the CPS but Student cannot be able to View updated CPS and cannot be able to Print the available CPS.

1. **How frequently is this task performed?**

This task is performed when Student wants to Print the CPS.

1. **How open is this task, especially in terms of its sequence or inputs?**

This task is performed when CPS is available.

1. **What if any are the specific usability expectations (e.g. ease of use, ease of learning) for this task and how do we anticipate determining if we have satisfied the expectations?**

The system should be user-friendly, so that Student will be able to Print the CPS with the ease of use.

1.3 Download CPS

1. **What is the goal of this task?**

The goal of this task is to allow the Student to Download the CPS.

1. **What are sub-tasks defining this task?**

There are no sub-tasks that define this task.

1. **Is this task a subset of another task?**

No, this task is not subset of another task.

1. **What non-user interface functions does this task require?**

This task requires displaying the Student CPS form.

1. **What kind of input or actions does this task require from the user?**

The inputs that are required to perform this task from the user are student need to login into the system and can Download CPS after clicking on View CPS hyperlink.

1. **What kind of output or results occurs by virtue of performing this task?**

After successfully login and viewing the CPS, Student can be able to Download the CPS form.

1. **What automatic actions does this task expect from the system?**

The system should Download the CPS when Student views the CPS.

1. **What special characteristics of this task should we record?**

This task doesn’t require any special characteristics to record.

1. **In this sub tree, is there a task that must come before?**

Yes, there is a task called View CPS that must come before this particular task.

1. **In this sub tree, is there a task for which this one is required to be immediately preceding? Is there any specific sequence in which the task must be performed?**

No, as this is independent task there is no specific sequence to be performed.

1. **Which if any primary entities (things) are involved in this sub-task? (Note: Are these listed as primary entities from the use case analysis?)**

The primary entities that are involved in this task are CPS form and student details which includes name, ID, major, and academic year.

1. **How can this task fail?**

This task fails when Faculty Advisor or Academic Advisor updated the CPS but Student cannot be able to View updated CPS and cannot be able to Download the available CPS.

1. **How frequently is this task performed?**

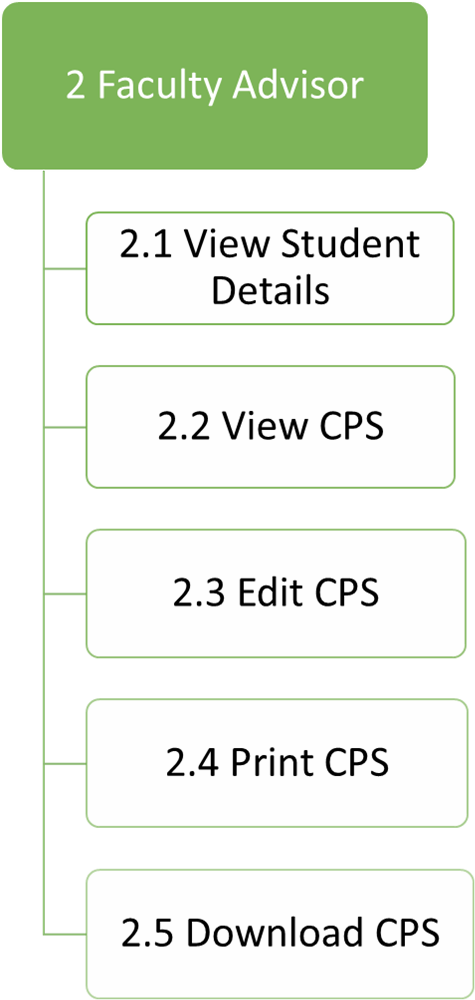
This task is performed when Student wants to Download the CPS.

1. **How open is this task, especially in terms of its sequence or inputs?**

This task is performed when CPS is available.

1. **What if any are the specific usability expectations (e.g. ease of use, ease of learning) for this task and how do we anticipate determining if we have satisfied the expectations?**

The system should be user-friendly, so that Student will be able to Download the CPS with the ease of use.



2.1 View Student Details

1. **What is the goal of this task?**

The goal of this task is to allow the Faculty Advisor to View the Student Details.

1. **What are sub-tasks defining this task?**

There are no sub-tasks that define this task.

1. **Is this task a subset of another task?**

No, this task is not a subset of another task.

1. **What non-user interface functions does this task require?**

The Faculty Advisor will be provided with all his/her assigned Student’s information.

1. **What kind of input or actions does this task require from the user?**

The inputs that are required to perform this task are Faculty Advisor need to login into the system to View all Student’s details.

1. **What kind of output or results occurs by virtue of performing this task?**

The Faculty Advisor will be provided with details of Student after successfully performing this task.

1. **What automatic actions does this task expect from the system?**

The system should display only the particular student details whose student name or identification number matches with the database.

1. **What special characteristics of this task should we record?**

This task doesn’t require any special characteristics to record.

1. **In this sub tree, is there a task that must come before?**

No, there is no task which comes immediately before002E

1. **In this sub tree, is there a task for which this one is required to be immediately preceding? Is there any specific sequence in which the task must be performed?**

No, as this is independent task there is no specific sequence to be performed.

1. **Which if any primary entities (things) are involved in this sub-task? (Note: Are these listed as primary entities from the use case analysis?)**

The primary entities that are involved in this task are student details which includes name, ID, major, and academic year.

1. **How can this task fail?**

This task fails when requested student details such as student name and identification number are not available in the database.

1. **How frequently is this task performed?**

This task is performed by the Faculty Advisor when student is assigned to them by Academic Advisor and when they make request for CPS or to modify the CPS.

1. **How open is this task, especially in terms of its sequence or inputs?**

This task is constrained such that Faculty Advisor will be able to see the Student details who are assigned to him/her and can view the student details only by Student name or identification number.

1. **What if any are the specific usability expectations (e.g. ease of use, ease of learning) for this task and how do we anticipate determining if we have satisfied the expectations?**

The system should be user-friendly and Faculty Advisor experiences ease of use for seeing Student details by just using student name or identification number.

2.2 View CPS

1. **What is the goal of this task?**

The goal of this task is to allow the Faculty Advisor to View the CPS which is helpful in updating and printing the CPS.

1. **What are sub-tasks defining this task?**

There are no sub-tasks that define this task.

1. **Is this task a subset of another task?**

No, this task is not a subset of another task.

1. **What non-user interface functions does this task require?**

Faculty Advisor will be provided with draft CPS and final CPS if available.

1. **What kind of input or actions does this task require from the user?**

The inputs that are required to perform this task are Faculty Advisor need to login into the system and can View CPS after searching particular student and then clicking on View CPS hyperlink.

1. **What kind of output or results occurs by virtue of performing this task?**

The Faculty Advisor can successfully View the CPS.

1. **What automatic actions does this task expect from the system?**

The system should display the Student CPS requested matches with the database.

1. **What special characteristics of this task should we record?**

This task doesn’t require any special characteristics to record.

1. **In this sub tree, is there a task that must come before?**

No, there is no sub-task which comes immediately before.

1. **In this sub tree, is there a task for which this one is required to be immediately preceding? Is there any specific sequence in which the task must be performed?**

No, as this is independent task there is no specific sequence to be performed.

1. **Which if any primary entities (things) are involved in this sub-task? (Note: Are these listed as primary entities from the use case analysis?)**

The primary entities that are involved in this task are CPS form and student details which includes name, ID, major, and academic year.

1. **How can this task fail?**

This task fails when Faculty Advisor or Academic Advisor updated the CPS but Faculty Advisor cannot be able to View updated CPS.

1. **How frequently is this task performed?**

This task is performed when Faculty Advisor wants to View CPS of particular student.

1. **How open is this task, especially in terms of its sequence or inputs?**

This task is done only when Draft cps is generated and available for particular student.

1. **What if any are the specific usability expectations (e.g. ease of use, ease of learning) for this task and how do we anticipate determining if we have satisfied the expectations?**

The system provides Student CPS easily for the Faculty Advisor by just clicking on View CPS link, so this depicts ease of use of this task.

2.3 Edit CPS

1. **What is the goal of this task?**

The goal of this task is to allow the Faculty Advisor to Edit the CPS which is helpful in updating the CPS.

1. **What are sub-tasks defining this task?**

There are no sub-tasks that define this task.

1. **Is this task a subset of another task?**

No, this task is not a subset of another task.

1. **What non-user interface functions does this task require?**

Faculty Advisor will be provided with draft CPS and final CPS if available.

1. **What kind of input or actions does this task require from the user?**

The inputs that are required to perform this task are Faculty Advisor need to login into the system and can Edit CPS after viewing the CPS.

1. **What kind of output or results occurs by virtue of performing this task?**

The Faculty Advisor can successfully Edit the CPS and can update the CPS.

1. **What automatic actions does this task expect from the system?**

The system should display the Student CPS requested matches with the database.

1. **What special characteristics of this task should we record?**

This task doesn’t require any special characteristics to record.

1. **In this sub tree, is there a task that must come before?**

Yes, there is a task View CPS that must be performed before editing the CPS.

1. **In this sub tree, is there a task for which this one is required to be immediately preceding? Is there any specific sequence in which the task must be performed?**

No, as this is independent task there is no specific sequence to be performed.

1. **Which if any primary entities (things) are involved in this sub-task? (Note: Are these listed as primary entities from the use case analysis?)**

The primary entities that are involved in this task are CPS form and student details which includes name, ID, major, courses that student enrolled and academic year.

1. **How can this task fail?**

This task fails when Faculty Advisor is unable to view the CPS and also when requested course is not with in the Rubric.

1. **How frequently is this task performed?**

This task is performed by Faculty Advisor when Student requests to edit the CPS.

1. **How open is this task, especially in terms of its sequence or inputs?**

This task is done only when CPS is already generated and available for particular student.

1. **What if any are the specific usability expectations (e.g. ease of use, ease of learning) for this task and how do we anticipate determining if we have satisfied the expectations?**

The system provides Student CPS easily for the Faculty Advisor by just clicking on View CPS link, so this depicts ease of use of this task and can easily edit CPS by clicking on edit link.

2.4 Print CPS

1. **What is the goal of this task?**

The goal of this task is to allow the Faculty Advisor to Print the CPS, so that it will be helpful for the Student to submit the CPS for various applications, for example, Teaching Assistant, Research Assistant.

1. **What are sub-tasks defining this task?**

There are no sub-tasks that define this task.

1. **Is this task a subset of another task?**

No, this task is not subset of another task.

1. **What non-user interface functions does this task require?**

This task requires displaying the Student CPS form.

1. **What kind of input or actions does this task require from the user?**

The inputs that are required to perform this task are Faculty Advisor needs to login into the system and can Print CPS after clicking on View CPS hyperlink.

1. **What kind of output or results occurs by virtue of performing this task?**

After successfully login and viewing the CPS, Faculty Advisor can be able to Print the CPS form.

1. **What automatic actions does this task expect from the system?**

The system should Print the CPS when Faculty Advisor views the CPS.

1. **What special characteristics of this task should we record?**

This task doesn’t require any special characteristics to record.

1. **In this sub tree, is there a task that must come before?**

Yes, there is a task View CPS that must come before this particular task.

1. **In this sub tree, is there a task for which this one is required to be immediately preceding? Is there any specific sequence in which the task must be performed?**

No, as this is independent task there is no specific sequence to be performed.

1. **Which if any primary entities (things) are involved in this sub-task? (Note: Are these listed as primary entities from the use case analysis?)**

The primary entities that are involved in this task are CPS form and student details which includes name, ID, major, and academic year.

1. **How can this task fail?**

This task fails when Faculty Advisor is unable to view student CPS.

1. **How frequently is this task performed?**

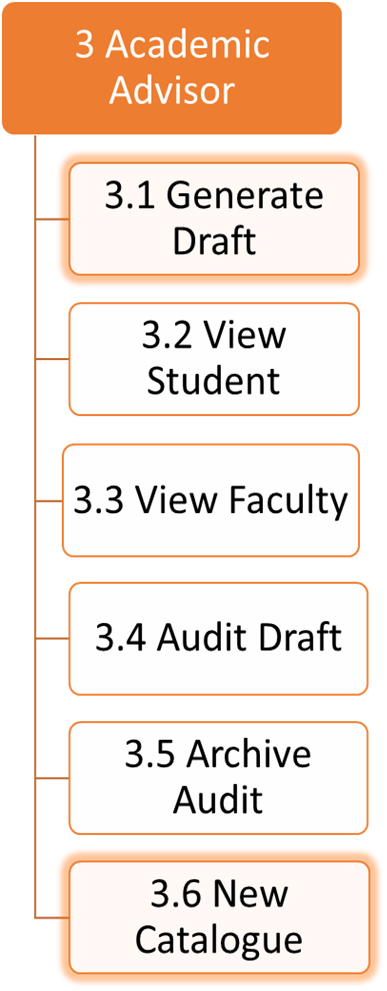
This task is performed when Faculty Advisor wants to Print the CPS.

1. **How open is this task, especially in terms of its sequence or inputs?**

This task is performed only when CPS is generated and available.

1. **What if any are the specific usability expectations (e.g. ease of use, ease of learning) for this task and how do we anticipate determining if we have satisfied the expectations?**

The system should be user-friendly, so that Faculty Advisor will be able to Print the CPS with the ease of use.



3.1 Generate Draft CPS

1. What is the goal of this task?

The goal of this task is to allow the Academic Advisor to Generate Draft CPS.

1. What sub-tasks define this task?

There are no sub-tasks that define this task.

1. Is this task a subset of another task?

Yes, this task is a subset of “Academic Advisor” task.

1. What non-user interface functions does this task require?

This task requires Student details like in which major they have enrolled in the University.

1. What kind of input or actions does this task require from the user?

Academic Advisor must be logged into the system and must have Student details like Student name, ID, major and in which year they are admitted in order to perform this task.

1. What kind of output or results occurs by virtue of performing this task?

By successfully performing this task, Academic Advisor will be able to generate Draft CPS for Students who are enrolled.

1. What automatic actions does this task expect from the system?

The system should generate Draft CPS of the specific Student whose database matches with the enrolled Students.

1. What special characteristics of this task should we record?

This task requires to record the generated initial CPS.

1. In this sub tree, is there a task that must come before this one?

The task that involves before this is Adding of courses by Student.

1. In this sub tree, is there a task for which this one is required to be immediately preceding? Is there any specific sequence in which the task must be performed?

The task it requires is generating course catalogue.

1. Which if any primary entities (things) are involved in this sub-task? (Note: Are these listed as primary entities from the use case analysis?)

The primary entities associated with this task are Student, Course Catalogue and CPS records.

1. How can this task fail?

This task fails when the Academic Advisor doesn’t have enough Student details to generate initial draft CPS.

1. How frequently is this task performed?

The Academic Advisor will perform this task for when the Student first enrolls into the University.

1. How open is this task, especially in terms of its sequence or inputs?

This task is simple with sequence of inputs like selecting semester, major and checking foundations and finally saving the changes.

1. What if any are the specific usability expectations (e.g. ease of use, ease of learning) for this task and how do we anticipate determining if we have satisfied the expectations?

In generating the Draft CPS, the Academic Advisor needs to have prior knowledge. As the Academic Advisor needs to provide with the foundation courses and the major in which Student is enrolled.

3.2 View Student

1. **What is the goal of this task?**

The goal of this task is to allow the Academic Advisor to view the Student details which are in the University records.

1. **What sub-tasks define this task?**

There are no sub-tasks that define this task.

1. **Is this task a subset of another task?**

No, this task is not a subset of any other task.

1. **What non-user interface functions does this task require?**

Academic Advisor will be provided with list of Students by entering Term and Major and their details who are currently enrolled in the program.

1. **What kind of input or actions does this task require from the user?**

The inputs that are required to perform this task from the user are user need to login in the application and can view Student details by entering particular Term and Major.

1. **What kind of output or results occurs by virtue of performing this task?**

The Academic Advisor can successfully view the Student details who are enrolled in particular major and term.

1. **What automatic actions does this task expect from the system?**

The system should display only the Students who are enrolled in particular term and major requested matches with the database.

1. **What special characteristics of this task should we record?**

This task doesn’t require any special characteristics to record.

1. **In this sub tree, is there a task that must come before?**

No, there is no sub-task which comes immediately before.

1. **In this sub tree, is there a task for which this one is required to be immediately preceding? Is there any specific sequence in which the task must be performed?**

No, as this is independent task there is no specific sequence to be performed.

1. **Which if any primary entities (things) are involved in this sub-task? (Note: Are these listed as primary entities from the use case analysis?)**

The primary entities that are involved in this task are Student records and Student details.

1. **How can this task fail?**

This task fails when requested Student data is not available in the database or a Student is not enrolled in the university and if a Student is transferred from another university and the data is not updated frequently.

1. **How frequently is this task performed?**

This task is performed when Academic Advisor wants to view particular Student CPS and their details.

1. **How open is this task, especially in terms of its sequence or inputs?**

This task is easy with simple input of searching Student with Id or name.

1. **What if any are the specific usability expectations (e.g. ease of use, ease of learning) for this task and how do we anticipate determining if we have satisfied the expectations?**

This task is very easy to perform as searching of Student with id or name will display the Student details.

3.3 View Faculty

1. **What is the goal of this task?**

The goal of this task is to allow the Academic Advisor to view the Faculty details which are in the University records.

1. **What sub-tasks define this task?**

There are no sub-tasks that define this task.

1. **Is this task a subset of another task?**

No, this task is not a subset of any other task.

1. **What non-user interface functions does this task require?**

Academic Advisor will be provided with list of faculties by entering Faculty name and Major and the Student list will appear who are currently assigned to that Faculty in the program.

1. **What kind of input or actions does this task require from the user?**

The inputs that are required to perform this task from the user are user need to login in the application and can view Faculty and assigned Students to that Faculty details by entering particular name of Faculty and Major.

1. **What kind of output or results occurs by virtue of performing this task?**

The Academic Advisor can successfully view the Faculty details who are enrolled in particular major and term.

1. **What automatic actions does this task expect from the system?**

The system should display the Faculty and Students who are assigned to that Faculty in particular term and major requested matches with the database.

1. **What special characteristics of this task should we record?**

This task doesn’t require any special characteristics to record.

1. **In this sub tree, is there a task that must come before?**

No, there is no sub-task which comes immediately before.

1. **In this sub tree, is there a task for which this one is required to be immediately preceding? Is there any specific sequence in which the task must be performed?**

No, as this is independent task there is no specific sequence to be performed.

1. **Which if any primary entities (things) are involved in this sub-task? (Note: Are these listed as primary entities from the use case analysis?)**

The primary entities that are involved in this task are Faculty records and Student details.

1. **How can this task fail?**

This task fails when requested Faculty data is not available in the database or a Student is not enrolled in the university and if a Faculty is incorrectly assigned to any Student and the data is not updated frequently.

1. **How frequently is this task performed?**

This task is performed when Academic Advisor wants to view particular Faculty and their assigned Students.

1. **How open is this task, especially in terms of its sequence or inputs?**

This task is easy with simple input of searching with Faculty name.

1. **What if any are the specific usability expectations (e.g. ease of use, ease of learning) for this task and how do we anticipate determining if we have satisfied the expectations?**

The system should be user-friendly, so that Academic Advisor should be able to easily view a Student with the ease of use.

3.4 Audit Draft

1) What is the goal of this task?

The goal of this task is to allow the Academic Advisor to Audit changes that are done to draft CPS.

1. What sub-tasks define this task?

There are no sub-tasks that define this task.

1. Is this task a subset of another task?

Yes, this task is a subset of “Academic Advisor” task.

1. What non-user interface functions does this task require?

The changes that are done by the Faculty Advisor will be stored in database and they are reflected.

1. What kind of input or actions does this task require from the user?

Academic Advisor must be logged into the system and must have Student details like Student name, ID, major and in which year they are admitted in order to perform this task.

1. What kind of output or results occurs by virtue of performing this task?

By successfully performing this task, Academic Advisor will be able to view the audited changes that are made to the draft CPS.

1. What automatic actions does this task expect from the system?

The system should audit all the changes that are performed to Draft CPS of the specific Student whose database matches with the enrolled Students.

1. What special characteristics of this task should we record?

This task requires to record the changes done by the Faculty Advisor.

1. In this sub tree, is there a task that must come before this one?

The task that involves before this is generate draft and all the changes that the Faculty Advisor perform to CPS

1. In this sub tree, is there a task for which this one is required to be immediately preceding? Is there any specific sequence in which the task must be performed?

The task it requires to be finished to perform archive audit.

1. Which if any primary entities (things) are involved in this sub-task? (Note: Are these listed as primary entities from the use case analysis?)

The primary entities related to this task are Student, CPS records.

1. How can this task fail?

This task fails when the Academic Advisor doesn’t save changes that are made to draft CPS and if Academic Advisor cannot view the changes made by the Faculty Advisor.

1. How frequently is this task performed?

The Academic Advisor will perform this task when the Student requests the final CPS for applying Teaching Assistant or Research Assistant posts and request the final CPS.

1. How open is this task, especially in terms of its sequence or inputs?

This task is simple with sequences of input from Academic Advisor to approve a Student final CPS.

1. What if any are the specific usability expectations (e.g. ease of use, ease of learning) for this task and how do we anticipate determining if we have satisfied the expectations?

This task is user friendly representation. It is very easy to use as all the courses of each department would be visible.

3.5 Archive Audit

1. What is the goal of this task?

The goal of this task is to allow the Academic Advisor to archive graduate Students and alumni Student CPS details.

1. What sub-tasks define this task?

There are no sub-tasks that define this task.

1. Is this task a subset of another task?

Yes, this task is a subset of “Academic Advisor” task.

1. What non-user interface functions does this task require?

This task requires to save the graduated Student CPS details and alumni CPS details to the database.

1. What kind of input or actions does this task require from the user?

Academic Advisor must be logged into the system and must have Student details like Student name, ID, major and in which year they are admitted in order to perform this task.

1. What kind of output or results occurs by virtue of performing this task?

By successfully performing this task, Academic Advisor will be able to separate the CPS details of current Students and graduated Students and also, he can save all the CPS data of all alumni Students.

1. What automatic actions does this task expect from the system?

The system should save all the final CPS details of the graduated Students into separate database from that of the current Student CPS details.

1. What special characteristics of this task should we record?

This task requires to record the audit draft CPS details.

1. In this sub tree, is there a task that must come before this one?

The task that involves before this is generate draft, audit draft.

1. In this sub tree, is there a task for which this one is required to be immediately preceding? Is there any specific sequence in which the task must be performed?

The task it requires is auditing the generated CPS draft.

1. Which if any primary entities (things) are involved in this sub-task? (Note: Are these listed as primary entities from the use case analysis?)

The primary entities related to this task are Student records and CPS records.

1. How can this task fail?

This task fails when the Academic Advisor cannot audit the changes to the CPS and also the task fails if the CPS details of a Student are not correctly stored in the database.

1. How frequently is this task performed?

The Academic Advisor will perform this task for when the Student graduates from the university.

1. How open is this task, especially in terms of its sequence or inputs?

This task requires simple sequence of inputs from the Secretary by archiving the graduated Student details to the database.

1. What if any are the specific usability expectations (e.g. ease of use, ease of learning) for this task and how do we anticipate determining if we have satisfied the expectations?

In archiving the audit, the Academic Advisor need not have any knowledge as he can easily distinguish the current Student for the graduated Students to save the CPS details.

3.6 New Catalogue

1. What is the goal of this task?

The goal of this task is to generate new course catalogue for each department.

1. What sub-tasks define this task?

This task does not define any other sub-tasks.

1. Is this task a subset of another task?

Yes. This task is a subset of Academic Advisor task.

1. What non-user interface functions does this task require?

There are no non-user interface functions for this task.

1. What kind of input or actions does this task require from the user?

The user needs to login into the system and should be able to access the courses list of all departments.

1. What kind of output or results occurs by virtue of performing this task?

By performing this task, the user will be able to generate new course catalogue for each department for particular year.

1. What automatic actions does this task expect from the system?

This task does not expect any automatic actions from the system.

1. What special characteristics of this task should we record?

This task does not record any special characteristics.

1. In this sub tree, is there a task that must come before?

No, there is no sub-task that comes before.

1. In this sub tree, is there a task for which this one is required to be immediately preceding? Is there any specific sequence in which the task must be performed?

There is no task such that this one is required to be immediately preceding.

1. Which if any primary entities (things) are involved in this sub-task? (Note: Are these listed as primary entities from the use case analysis?)

The primary entities are departments list and their courses.

1. How can this task fail?

This task fails when there is a problem in retrieving the courses that are recorded in the database.

1. How frequently is this task performed?

This task is performed when the Academic Advisor wants to change a course catalogue for particular department for particular year.

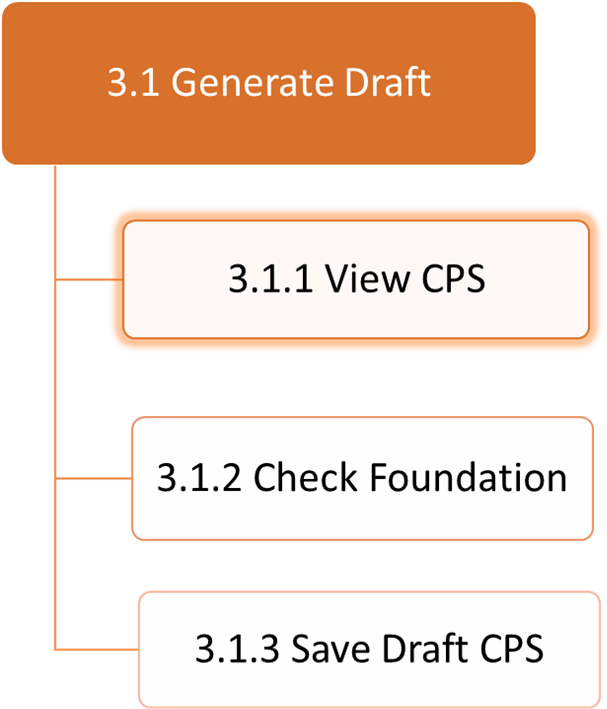
1. How open is this task, especially in terms of its sequence or inputs?

This task add catalogue can be done by Academic Advisor.

1. What if any are the specific usability expectations (e.g. ease of use, ease of learning) for this task and how do we anticipate determining if we have satisfied the expectations?

This task is user friendly representation. It is very easy to use as all the courses of each department would be visible.

## Task Analysis for Generate Draft



3.1.1 View CPS

1. What is the goal of this task?

The goal of this task is to allow the Academic Advisor to view CPS of particular semester and major.

1. What sub-tasks define this task?

The subtasks define this task are select semester and select major.

1. Is this task a subset of another task?

Yes, this task is a subset of “Generate Draft” task.

1. What non-user interface functions does this task require?

This task requires to save initial CPS with filled core courses as soon as the Student enrolled in a major.

1. What kind of input or actions does this task require from the user?

Academic Advisor must be logged into the system and once he selects GENERATE DRAFT button, He can view the CPS by selecting semester and major.

1. What kind of output or results occurs by virtue of performing this task?

The Academic Advisor can view the initial CPS of a Student in required semester and major.

1. What automatic actions does this task expect from the system?

The system should display initial CPS of the Student with filled in core courses for a selected Student.

1. What special characteristics of this task should we record?

We should record the initial CPS with core courses.

1. In this sub tree, is there a task that must come before this one?

The select semester and select major sub tasks should come before this one.

1. In this sub tree, is there a task for which this one is required to be immediately preceding? Is there any specific sequence in which the task must be performed?

The task for which this one is required to be immediately preceding is check foundations.

1. Which if any primary entities (things) are involved in this sub-task? (Note: Are these listed as primary entities from the use case analysis?)

There are no primary entities listed for this sub task from the use case analysis.

1. How can this task fail?

This task fails when the system doesn’t display the initial CPS of enrolled Student and if the system displays the CPS of another major different from the major that the Student was enrolled.

1. How frequently is this task performed?

The Academic Advisor will perform this task for when the Student first enrolls into the University.

1. How open is this task, especially in terms of its sequence or inputs?

The view CPS task is performed by the Academic Advisor before generating the draft CPS.

1. What if any are the specific usability expectations (e.g. ease of use, ease of learning) for this task and how do we anticipate determining if we have satisfied the expectations?

This task is easy to use when the Academic Advisor correctly enters the semester and major that he is looking for.

3.1.2 Check Foundation

1. What is the goal of this task?

The goal of this task is to allow the Academic Advisor to select foundation courses for a Student.

1. What sub-tasks define this task?

There are no subtasks defining this task.

1. Is this task a subset of another task?

Yes, this task is a subset of “Generate Draft” task.

1. What non-user interface functions does this task require?

This task requires the database to save foundation courses for different majors to display in CPS, so that Academic Advisor can choose foundation courses for a Student.

1. What kind of input or actions does this task require from the user?

Academic Advisor must be logged into the system and once he selects GENERATE DRAFT button, He can view the CPS and from the drop-down menu, he can view and select from check boxes in foundation courses.

1. What kind of output or results occurs by virtue of performing this task?

The Academic Advisor can assign foundation courses for a Student.

1. What automatic actions does this task expect from the system?

The system should display list of foundations courses for different majors.

1. What special characteristics of this task should we record?

We should record what foundations are checked in generating draft so that they can be viewed by Student.

1. In this sub tree, is there a task that must come before this one?

The view CPS task should come before this one.

1. In this sub tree, is there a task for which this one is required to be immediately preceding? Is there any specific sequence in which the task must be performed?

The task for which this one is required to be immediately preceding is saving the changes for draft CPS. We can save the draft CPS after checking the foundation courses.

1. Which if any primary entities (things) are involved in this sub-task? (Note: Are these listed as primary entities from the use case analysis?)

The primary entities involved in this sub task are course catalogue, foundation courses, and major.

1. How can this task fail?

This task fails when different foundation courses are listed which are not present in that major.

1. How frequently is this task performed?

The Academic Advisor will perform this task for when the Student first enrolls into the University.

1. How open is this task, especially in terms of its sequence or inputs?

This task is clear with clear sequence of steps. Academic Advisor has to check the foundation courses which is an input to this task.

1. What if any are the specific usability expectations (e.g. ease of use, ease of learning) for this task and how do we anticipate determining if we have satisfied the expectations?

This task needs some knowledge on what foundation courses the Academic Advisor needs to assign for a Student and under what conditions they are applied.

3.1.3 Save Draft CPS

1. What is the goal of this task?

The goal of this task is to allow the Academic Advisor to save changes to CPS while generating draft CPS.

1. What sub-tasks define this task?

There are no such tasks defining this task.

1. Is this task a subset of another task?

Yes, this task is a subset of “Generate Draft” task.

1. What non-user interface functions does this task require?

This task requires the database to store the foundation courses that are checked by the Academic Advisor once he clicks on save draft CPS.

1. What kind of input or actions does this task require from the user?

Academic Advisor must be logged into the system and once he selects GENERATE DRAFT button, He can view the CPS by selecting semester and major and once he checks the foundation courses he can save changes.

1. What kind of output or results occurs by virtue of performing this task?

The changes that are performed by Academic Advisor to the draft CPS will be saved.

1. What automatic actions does this task expect from the system?

The system should display the saved changes in CPS for Student and Faculty.

1. What special characteristics of this task should we record?

We should record what changes were done before saving the changes so that we can check whether the changes are correctly saved or not.

1. In this sub tree, is there a task that must come before this one?

The tasks that come before this task are view CPS and check foundation.

1. In this sub tree, is there a task for which this one is required to be immediately preceding? Is there any specific sequence in which the task must be performed?

The task for which this one is required to be immediately preceding is check foundations.

1. Which if any primary entities (things) are involved in this sub-task? (Note: Are these listed as primary entities from the use case analysis?)

There are no primary entities listed for this sub task from the use case analysis.

1. How can this task fail?

This task fails when the system doesn’t save the changes that are made by the Academic Advisor.

1. How frequently is this task performed?

The Academic Advisor will perform this task whenever he makes any changes to the draft CPS.

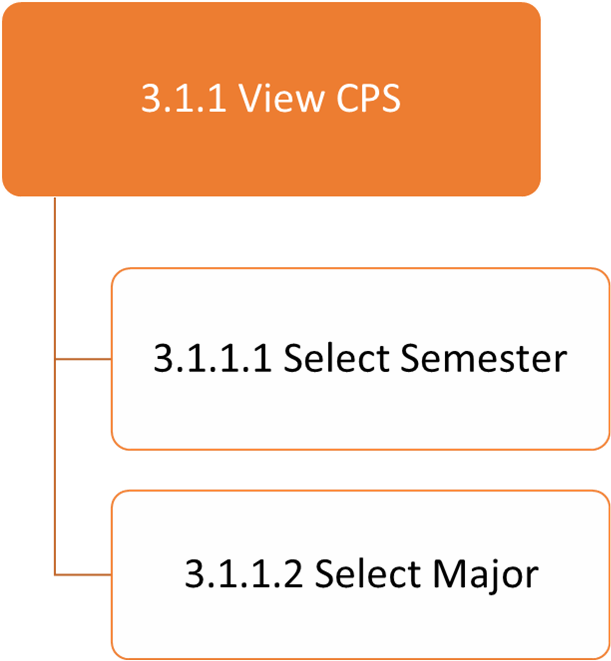
1. How open is this task, especially in terms of its sequence or inputs?

This task is simple sequence with one input from the Academic Advisor.

1. What if any are the specific usability expectations (e.g. ease of use, ease of learning) for this task and how do we anticipate determining if we have satisfied the expectations?

This task does not need any technical knowledge and the only thing Academic Advisor has to do is he has to save changes whenever he makes any changes to the draft CPS.

Task Analysis for View CPS



3.1.1.1 Select Semester

1. What is the goal of this task?

The goal of this task is to allow the Academic Advisor to select semester term.

1. What sub-tasks define this task?

There are no sub-tasks that define this task.

1. Is this task a subset of another task?

Yes, this task is a subset of “View CPS” task.

1. What non-user interface functions does this task require?

This task requires to have different terms listed in the database. So that once the user selects the semester it should show the details related to that term

1. What kind of input or actions does this task require from the user?

Academic Advisor must be logged into the system and once he selects GENERATE DRAFT button, select semester will appear along with select major. He has to enter for which semester he is generating the draft.

1. What kind of output or results occurs by virtue of performing this task?

By selecting the semester, and major, the system will display the Students and their initial CPS to that semester and major.

1. What automatic actions does this task expect from the system?

The system should display all the Students and their initial CPS once he selects the semester and major.

1. What special characteristics of this task should we record?

We should record different terms and majors.

1. In this sub tree, is there a task that must come before this one?

There is no such task that must come before this task.

1. In this sub tree, is there a task for which this one is required to be immediately preceding? Is there any specific sequence in which the task must be performed?

The task for which this one is required to be immediately preceding is select major.

1. Which if any primary entities (things) are involved in this sub-task? (Note: Are these listed as primary entities from the use case analysis?)

The primary entities related to this task are semester records and Student records.

1. How can this task fail?

This task fails when the Academic Advisor is not able to enter required semester.

1. How frequently is this task performed?

The Academic Advisor will perform this task for when the Student first enrolls into the University.

1. How open is this task, especially in terms of its sequence or inputs?

This task is single sequence of input from the Academic Advisor.

1. What if any are the specific usability expectations (e.g. ease of use, ease of learning) for this task and how do we anticipate determining if we have satisfied the expectations?

In selecting the semester, the Academic Advisor need to know the semesters that are currently applicable are.

3.1.1.2 Select Major

1. What is the goal of this task?

The goal of this task is to allow the Academic Advisor to select the major.

1. What sub-tasks define this task?

There are no sub-tasks that define this task.

1. Is this task a subset of another task?

Yes, this task is a subset of “View CPS” task.

1. What non-user interface functions does this task require?

This task requires to have different majors listed in the database. So that once the user selects the major, it should show the Student details in that major.

1. What kind of input or actions does this task require from the user?

Academic Advisor must be logged into the system and once he selects GENERATE DRAFT button, select semester will appear along with select major. He has to enter for which major he is generating the draft.

1. What kind of output or results occurs by virtue of performing this task?

By selecting the major, the system will display the Students and their initial CPS with the core details to that major.

1. What automatic actions does this task expect from the system?

The system should display all the Students and their initial CPS with filled core courses once the Academic Advisor selects the major.

1. What special characteristics of this task should we record?

We should record different majors listed in the university.

1. In this sub tree, is there a task that must come before this one?

There is a select semester task that must come before this task.

1. In this sub tree, is there a task for which this one is required to be immediately preceding? Is there any specific sequence in which the task must be performed?

The task for which this one is required to be immediately preceding is view CPS.

1. Which if any primary entities (things) are involved in this sub-task? (Note: Are these listed as primary entities from the use case analysis?)

The primary entities related to this task are department records and Student records.

1. How can this task fail?

This task fails when the system displays the Students that do not belong to that major or if the system doesn’t display any existing Student belonging to that major.

1. How frequently is this task performed?

The Academic Advisor will perform this task for when the Student first enrolls into the University.

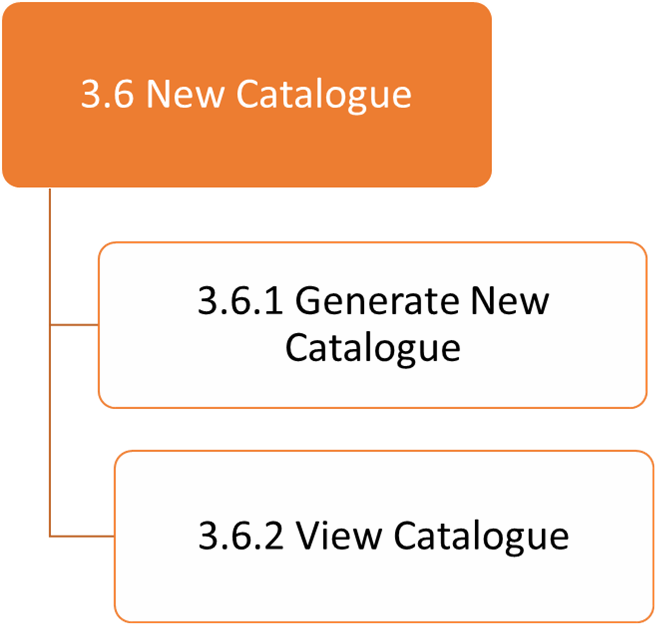
1. How open is this task, especially in terms of its sequence or inputs?

This task is single sequence of input from the Academic Advisor.

1. What if any are the specific usability expectations (e.g. ease of use, ease of learning) for this task and how do we anticipate determining if we have satisfied the expectations?

In selecting the semester, the Academic Advisor need to know what majors are offered in that term.

Task analysis for New Catalogue



3.6.1 Generate New Catalogue

1. What is the goal of this task?

The goal of this task is to generate new course catalogue for each department.

1. What sub-tasks define this task?

The sub- tasks that define this task are Select Course, Select Foundation, Select Elective and Save.

1. Is this task a subset of another task?

Yes. This task is a subset of New Catalogue task.

1. What non-user interface functions does this task require?

This task requires to store the course catalogue information for various departments into the database.

1. What kind of input or actions does this task require from the user?

The Academic Advisor need to select the core courses, electives and foundations for all the departments once every year.

1. What kind of output or results occurs by virtue of performing this task?

By performing this task, the user will be able to generate new course catalogue for each department for particular year.

1. What automatic actions does this task expect from the system?

This task expects the system to display all the courses in all departments and once the Academic Advisor selects core, foundation and electives, they are saved and modified every time he makes changes to the catalogue.

1. What special characteristics of this task should we record?

This task has an important characteristic of displaying a drop-down menu for each course as ‘C’ for selecting core, ‘F’ for foundations and ‘E’ for selecting electives.

1. In this sub tree, is there a task that must come before?

No, there is no sub-task that comes before.

1. In this sub tree, is there a task for which this one is required to be immediately preceding? Is there any specific sequence in which the task must be performed?

For view catalogue, this task is to be immediately preceding.

1. Which if any primary entities (things) are involved in this sub-task? (Note: Are these listed as primary entities from the use case analysis?)

The primary entities listed from use case analysis are course catalogue, departments, Year.

1. How can this task fail?

This task fails when the selected courses are not displayed in the CPS form and if the changes made by the Academic Advisor are not displayed for a required year.

1. How frequently is this task performed?

This task is performed when the Academic Advisor wants to change a course catalogue for particular department once a year.

1. How open is this task, especially in terms of its sequence or inputs?

This task requires multiple inputs from the Academic Advisor like selecting, term, and selecting, major, selecting foundation and saving the changes.

1. What if any are the specific usability expectations (e.g. ease of use, ease of learning) for this task and how do we anticipate determining if we have satisfied the expectations?

This task is user friendly representation. It is very easy to use as all the courses of each department would be visible.

3.6.2 View Catalogue

1. What is the goal of this task?

The goal of this task is to view generated course catalogue for each department.

1. What sub-tasks define this task?

There are no such sub tasks.

1. Is this task a subset of another task?

Yes. This task is a subset of New Catalogue task.

1. What non-user interface functions does this task require?

This task requires to store the generated course catalogue information for various departments into the database in order to view.

1. What kind of input or actions does this task require from the user?

The Academic Advisor can click on View Catalogue to view the generated course catalogue.

1. What kind of output or results occurs by virtue of performing this task?

The Academic Advisor can view the course catalogue for every year.

1. What automatic actions does this task expect from the system?

This task expects the system to display the new catalogue generated by the Academic Advisor.

1. What special characteristics of this task should we record?

This task has an important characteristic of displaying a drop-down menu for each course as ‘C’ for selecting core, ‘F’ for foundations and ‘E’ for selecting electives.

1. In this sub tree, is there a task that must come before?

Generate new catalogue should come before this task.

1. In this sub tree, is there a task for which this one is required to be immediately preceding? Is there any specific sequence in which the task must be performed?

There is no such task.

1. Which if any primary entities (things) are involved in this sub-task? (Note: Are these listed as primary entities from the use case analysis?)

The primary entities listed from use case analysis are course catalogue, departments, Year.

1. How can this task fail?

This task fails when the Academic Advisor is not able to view the generated course catalogue.

1. How frequently is this task performed?

This task is performed whenever the Academic Advisor want to view the course catalogue.

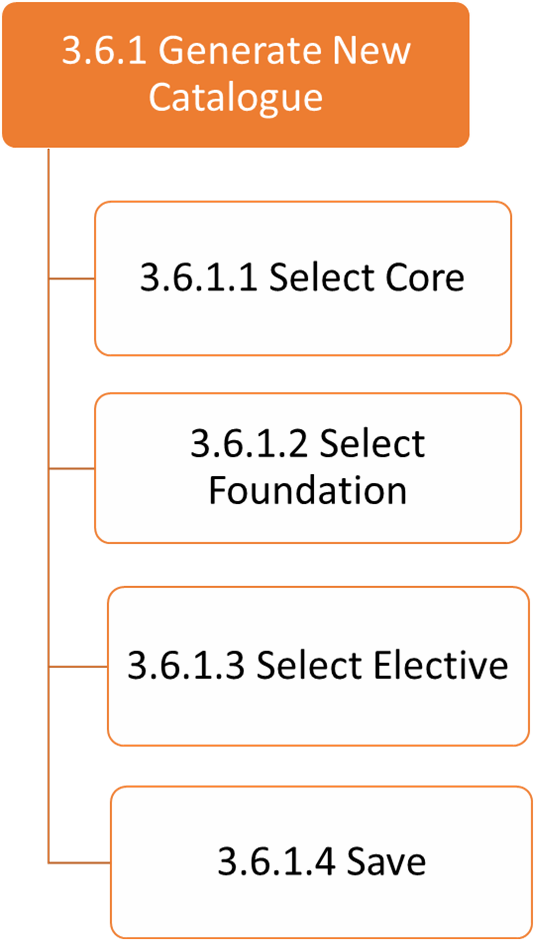
1. How open is this task, especially in terms of its sequence or inputs?

This task is simple sequence of single input by selecting the ‘view catalogue’ button

1. What if any are the specific usability expectations (e.g. ease of use, ease of learning) for this task and how do we anticipate determining if we have satisfied the expectations?

This task is user friendly representation. It is very easy to use as all the courses of each department would be visible.

Task analysis for Generate New Catalogue



3.6.1.1 Select Core

1. What is the goal of this task?

The goal of this task is to generate new course catalogue by selecting core courses from the course catalogue

1. What sub-tasks define this task?

There are no sub tasks required to define this task.

1. Is this task a subset of another task?

Yes. This task is a subset of ‘Generate New Catalogue’ task.

1. What non-user interface functions does this task require?

This task requires the database to store and display courses available in all departments.

1. What kind of input or actions does this task require from the user?

The Academic Advisor need to select the core courses from the list of courses available in the catalogue.

1. What kind of output or results occurs by virtue of performing this task?

By performing this task, the Academic Advisor can select which core courses are offered for all departments in that term.

1. What automatic actions does this task expect from the system?

This task expects the system to display all the courses in all departments and once the Academic Advisor selects core and they are saved and modified every time he makes changes to the catalogue.

1. What special characteristics of this task should we record?

This task has an important characteristic of displaying a drop-down menu for each course as ‘C’ for selecting core, ‘F’ for foundations and ‘E’ for selecting electives.

1. In this sub tree, is there a task that must come before?

No, there is no sub-task that comes before.

1. In this sub tree, is there a task for which this one is required to be immediately preceding? Is there any specific sequence in which the task must be performed?

No, there is no such task.

1. Which if any primary entities (things) are involved in this sub-task? (Note: Are these listed as primary entities from the use case analysis?)

The primary entities listed from use case analysis are course catalogue, core courses.

1. How can this task fail?

This task fails when the selected courses are not displayed in the CPS form and if the changes made by the Academic Advisor are not displayed for a required year.

1. How frequently is this task performed?

This task is performed when the Academic Advisor wants to change a core course from catalogue for particular department once a year.

1. How open is this task, especially in terms of its sequence or inputs?

This task requires single input from the Academic Advisor like selecting the core courses for each department.

1. What if any are the specific usability expectations (e.g. ease of use, ease of learning) for this task and how do we anticipate determining if we have satisfied the expectations?

This task is user friendly representation. It is very easy to use as all the courses of each department would be visible.

3.6.1.2 Select Foundation

1. What is the goal of this task?

The goal of this task is to generate new course catalogue by selecting foundation courses from the course catalogue

1. What sub-tasks define this task?

There are no sub tasks required to define this task.

1. Is this task a subset of another task?

Yes. This task is a subset of ‘Generate New Catalogue’ task.

1. What non-user interface functions does this task require?

This task requires the database to store and display courses available in all departments.

1. What kind of input or actions does this task require from the user?

The Academic Advisor need to select the foundation courses from the list of courses available in the catalogue.

1. What kind of output or results occurs by virtue of performing this task?

By performing this task, the Academic Advisor can select which foundation courses are offered for all departments in that term.

1. What automatic actions does this task expect from the system?

This task expects the system to display all the courses in all departments and once the Academic Advisor selects foundation and they are saved and modified every time he makes changes to the catalogue.

1. What special characteristics of this task should we record?

This task has an important characteristic of displaying a drop-down menu for each course as ‘C’ for selecting core, ‘F’ for foundations and ‘E’ for selecting electives.

1. In this sub tree, is there a task that must come before?

No, there is no sub-task that comes before.

1. In this sub tree, is there a task for which this one is required to be immediately preceding? Is there any specific sequence in which the task must be performed?

No, there is no such task.

1. Which if any primary entities (things) are involved in this sub-task? (Note: Are these listed as primary entities from the use case analysis?)

The primary entities listed from use case analysis are course catalogue, foundation courses.

1. How can this task fail?

This task fails when the selected courses are not displayed in the CPS form and if the changes made by the Academic Advisor are not displayed for a required year.

1. How frequently is this task performed?

This task is performed when the Academic Advisor wants to change a foundation from course catalogue for particular department once a year.

1. How open is this task, especially in terms of its sequence or inputs?

This task requires single input from the Academic Advisor like selecting the foundation courses for each department.

1. What if any are the specific usability expectations (e.g. ease of use, ease of learning) for this task and how do we anticipate determining if we have satisfied the expectations?

This task is user friendly representation. It is very easy to use as all the courses of each department would be visible.

3.6.1.3 Select Elective

1. What is the goal of this task?

The goal of this task is to generate new course catalogue by selecting Elective courses from the course catalogue

1. What sub-tasks define this task?

There are no sub tasks required to define this task.

1. Is this task a subset of another task?

Yes. This task is a subset of ‘Generate New Catalogue’ task.

1. What non-user interface functions does this task require?

This task requires the database to store and display courses available in all departments.

1. What kind of input or actions does this task require from the user?

The Academic Advisor need to select the Elective courses from the list of courses available in the catalogue.

1. What kind of output or results occurs by virtue of performing this task?

By performing this task, the Academic Advisor can select which Elective courses are offered for all departments in that term.

1. What automatic actions does this task expect from the system?

This task expects the system to display all the courses in all departments and once the Academic Advisor selects Electives and they are saved and modified every time he makes changes to the catalogue.

1. What special characteristics of this task should we record?

This task has an important characteristic of displaying a drop-down menu for each course as ‘C’ for selecting core, ‘F’ for foundations and ‘E’ for selecting electives.

1. In this sub tree, is there a task that must come before?

No, there is no sub-task that comes before.

1. In this sub tree, is there a task for which this one is required to be immediately preceding? Is there any specific sequence in which the task must be performed?

No, there is no such task.

1. Which if any primary entities (things) are involved in this sub-task? (Note: Are these listed as primary entities from the use case analysis?)

The primary entities listed from use case analysis are course catalogue, Elective courses.

1. How can this task fail?

This task fails when the selected courses are not displayed in the CPS form and if the changes made by the Academic Advisor are not displayed for a required year.

1. How frequently is this task performed?

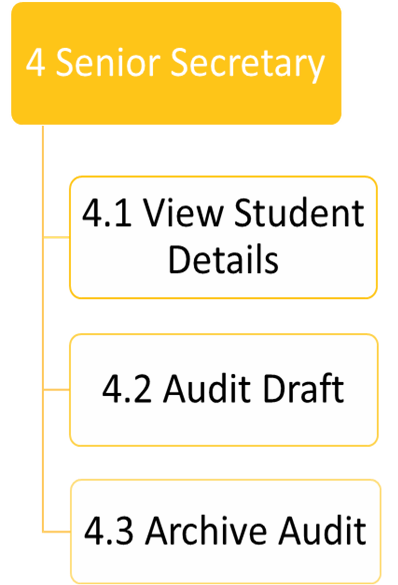
This task is performed when the Academic Advisor wants to change an elective course from course catalogue for particular department once a year.

1. How open is this task, especially in terms of its sequence or inputs?

This task requires single input from the Academic Advisor like selecting the Elective courses for each department.

1. What if any are the specific usability expectations (e.g. ease of use, ease of learning) for this task and how do we anticipate determining if we have satisfied the expectations?

This task is user friendly representation. It is very easy to use as all the courses of each department would be visible.



4.1 View Student Details

1. **What is the goal of this task?**

The goal of this task is to allow the Senior Secretary to view the Student details which are in the University records.

1. **What sub-tasks define this task?**

There are no sub-tasks that define this task.

1. **Is this task a subset of another task?**

This task is a subset of Senior Secretary task.

1. **What non-user interface functions does this task require?**

Senior Secretary will be provided with list of Students by entering Term and Major and their details who are currently enrolled in the program.

1. **What kind of input or actions does this task require from the user?**

The inputs that are required to perform this task from the user are Secretary need to login in the application and can view Student details by entering particular Term and Major.

1. **What kind of output or results occurs by virtue of performing this task?**

The Senior Secretary can successfully view the Student details who are enrolled in particular major and term.

1. **What automatic actions does this task expect from the system?**

The system should display only the Students who are enrolled in particular term and major requested matches with the database.

1. **What special characteristics of this task should we record?**

This task doesn’t require any special characteristics to record.

1. **In this sub tree, is there a task that must come before?**

No, there is no sub-task which comes immediately before.

1. **In this sub tree, is there a task for which this one is required to be immediately preceding? Is there any specific sequence in which the task must be performed?**

No, as this is independent task there is no specific sequence to be performed.

1. **Which if any primary entities (things) are involved in this sub-task? (Note: Are these listed as primary entities from the use case analysis?)**

The primary entities that are involved in this task are Student records and Student details.

1. **How can this task fail?**

This task fails when requested Student data is not available in the database or a Student is not enrolled in the university and if a Student is transferred from another university and the data is not updated frequently.

1. **How frequently is this task performed?**

This task is performed when Senior Secretary wants to view particular Student CPS and their details.

1. **How open is this task, especially in terms of its sequence or inputs?**

Senior Secretary can view a Student by their name or Student ID. This task is controlled in a means that it will be achieved only when the Student got admitted and when Student details are stored in the database.

1. **What if any are the specific usability expectations (e.g. ease of use, ease of learning) for this task and how do we anticipate determining if we have satisfied the expectations?**

The system should be user-friendly, so that Senior Secretary should be able to easily view a Student with the ease of use.

4.2 Audit Draft

1. What is the goal of this task?

The goal of this task is to allow the Senior Secretary to Audit changes that are done to draft CPS.

1. What sub-tasks define this task?

There are no sub-tasks that define this task.

1. Is this task a subset of another task?

Yes, this task is a subset of Senior Secretary task.

1. What non-user interface functions does this task require?

The changes that are done by the Faculty Advisor and Academic Advisor will be stored in database and they are reflected.

1. What kind of input or actions does this task require from the user?

Senior Secretary must be logged into the system and must have Student details like Student name, ID, major and in which year they are admitted in order to perform this task.

1. What kind of output or results occurs by virtue of performing this task?

By successfully performing this task, Senior Secretary will be able to view the audited changes that are made to the draft CPS.

1. What automatic actions does this task expect from the system?

The system should audit all the changes that are performed to Draft CPS of the specific Student whose database matches with the enrolled Students.

1. What special characteristics of this task should we record?

This task requires to record the changes done by the Faculty Advisor and Academic Advisor.

1. In this sub tree, is there a task that must come before this one?

The task that involves before this is generate draft and all the changes that the Faculty Advisor and Academic Advisor perform to CPS.

1. In this sub tree, is there a task for which this one is required to be immediately preceding? Is there any specific sequence in which the task must be performed?

The task it requires to be finished to perform archive audit.

1. Which if any primary entities (things) are involved in this sub-task? (Note: Are these listed as primary entities from the use case analysis?)

The primary entities listed are Student records and their CPS records.

1. How can this task fail?

This task fails when the Academic Advisor or Faculty Advisor doesn’t save changes that are made to draft CPS and if Secretary cannot view the changes made by the Faculty Advisor and Academic Advisor.

1. How frequently is this task performed?

The Senior Secretary will perform this task when the Student requests the final CPS for applying Teaching Assistant or Research Assistant posts and request the final CPS.

1. How open is this task, especially in terms of its sequence or inputs?

This task is simple with sequences of input from Senior Secretary to approve a Student final CPS.

1. What if any are the specific usability expectations (e.g. ease of use, ease of learning) for this task and how do we anticipate determining if we have satisfied the expectations?

This task is user friendly representation. It is very easy to use as all the courses of each department would be visible.

4.3 Archive Audit

1. What is the goal of this task?

The goal of this task is to allow the Senior Secretary to archive graduate Students and alumni Student CPS details.

1. What sub-tasks define this task?

There are no sub-tasks that define this task.

1. Is this task a subset of another task?

Yes, this task is a subset of “Senior Secretary” task.

1. What non-user interface functions does this task require?

This task requires to save the graduated Student CPS details and alumni CPS details to the database.

1. What kind of input or actions does this task require from the user?

Senior Secretary must be logged into the system and must have Student details like Student name, ID, major and in which year they are admitted in order to perform this task.

1. What kind of output or results occurs by virtue of performing this task?

By successfully performing this task, Senior Secretary will be able to separate the CPS details of current Students and graduated Students and also, he can save all the CPS data of all alumni Students.

1. What automatic actions does this task expect from the system?

The system should save all the final CPS details of the graduated Students into separate database from that of the current Student CPS details.

1. What special characteristics of this task should we record?

This task requires to record the audit draft CPS details.

1. In this sub tree, is there a task that must come before this one?

The task that involves before this is generate draft, audit draft.

1. In this sub tree, is there a task for which this one is required to be immediately preceding? Is there any specific sequence in which the task must be performed?

The task it requires is auditing the generated CPS draft.

1. Which if any primary entities (things) are involved in this sub-task? (Note: Are these listed as primary entities from the use case analysis?)

The primary entities associated with this task are Student records and CPS records.

1. How can this task fail?

This task fails when the Senior Secretary cannot audit the changes to the CPS and also the task fails if the CPS details of a Student are not correctly stored in the database.

1. How frequently is this task performed?

The Senior Secretary will perform this task for when the Student graduates from the University.

1. How open is this task, especially in terms of its sequence or inputs?

This task requires simple sequence of inputs from the Secretary by archiving the graduated Student details to the database.

1. What if any are the specific usability expectations (e.g. ease of use, ease of learning) for this task and how do we anticipate determining if we have satisfied the expectations?

In archiving the audit, the Senior Secretary need not have any knowledge as he can easily distinguish the current Student for the graduated Students to save the CPS details.

# 5.Architecture



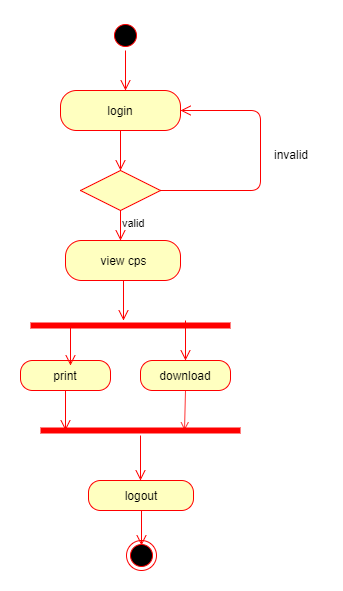
**Figure 11: Architecture Diagram for Online CPS System**

Description:

* We have chosen a 3-tier architecture for developing this CPS portal. The three tiers here are: Application layer, Business Logic layer and Data Access layer. The framework that is used for the developing CPS portal is ASP.NET.
* The First layer here is the Application layer and its main purpose is to translate tasks such a way that user can understand.
* The second layer is the Business layer and the main purpose of this layer is to communicate with the Application layer and validate the inputs coming from them.
* The third layer here is the Data Access Layer and the main purpose of this layer is to store all the related dBs like PeopleSoft dB and Access DB. The Business Access layer and Data Access layer are connected through an API which helps in fetching data from related databases. This layer performs actions like insert, delete and update.
* The user interface uses html and the web forms are created using .aspx
* Likewise, the code in the Business layer also created using the.aspx.cs.
* The Application Programming Interface(API) is used to communicate with the Business logic layer.
* The Online CPS database is nothing but the combination of UHCL Access Database and UHCL people soft database.

# 6. Activity Diagrams

## 6.1 Activity diagram for the Student



**Figure 11: Activity Diagram for Student**

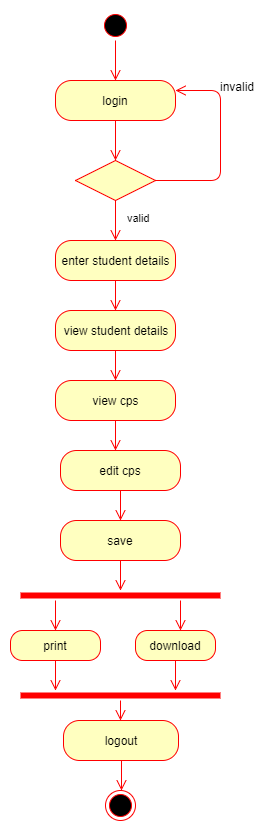
Activity Diagram Description for Student:

|  |  |  |
| --- | --- | --- |
| **S.NO** | **Activity** | **Description** |
|  | Login | Students will login into the application by entering valid username and password. |
|  | View CPS | If the students enter valid credentials, they can view their draft/final CPS. |
|  | Print | After viewing their CPS, if the student wants to print their CPS, they can print it. |
|  | Download | After viewing their CPS, if the student wants to download their CPS, they can download it. |
|  | logout | They can logout of the application after viewing or downloading or printing their CPS. |

**Table 1: Activity Diagram Description for Student**

## 

## 6.2 Activity diagram for Faculty Advisor



**Figure 12: Activity Diagram for Faculty Advisor**

Activity Diagram Description for Faculty Advisor:

|  |  |  |
| --- | --- | --- |
| **S.NO** | **Activity** | **Description** |
|  | Login | Faculty will login into the application by entering valid username and password. |
|  | Enter student name or Id | Faculty will enter the student name or the student Id to view their details. |
|  | View student details | Faculty advisor can view that particular student details. |
|  | View CPS | Faculty can view the student CPS if required by clicking on the hyperlink “VIEW CPS”. |
|  | Edit CPS | Faculty can edit the student CPS after viewing it if needed. |
|  | Save | The changes made to CPS have to be saved. |
|  | Print | Faculty advisor can print the student CPS if required. |
|  | Download | Faculty advisor can download the student CPS if required. |
|  | Logout | Faculty may logout of the application after viewing or editing or downloading or printing the CPS. |

Table 2: Activity Diagram Description for Faculty Advisor

## Activity diagram for Academic Advisor

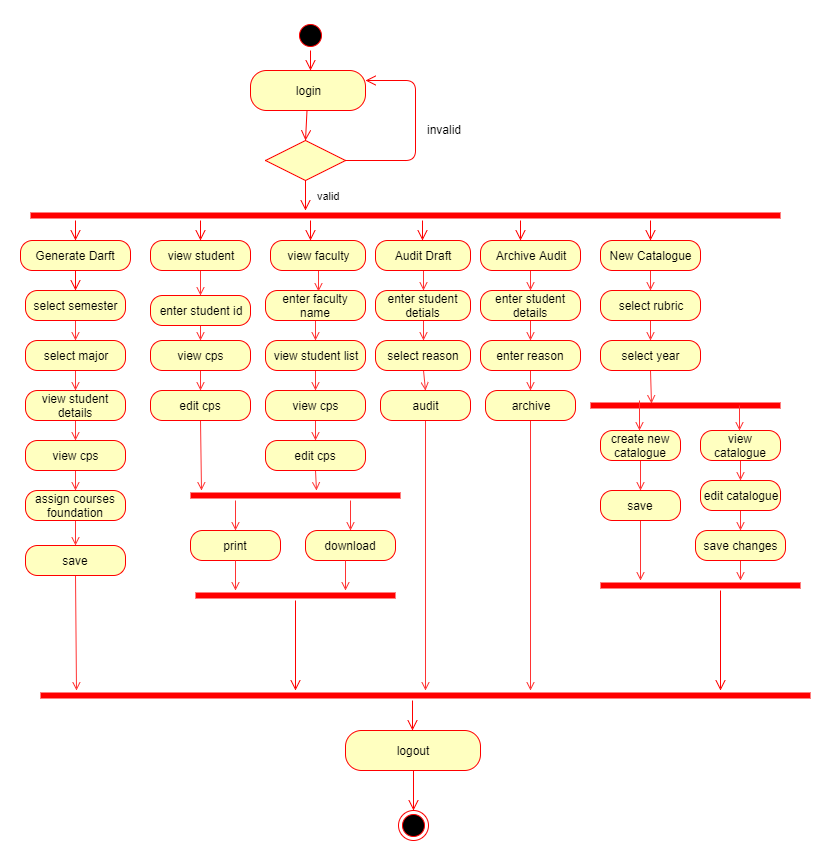


Figure 13: Activity Diagram for Academic Advisor

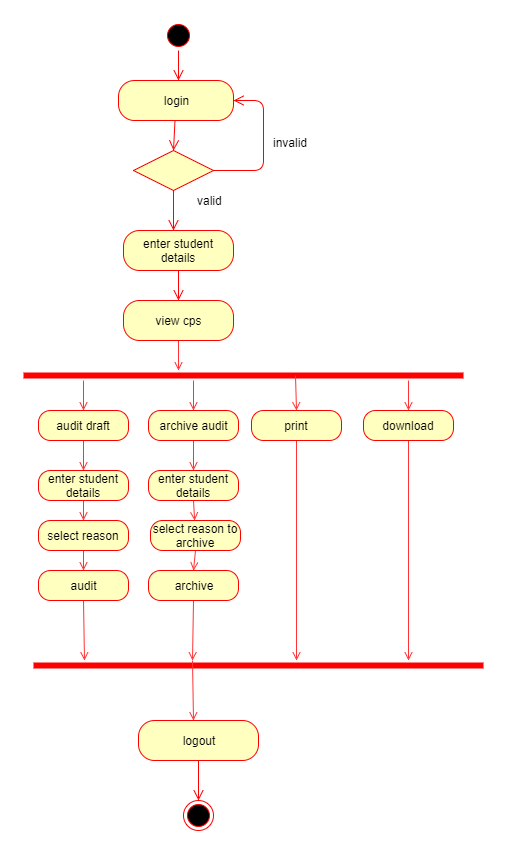
Activity Diagram Description for Academic Advisor:

Academic Advisors will login into the application and they can generate CPS, they can view the student details, they can view faculty, create a new catalog, Audit CPS and archive audit CPS.

|  |  |  |
| --- | --- | --- |
| **S.NO** | **Activity** | **Description** |
|  | Login | Academic Advisor will login into the application by using valid username and password. |
|  | Generate Draft | Academic advisor will generate a draft CPS:   * Select the semester, major of the students to view their CPS. * View the student CPS. * Click on the hyperlink “VIEW CPS”. * Fills the foundation courses in CPS form. * Save the draft CPS. |
|  | View student | Academic advisor can view the student:   * Enter student details. * View the CPS and edit it if required. * Save the CPS modified. * Print the CPS or download it if needed. |
|  | View Faculty | Academic advisor can view the faculty:   * Enter faculty details. * View the student list assigned to that faculty. * View CPS of a student by clicking on the hyperlink “VIEW CPS” and edit the CPS if required. |
|  | Audit draft | Academic advisor can audit CPS:   * Enter student details. * Select reason to audit from the dropdown like Teaching assistant, Research assistant and capstone enrolment. * Audit. |
|  | Archive audit | Academic advisor can archive CPS:   * Enter student details. * Enter reason to archive (For example, Transferred, Graduated and so on). * Archive CPS.   Academic advisor can also view the archived CPS:   * View student list whose CPS are archived. * View the CPS if required. |
|  | New catalogue | * Select the rubric. * Select the academic year.   Academic advisor can create a new catalogue:   * Fills the new catalogue. * Save it.   Academic advisor can create a new catalogue by editing the previous year catalogue:   * View the previous year catalogue. * Edit the course catalogue. * Save the edited course catalogue to the next academic year. |
|  | Logout | Academic advisor may logout of the application after performing their tasks. |

**Table 3: Activity Diagram Description for Academic Advisor**

## Activity diagram for a Senior Secretary



**Figure 14: Activity Diagram for Senior Secretary**

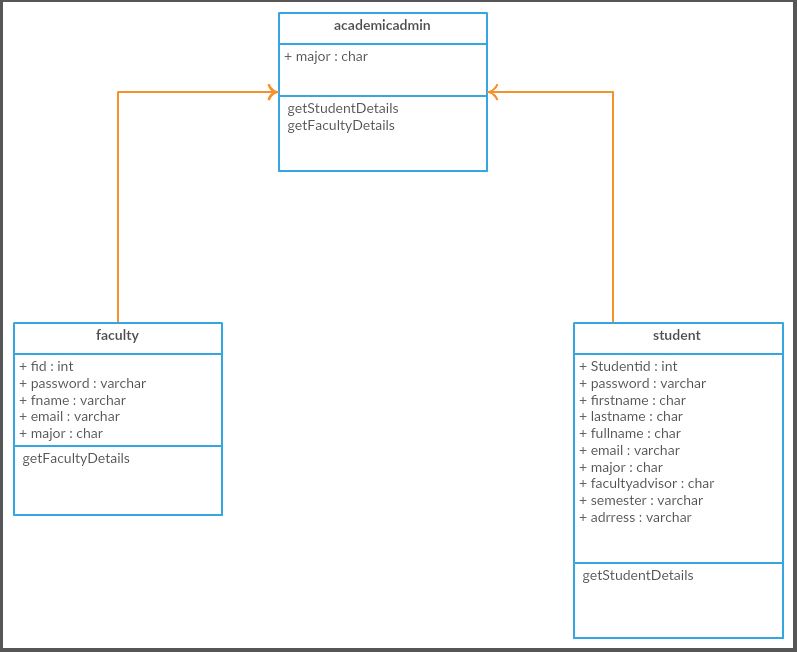
Activity Diagram Description for Senior Secretary:

|  |  |  |
| --- | --- | --- |
| **S.NO** | **Activity** | **Description** |
|  | Login | Senior Secretary will login to the application by entering valid username and password. |
|  | Enter student details | Search student by entering student name or student Id. |
|  | View CPS | View the student CPS if required by clicking on the hyperlink “VIEW CPS”. |
|  | Audit draft | Senior Secretary can audit CPS.   * Enter student details. * Select reason to audit from the dropdown like Teaching assistant, Research assistant and capstone enrolment. * Audit. |
|  | Archive audit | Senior secretary can archive CPS:   * Enter student details. * Enter reason to archive (For example, Transferred, Graduated and so on). * Archive CPS.   Senior secretary can also view the archived CPS:   * View student list whose CPS are archived. * View the CPS if required. |
|  | Print | After viewing the CPS, if the senior secretary wants to print the CPS, they can print it. |
|  | Download | After viewing the CPS, if the senior secretary wants to download the CPS, they can download it. |
|  | logout | The senior secretary may logout of the application after performing their tasks. |

**Table 4: Activity Diagram Description for Senior Secretary**

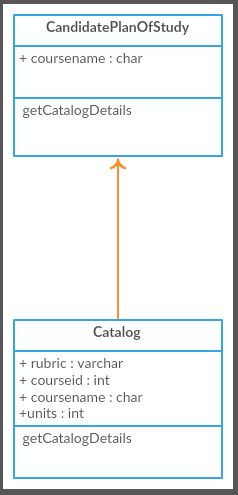
# 7.Class Diagrams

## 7.1 Class Diagram for academicadmin Database



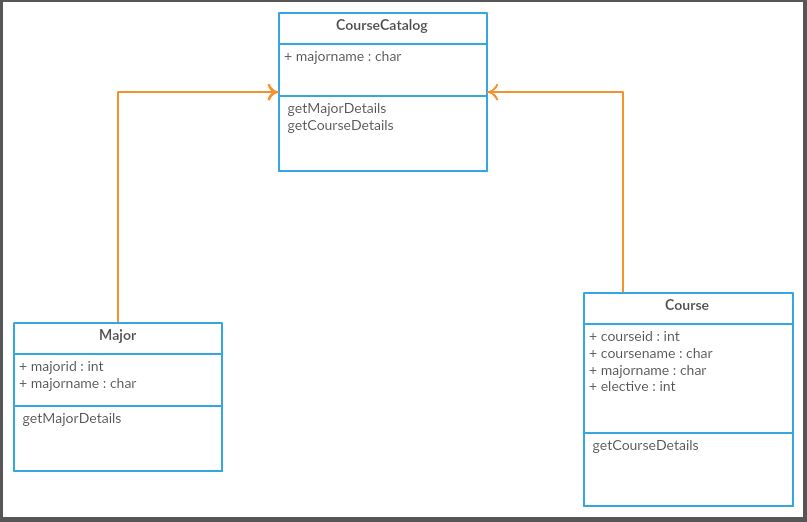
**Figure 15: Class Diagram for academicadmin Database**

## 7.2 Class Diagram for CandidatePlanOfStudy Database



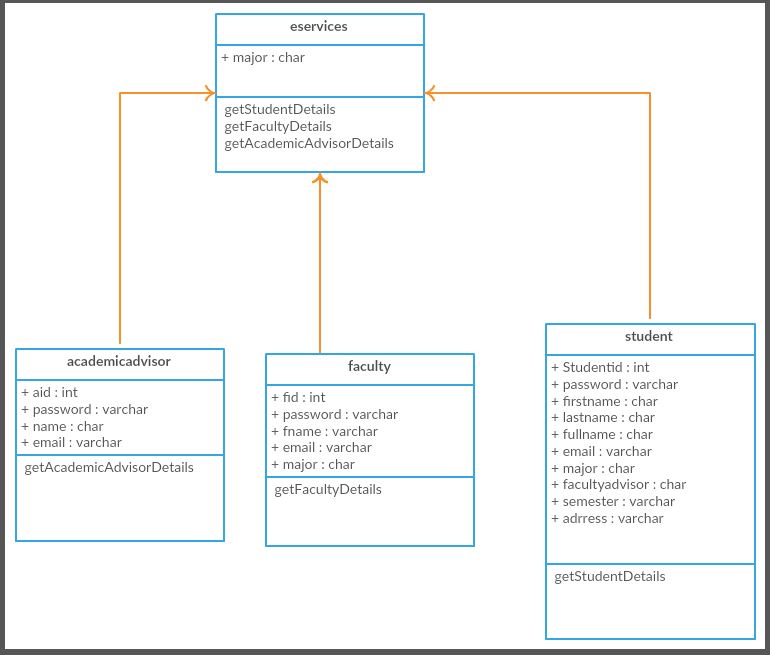
**Figure 16: Class Diagram for CandidatePlanOfStudy Database**

## 7.3 Class Diagram for CourseCatalog Database



**Figure 17: Class Diagram for CourseCatalog Database**

## Class Diagram for eservices Database



**Figure 18: Class Diagram for eservices Database**

# 8.Project Guidelines

Introduction:

The Project guidelines helps us to understand the visual elements that is being used in the application. The several types of elements used in the User interface of the application are textbox, buttons, panels, icons, layouts and so on. This can act as a reference document for the users who will be using it in the future.

**Format Guidelines:**

In this section, various elements of the website are described with a description and image.

1. **Textbox:** The textboxes are nothing but the text areas where users can enter

username, password, search for a student profile.

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Image** |
| Default textbox | It is used to enter user id, to search for a user profile. |  |
| Password textbox | It is used to enter user password and the use of this textbox to hide the text. |  |

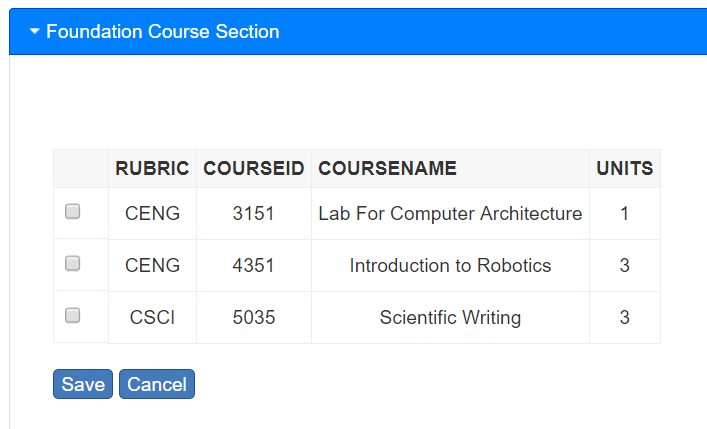
1. **Buttons:** The buttons are used to trigger an event or an action. In our application, we have used buttons when a user wants to trigger the login function after entering all the details or when he wants to search for some details or when he needs to save the progress.

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Image** |
| Default button | The default button is the basic type of button that is used when the user tries to log into the system and when he searches using a student id or  when he tries to archive a CPS. |  |
| Link button | The link button helps the user to be redirected to the CPS form when he wants to add a foundation course or make any changes. |  |

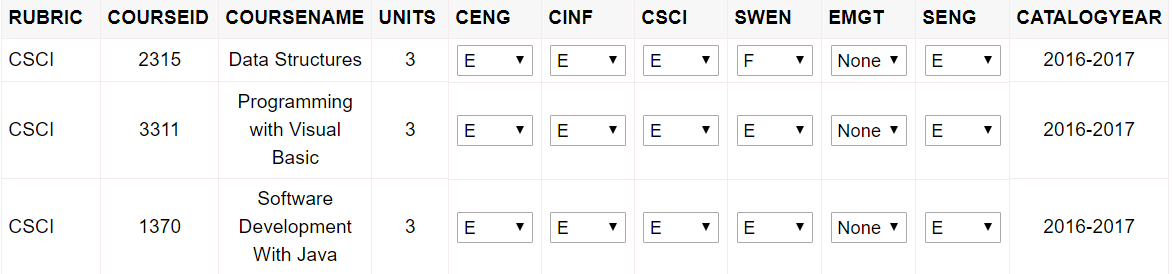
1. **Checkbox:** The main functionality of checkbox is when a user is listed with multiple options, he/she can able to select more than one option.

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Image** |
| Default checkbox | This checkbox is used to select multiple foundation courses for a student. |  |

1. **Panels:** Panel is used to indicate dropdown menu and to wrap the components. A panel in bootstrap is a bordered box with some padding around its content.

****

1. **Table:** We have displayed the new catalogue information in the table format with data in the rows and columns. This makes the formatting simpler and easy understandable.



1. **Dropdown:** Thedropdown list allows user to select only one value from the pre-defined list.

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Image** |
| Dropdown Menu | The user can select the type of Rubric from the select Rubric dropdown and select the year as well from the select year dropdown. |  |

1. **Alerts:** The alerts are nothing but the warnings or important messages that are conveyed to the user. We have used the below alert in our login screen in the application.

**Login failure alert:** The below alert or message will be displayed when the user enters wrong credentials.

****

1. **Fonts:** We have mentioned the fonts and their descriptions that has been used in the application in the table below.

|  |  |
| --- | --- |
| **Type** | **Font description** |
| Login Headings | Size: 22px  font-family: Verdana, Geneva, sans-serif  Weight: 400 |
| Panel Heading | Size: 22px  Color: Blue  font-family: Verdana, Geneva, sans-serif |
| Menu text | Size: 4px  Color: RGB (255,255,255) |
| User inputs | Size: 10px  Color: Black  font-family: Verdana, Geneva, sans-serif |

# 9.Design Rationale

The objective of the design rationale document is to provide some insights on the overall experience of the Candidate plan of study website design. The goal of the project is to automate the manual process of CPS. The project revolves around four actors: student, academic advisor, faculty advisor, senior secretary. The project can be considered as a successful one when these four actors are able to do their tasks to perfection. Each actor does different set of activities. For instance, senior secretary can have access to all student information including their password information and they can audit and archive the CPS of students which can be used in the future. We have collected different set of requirements for each actor to make the application more pleasing to use.

The collected requirements are transformed into paper prototypes which helped us to get into an agreement with the user. We have used different design frameworks to build this online CPS system. Each actor in the system has been provided with different frameworks so that they can view only the or designated information. We have provided unique panel frameworks for foundation courses, core courses, electives, thesis. The major design constraint we faced is due to short time and limited number of resources, we have chosen web form based frameworks for the project.

# 11.QUIS Report

Introduction:

The purpose of this document is analyzing the QUIS (Questionnaire for User Interface Satisfaction) for Online CPS Portal website. QUIS is a measure of overall system satisfaction. A usability test was conducted to assess the system functionality, system performance, user interface, system usability.

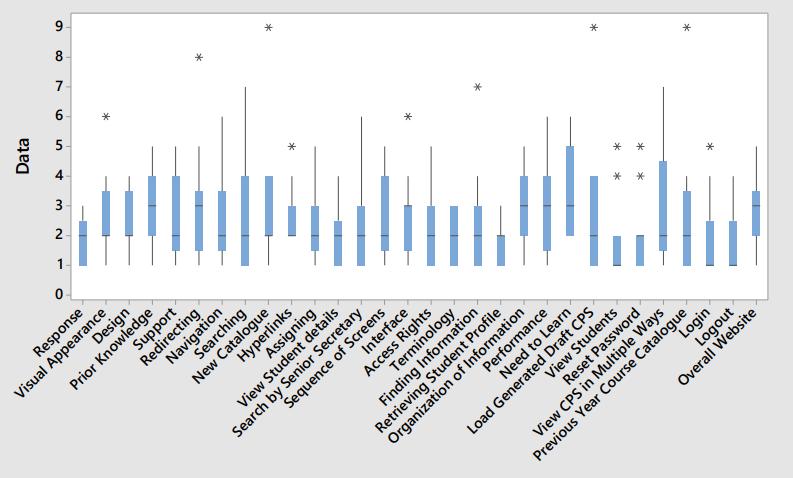
For this usability test, we have invited 13 users to test our website. They were given 4 scenarios which includes student login, faculty advisor login, academic advisor tasks, and senior secretary tasks. Users tested the website functionality, user interface, system usability, system performance. All the user responses, user feedback was taken and generated a box plot using “Minitab”.

**QUESTIONNAIRE FOR ONLINE CPS PORTAL:**

|  |  |
| --- | --- |
| Q1 | All the buttons on the website are responding properly and meaningfully. |
| Q2 | The website has good visual appearance. |
| Q2 | The website design matches its purpose. |
| Q4 | First time user doesn’t need any prior knowledge to use the website. |
| Q5 | Website support perspective of different users. |
| Q6 | It is easy to get to home page from the page you are using. |
| Q7 | The navigation from one page to another page is taking minimum time. |
| Q8 | Website search option meets user expectations. |
| Q9 | Creating a new catalogue for a specific year. |
| Q10 | Number of hyperlinks and buttons in Academic Advisor page are reasonable. |
| Q11 | Assigning foundation courses to a particular student is easy. |
| Q12 | The Senior Secretary is able to view all the student’s details. |
| Q13 | The Senior Secretary can also search a particular student by using student ID. |
| Q14 | The website has clear sequence of screens. |
| Q15 | The interface of this website is easy to read (both font and size). |
| Q16 | The website had the perfect access rights provided to users. |
| Q17 | Website provides terminology related to task. |
| Q18 | Users can able to find all the information they are looking for on this website. |
| Q19 | Retrieving Student profile in Academic Advisor page. |
| Q20 | Organization of information on screen. |
| Q21 | Tasks can be performed in a straight-forward manner. |
| Q22 | Need to learn a lot of things before getting into the website. |
| Q23 | Faculty Advisor is able to load the generated draft CPS. |
| Q24 | Academic advisor can view the students assigned to the particular Faculty Advisor. |
| Q25 | It is easy to reset the password. |
| Q26 | Academic Advisor can view student CPS in multiple ways. |
| Q27 | Academic Advisor can view the previous year course catalogue. |
| Q28 | Authorized Students can login successfully. |
| Q29 | User was able to logout from the website whenever he/she liked to. |
| Q30 | Overall website is satisfactory. |

**Table 5: Questionnaire for Online CPS System**

Generated Box plot for above Questionnaire:



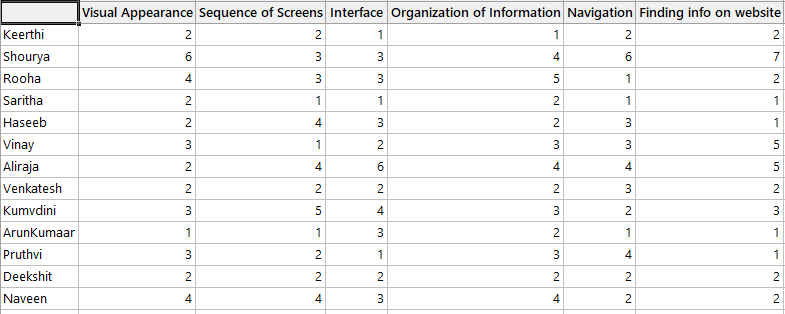
**Figure 19: Generated Box plot for Questionnaire**

The above figure shows the user responses for the website. Based on these results, we changed the user interface, few system functionalities, and made the system performance reliable.

This usability test is performed by 13 users and the X-axis above represents the questions asked to the users, Y-axis represents the 0-9 scale where 0 is strongly agreed and 9 is strongly disagreed.

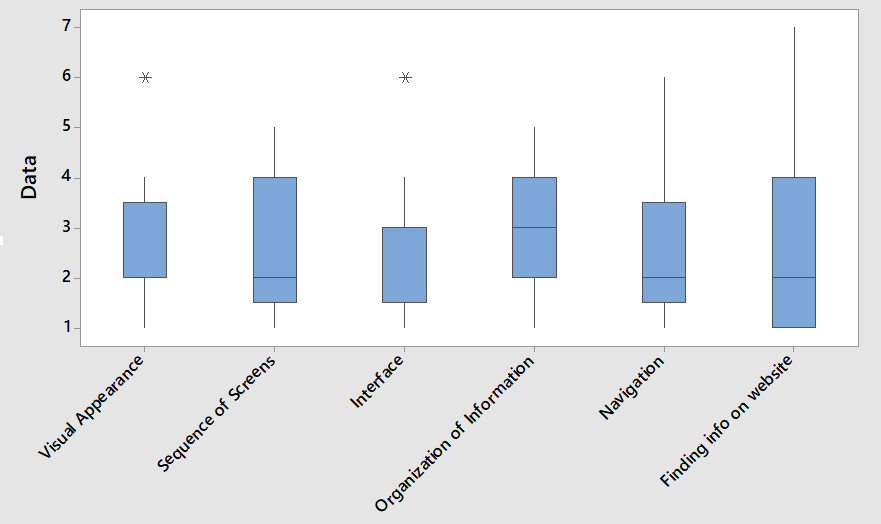
## 11.1 QUIS for User Interface

The goal of user interface design is to produce the user interface which is simple, easy to access, user friendly, efficient. This means that the user should give minimal input to achieve the desired tasks. Any website developed should have a good user interface which is clear, concise, responsive, consistent, familiar, attractive, and efficient.



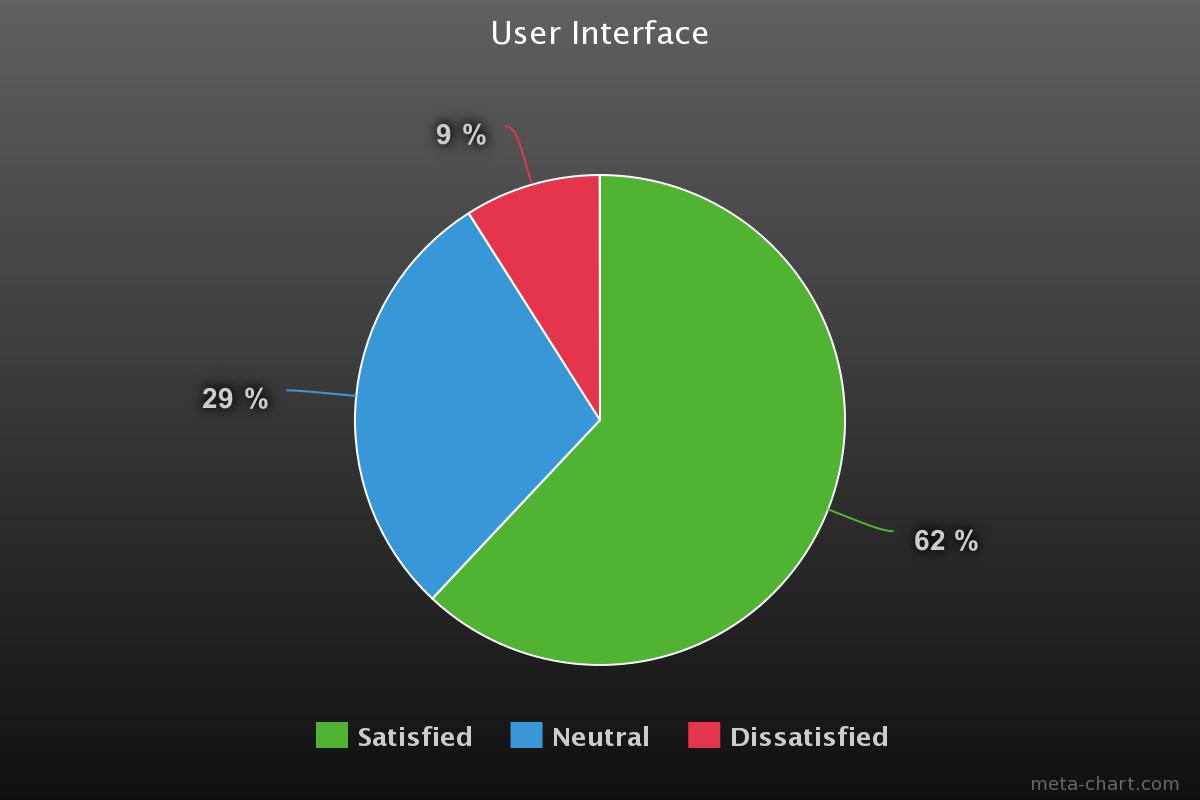
**Table 6: Raw data for User Interface**

This is the raw data obtained for the user interface design from the users. From this raw data, we generated a box plot in “Minitab”.



**Figure 20: Box plot for User Interface**

The response given to user interface includes visual appearance, sequence of screens, interface, organization of information, navigation, finding information on website.



**Figure 21: User response for user interface**

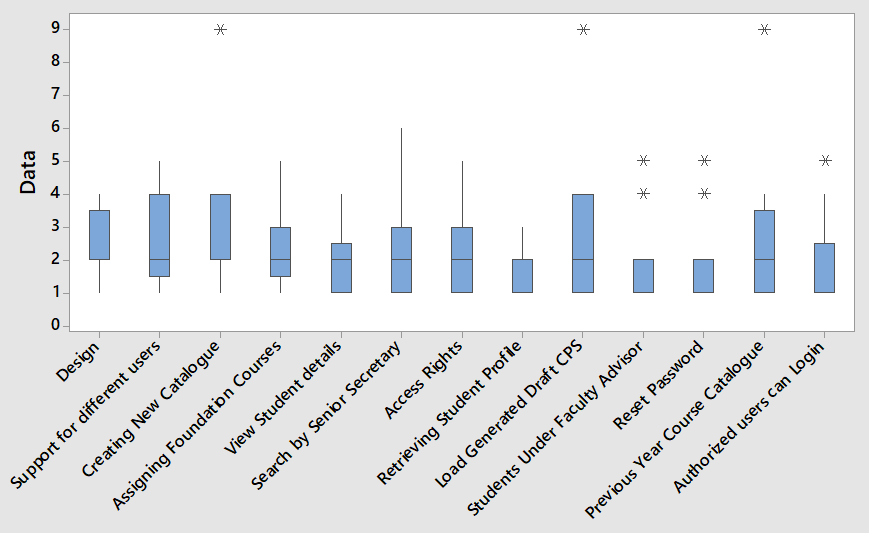
The above pie chart shows the user interface response. It shows 62% of users are satisfied, 29% of the users are neutral and 9% of the users are dissatisfied with our website user interface.

After studying all these results, we worked on the feedback given and tried to achieve 100% user interface. Following are the changes made after the QUIS analysis on user interface.

|  |  |
| --- | --- |
| User Interface | After QUIS |
| Visual appearance | Changed the colour of login button to differentiate it from the background colour.  Changed the CPS format with proper alignment. |
| Sequence of screens | Sequence of screens is proper. No changes made. |
| Interface | Changed the font size to show the differentiation.  Example:  Student name: Niharika  Major: SWEN  Font size of major is different from SWEN. |
| Organization of information | CPS format is organized into 4 panels based on user feedback. |
| Navigation | Navigation to back page is made possible now. |

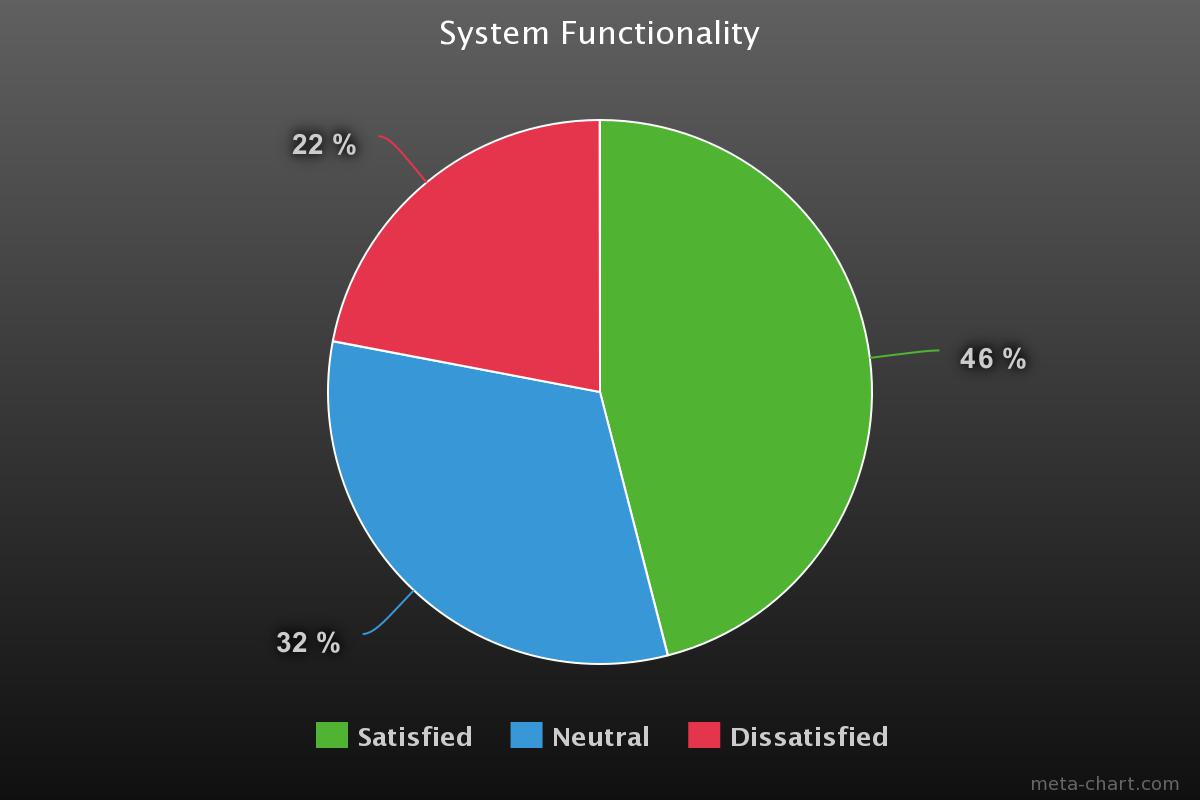
## 11.2 QUIS for System Functionality

System functionality is the interactive part of the website. The purpose of QUIS on system functionality is to make sure whether the users are satisfied with all the functionalities and whether all the functionalities are meeting their purpose.



**Figure 22: Box plot for System Functionality**

The response given by users for system performance includes support for different users, creating new catalogue, assigning foundation courses, view student details, access rights, retrieving student profile, loading generated draft CPS, resetting the password, login for authorized users and so on.



**Figure 23: User response to system functionality**

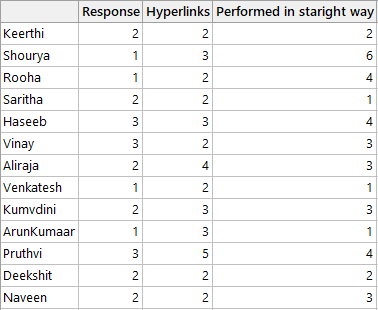
The above pie chart shows the user response to the system functionality. 46% of the users are satisfied with the system functionality, 32% of the users are neutral and 22% of the users are dissatisfied.

From the responses and the feedback taken from the users, we analyzed and worked on system functionality to satisfy all the users.

|  |  |
| --- | --- |
| System Functionality | After QUIS |
| Design matches its purpose | Overall website design matched its purpose well. No changes made. |
| Support for different users | Senior secretary has login. No functionality is done before.  After QUIS, we added the functionality to senior secretary to view the students, audit draft, archive audit. |
| Creating new catalogue | This functionality didn’t work well before QUIS. We worked on creating a new catalogue by filling all the courses and saving them to the database properly without getting any errors. |
| Assigning foundation courses | Academic advisor is able to assign foundation courses to the students. No changes made here. |
| View student details | Faculty advisor, academic advisor can view the student details.  After QUIS, we made senior secretary also to view the student details. |
| Search by senior secretary | Senior secretary can search for a student. No changes made. |
| Access rights | The website had perfect access rights provided to the users. No changes made. |
| Retrieving student profile in Academic Advisor page | Student profile can be retrieved well in the Academic Advisor page. Hence, there are no changes to be made. |
| Load generated draft CPS | After QUIS, we made the faculty advisor to load the draft CPS generated. Previously, this functionality didn’t work well. Now, we made it to work accurately so that the courses that are generated by Academic Advisor in draft CPS can be loaded and viewed by faculty advisor. |
| Students under faculty advisor | Academic advisor can view the students assigned under faculty advisor. No changes made. |
| Reset password | Resetting the password worked well by getting an email. |
| Previous year course catalogue | Academic advisor can view the previous year catalogue, but cannot create a new catalogue from previous year.  After QUIS we made the Academic Advisor to edit the previous year catalogue. |
| Authorized student’s login | Only authorized students can login. This functionality worked well. No changes made. |

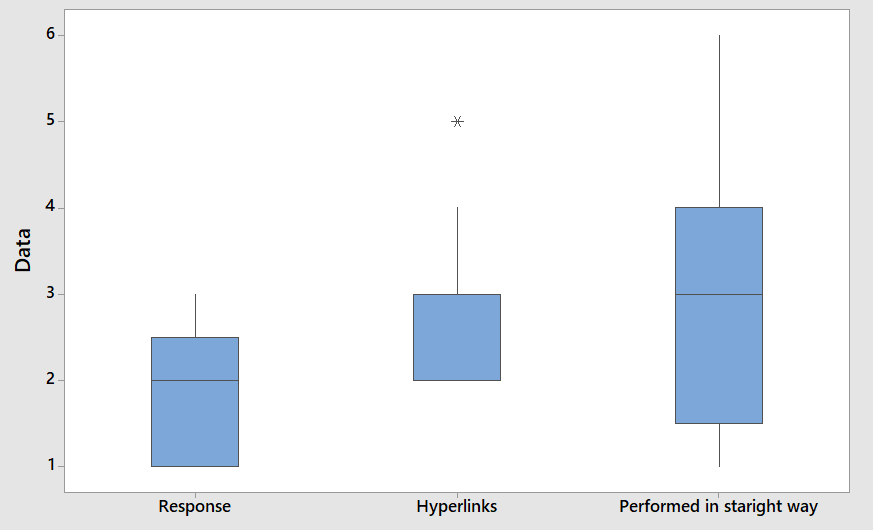
## 11.3 QUIS for System Performance

Performance determines total effectiveness of a website including response time, availability, webpage load time, speed.



**Table 7: Raw data for System Performance**

This is the raw data obtained from the users for system performance. We generated a box plot from the above raw data in “Minitab”.



**Figure 24: Generated Box plot for System Performance**

The response given by users includes website response, performance of tasks and so on.



**Figure 25: User Response to System Performance**

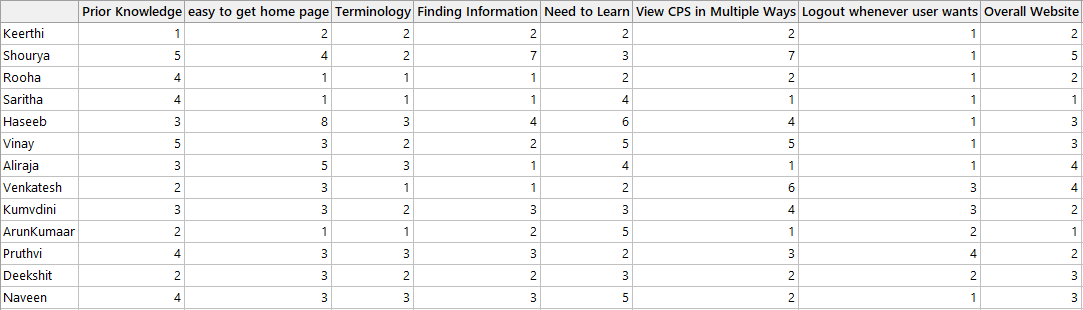
The above pie chart shows the user response to the website performance. 80% of the users are satisfied with the performance of our website, 17% of the users are neutral, 3% of the users are dissatisfied with our website.

From the responses and feedback taken from the users, we analyzed the performance of the website and tried to improve that.

|  |  |
| --- | --- |
| System Performance | After QUIS |
| Website buttons response | The response of website buttons is speed. So, no changes are made. |
| Reasonable number of hyperlinks and buttons | Removed few buttons which are not that useful. |
| Tasks performance | The performance of the tasks is in a straight forward manner. No changes are made as the performance is good. |

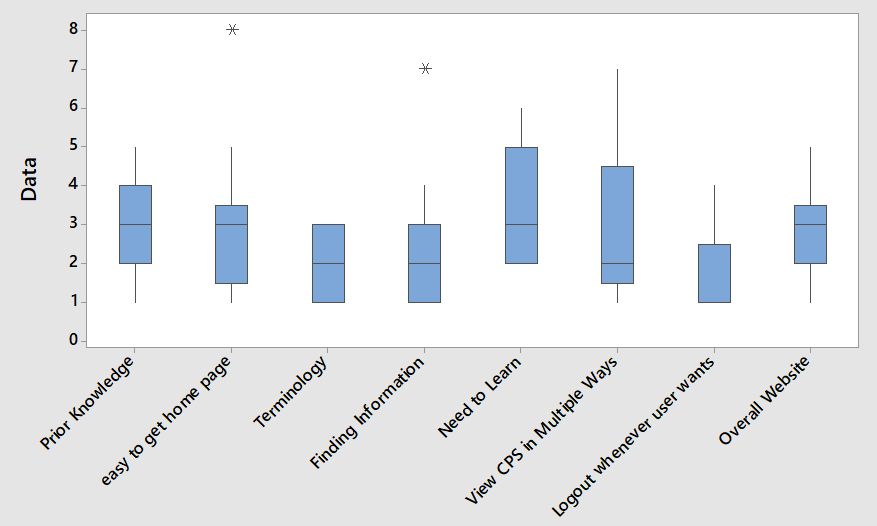
## 11.4 QUIS for System Usability

It is the ease of use of a website. Usability of a website includes making the website simple, consistent, relevant, accessible. The main goal is users should not find any inconvenience while browsing the website, while using it.



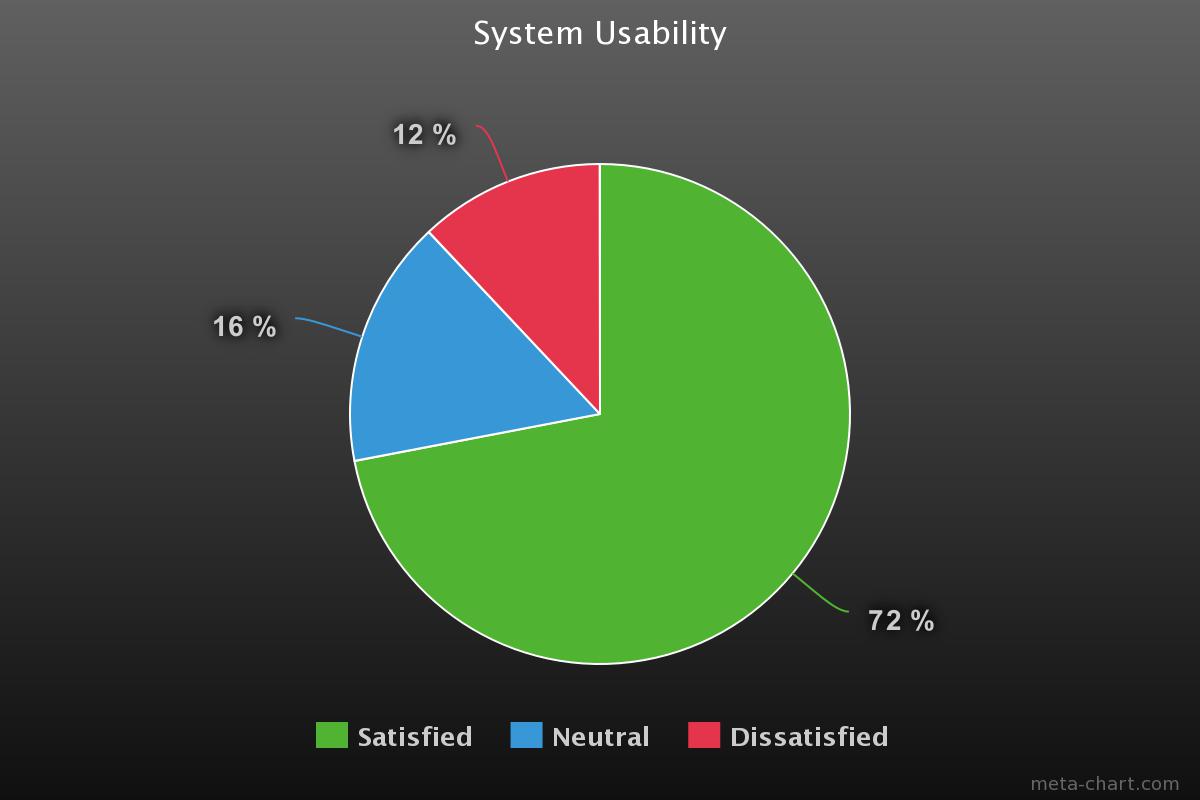
**Table 8: Raw data for System Usability**

This is the raw data obtained for the usability from the users. From this raw data, we generated a box plot in “Minitab”.



**Figure 26: Generated Box plot for System Usability**

The responses obtained from the users includes easy to navigate to home page, finding the required information in the website, Viewing CPS in multiple ways, satisfaction of the overall website, logging out whenever user wants to, terminology describing the tasks.



**Figure 27: User Response to System Usability**

The above pie chart shows the user response to our website usability. 72% of the users are satisfied with the usability of our website, 16% of the users are neutral and 12% of the users are dissatisfied.

From the feedback obtained from the users, we worked on increasing the usability of the website.

|  |  |
| --- | --- |
| Usability | After QUIS |
| Users having prior knowledge | Users need not have prior knowledge to use our website. No changes were made. |
| Easy to get home page | From the CPS page, it is not possible to go the home page.  After QUIS, we fixed that problem and now we can easily go to home page from any page. |
| Terminology related to task | Website provides the terminology related to the task well so that the user can understand what task is to do what. So, no changes were made. |
| Finding information | Users could not find the information they are looking for on this website sometimes. Because, few functionalities were not included before OUIS.  Now, after adding audit draft, archive audit, viewing previous year catalogue, users can find all the information they are searching for. |
| Need to learn | Users need not learn lot of things before getting into the website. No changes were made here. |
| View CPS in multiple ways | Academic advisor can view CPS in multiple ways. We worked on this to make it efficient. |
| Logout whenever user needs | No changes made here because user can logout whenever needed. |
| Overall website | The overall website is made satisfactory to the users. |

# 12. Feedback obtained from the users:

While doing a usability test, we got the feedback from the users. We noted the changes to be made and worked on these changes to make the website satisfactory.

**For Academic Advisor:**

**Generate Draft:** In the generate draft functionality, following changes are to be made.

1. After clicking on the generate draft button, differentiate generate draft and filter names with font size and font style.
2. In the drop-downs, it should be displayed as “select” by default.
3. First letter in the column names of the table should be in capitals. (Example: First Name, Faculty Advisor)
4. By default, it should not display any collapse format section. It should open when we click particular section. Fix the problem and make it efficient.
5. In the foundation courses section:

* Reduce the column space for the check box button.
* Whenever we change the foundation courses, they should override the previously assigned courses.
* Reduce the column space between rubric and the course Id.

**New Catalogue:**

1. After selecting an academic year, it should display the saved data for that academic year and we should be able to edit that and save it to next academic year.
2. Change save button to “Save To” button and keep another drop-down for the academic year below.

**View students:**

1. Remove select semester and major filters in “view student”. There should be only search option where the student details can be retrieved using the student Id or name. (Auto complete for this search)

**For Faculty Advisor:**

1. Electives should have drop-downs for major, courses, grades to select and save. If we select one major from that drop-down, then it should display all courses under that major automatically, so that we can select that one course which we need.

