comp2606: Software Engineering 1 Project

**Group Name: R.O.K.S. Group Members:**

Raphael Superville-816010927

Joshua Oxley-816009424

Kenaird Saunders -816008473

Adrian Santo-816008774

**Project Name:** Student Online Advisory System

Contents

[List of Diagrams 2](#_Toc6220009)

[Glossary of Terms 2](#_Toc6220010)

[Introduction 2](#_Toc6220011)

[Problem 2](#_Toc6220012)

[Goals/ Aims of System 3](#_Toc6220013)

[System Requirements 3](#_Toc6220014)

[User Requirements 4](#_Toc6220015)

[System Requirements 4](#_Toc6220016)

[Functional requirements 4](#_Toc6220017)

[Non-functional requirements 5](#_Toc6220018)

[User Stories 6](#_Toc6220019)

[Scenarios/ User Stories 6](#_Toc6220020)

[Use Cases 11](#_Toc6220021)

[Simple Use Cases from User Stories 11](#_Toc6220022)

[Use Case Diagram 11](#_Toc6220023)

[Ranking of Use Cases 14](#_Toc6220024)

[Expanded Use Cases 14](#_Toc6220025)

[Sequence Diagrams 18](#_Toc6220026)

[Top use Case 19](#_Toc6220027)

[Class Diagrams 19](#_Toc6220028)

[Testing Plans 20](#_Toc6220029)

[Method Cases: 20](#_Toc6220030)

[Risk Management 21](#_Toc6220031)

[Cost Estimation 23](#_Toc6220032)

[Interfaces 23](#_Toc6220033)

[References 23](#_Toc6220034)

# List of Diagrams

Figure1.1-Page

Figure1.2-Page

Figure1.3-Page

Figure 1.4-Page

Figue2-Page

Figure3-Page

Figure4.1-Page

Figure4.2-Page

Figure4.3-Page

Table1-Page

Table2.1-Page

Figure2.2-Page

Table2.3-Page

# Introduction

Our proposed project entitled, Student Online Advisory System, seeks to provide easier access to an important service provided by the university which is the consultation and advice with respect to the selection of courses. To achieve this, an automated academic advising feature will include a Requirement to Withdrawal (RTW) query/check, recommendations on course selections for the coming semester and warning/prompts on a student’s degree progress. Furthermore, a student profiling feature will generate student’s information and their degree information and statistics. In addition, our project will utilise forums and digital bulletin boards for users (students, teachers, tutors, head of departments and deans) to enable discussion and announcements of topics of similar concern or interest and create a virtual help-desk where students can communicate any desired query with relevant personnel. Degree audit reports, in collaboration with academic record units, will feed the necessary information to the various features provided by the system. In addition, for consistency of user accounts, each account registered to the system will be associated to the user’s university-wide account credentials.

## Problem

* Uncertainty with the most suitable selection of courses per semester.
* Inconsistent information communicated through various sources without an avenue for review or vetting
* Eliminates the need for mandatory face-to-face and physical interview or meeting.
* To meet the demands or requests made by internal or external stakeholders for a comprehensive profile of a student.
* Need for a virtual assistant or workspace which monitors a student’s progress towards graduation

## Goals/ Aims of System

* Improved quality of services provided by the university (timely advice (e.g. course selection)) that can be done without the need for face-to-face interaction.
* Allows for transparency and consistency of information communicated to various users.
* Timely feedback
* Increased accessibility and convenience for users
* More efficient administrative procedures (e.g. registration)

## System Requirements

* The system shall be a menu driver interface with dialog boxes to help all users understand what each function does and for easy navigation of the system.
* The system’s authentication scheme will utilise users’ university-wide identification credentials as usernames. Upon registration, the users will be prompted for their passwords which would then be hashed using the Secure Hashing Algorithm (SHA) and stored the system’s internal database.
* Connection with back-end databases for the retrieval of data to display to users in addition to its storage.
  + External back-end databases: academic record units and degree audit records.
  + Internal back-end databases: for the storage of course/department reports and password hashes.
* Use of screen prompts, flags, notifications and confirmation messages will facilitate the alerts needed by the system. Upon the selection of appropriate options or buttons, the user will be redirected to the required page or interface.
* To facilitate the recommendations on course selections for the upcoming semester, all the scheduled courses (core, elective and compulsory), aligned with students’ regulations and syllabuses handbook, and their credit amounts will be sourced from the academic record units and display to the student.
* For each user, all authorized forums are displayed with their title, description, list of threads and number of unread posts.
  + Threads are display in reverse chronological order. Important threads and bulletin boards are fixed at the top of the forum. If authorised, a “Add a new thread” button is displayed to the user. Each thread provides a dropdown menu to visit other threads in the forum.
  + Users can reply, edit and if permitted, delete their existing post.
  + Each post contains:
    - Picture(optional) and name of the author.
    - Date of post
    - Text of post (HTML content) and attachments if any
* Referencing the student academic advising forum:
  + In addition to the defined forum features, the advisor will be presented with a button command to generate that specific student’s transcript.
  + The system will capture the student’s username which will be used to query the academic record units to retrieve and generate a report of the student’s transcript information.
* The system should backup all data daily, weekly, monthly and bi-yearly.
* System should undergo maintenance every month

# User Requirements

The Student Online Advisory System is required to recommend courses that can be done in the upcoming semester and provides alerts on degree status warnings, requirements to withdrawal and student’s academic advisor information.

Users are required to register and login using their university identification credentials and access their profiles.

Users should only be able to interact with their authorised forums. These interactions include commenting, starting threads or posting on the forum.

There is to be a forum dedicated to the academic advising process accessed only by students and their assigned advisor. Deans, Head of Departments (H. O. Ds), Teaching staff and Tutors can be assigned as academic advisors. The type of user account should determine which student transcript can be viewed.

Students are required to access their degree information and statistics; and degree audit information.

Deans and Head of Departments (H. O. Ds) are required to create and manage course or department academic reports.

# System Requirements

## Functional requirements

The Student Online Advisory System is required to provide facilities such that

* Each user/’s:
  + Is uniquely identified by his or her 9-digit university id number.
  + Shall be able to register and login into the system by inputting their university id number and corresponding password.
  + Will have access to their authorized forums on the website which can be used to share information within certain groups.
  + Has access to their student academic advising forum.
  + Has access to their associated bulletin boards.
  + Shall be able to view and update their profile.
  + Forum and profile information is restricted to their dedicated faculty and department.
* Each student:
  + Shall be able only to view their allocated bulletin boards.
  + Shall be able to reply, edit and delete (if permitted) their existing post within their allocated forums.
  + Shall be able to select any option within the Academic Advising feature.
  + Shall be given an alert that they are on warning or RTW.
  + Shall be provided with a list of recommended courses for the upcoming semester.
  + The student Profile should allow users to view their individual transcripts, degree statistics and degree audit information.
* Each tutor:
  + Is provided with all features of the student account.
  + Can be assigned as an academic advisor but within their student academic advising forum they cannot access student’s individual transcripts.
* Each teaching staff member:
  + Can post on their forum or bulletin board only for student access.
  + Who is assigned to be academic advisor, can view student’s transcripts upon request.
* Each Dean or H. O. D:
  + Can post on their forum or bulletin board only for student access.
  + Who is assigned to be academic advisor, can view student’s transcripts upon request.
  + Shall be able to create and manage department/course academic reports.
* System administrators:
  + Shall be able to set up security settings.
  + Shall be able to manage all views and restrictions on a user account.
  + Oversees the registration of user accounts and distribution of login information of existing and expired accounts.
  + Can monitor system logs together with generate and submit maintenance reports.
  + Oversees and monitors all activities to ensure all linked databases remain stable.
* The system will automatically grant users the required access level to various resources according to their type of account.

## Non-functional requirements

* Scalability- they system should be able to handle 10000 users accessing the site at the same time
* Capacity- to be able to have all old, current and new users to be enlisted unto the system
* Availability- information is restricted to each user’s limited access
* Response time- should reduce delay in information when hundreds of requests are given (TBC)
* Performance (TBD)
* Security- The system should automatically be able to authenticate all users with their unique 9-digit user id and password
* Each password
  + MUST contain at least 8 alphanumeric characters.
  + MUST contain at least one uppercase letter.
  + MUST contain at least one lowercase letter.
  + MUST contain at least one number.
  + MUST contain at least one special character.
* Usability: system design should be easily understood and user friendly to users. Furthermore, users of all skill levels of users should be able to navigate it without problems.
* Validation of accounts should not exceed 10 seconds.
* Downtime of system should not exceed 60 minutes.
* Users are allowed to make one account utilizing the university id only
* Once the user interacts with the system, they MUST receive a response from the system in under 15 seconds.
* System should be cross-platform on all devices
* Confirmation checks, screen prompts and alerts are presented in under 10 seconds with centre alignment to the users prior to specific interactions.
* Reports are to be generated in a PDF format for download.
* System information should be consistent and up-to-date as it relates to student and degree information
* Posts/Threads on the forums should be in reverse chronological order with exception to bulletin boards and FAQs.

# User Stories

## Scenarios/ User Stories

* Student user story:
  + Login/Register
* Forum
  + H.O.D / Dean Bulletin Board
  + H.O.D / Dean Forum
    - Questions, Queries and Concerns Thread
      * Add Post
  + Frequently Asked Questions
  + Teaching Staff Bulletin Board
  + Teaching Staff Forum
    - Questions, Queries and Concerns Thread
      * Add Post
* Get Academic Advising
  + Requirements to Withdrawal (RTW) Query/Notification
  + Recommended courses for upcoming semester
  + Degree Warning Prompt
  + Forum (dedicated to communicating with their advisor)
    - Add Post
* Student Profile
  + View Transcript
  + Generate Degree Statistics
  + View Profile
  + Generate Degree Audit
    - Number of Credits Remaining (Split into degree, foundation, elective)
* Logout
* Tutors (Student) user story:
  + Login/Register
* Forum
  + H.O.D / Dean Bulletin Board
  + H.O.D / Dean Forum
    - Questions, Queries and Concerns Thread
      * Add Post
  + Frequently Asked Questions
  + Teaching Staff Bulletin Board
  + Teaching Staff Forum
    - Questions, Queries and Concerns Thread
      * Add Post
  + Student Academic Advising Forum
    - Add Post
* Get Academic Advising
  + Requirements to Withdrawal (RTW) Query/Notification
  + Recommended courses for upcoming semester
  + Degree Warning Prompt
  + Forum (dedicated to communicating with advisor)
    - Add Post
* Student Profile
  + View Transcript
  + Generate Degree Statistics
  + View Profile
  + Generate Degree Audit (Either undergraduate or postgraduate)
    - Number of Credits Remaining (Split into degree, foundation, elective)
* Logout
* Teaching staff/ user story:
  + Login/Register
  + Forum
  + Student Academic Advising Forum (to communicate with requesting students)
    - Add Post
    - View Student Transcript
  + Teaching Staff Bulletin Board and Forum
    - Add Bulletin Board/Forum Thread or Post
* User Profile
  + View Profile information
* Logout

**Information is restricted to each department.**

* Deans and Head of Departments (H.O.D.s) user story:
  + Login/Register
  + Forum
    - H.O.D / Dean Bulletin Board and Forum
      * Add Bulletin Board/Forum Tread or Post
    - Student Academic Advising Forum (to communicate with requesting students)
      * Add Post
      * View Student Transcript
  + Faculty/Department Overview
    - Generate Course/Department Academic Report
    - Create New Course/Department Academic Report
  + User Profile
    - View Profile information
  + Logout

**Information is restricted to each department and /or faculty.**

* External Stakeholders (Administrative Staff) user story:

Generate Requested Student Records/Reports

Scenarios

1. Scenario name: displayRecommendedCourses

Actor instances: Student, Degree Audit Records and Student Academic Record Units

Flow of events:

1. Degree Audit Records and Student Academic Record Units are accessed for student’s course information.
2. This information is fed into a report template that generates a list of scheduled courses (core, elective and compulsory) for the upcoming semester.
3. The course list is presented to the student
4. Scenario name: degreeWarningPrompt

Actor instances: Student, Degree Audit Records and Student Academic Record Units

Flow of events:

1. The system accesses the Degree Audit Records to obtain the requirements needed for graduation.
2. The Student Academic Record Units are used to access student’s transcript information.
3. A semester-bound minimum requirement specification is generated using the Degree Audit Records.
4. The transcript information goes through a comparison test against the minimum requirement specification.
5. If the test is failed, the student is notified immediately or upon login
6. Else, nothing else is done and the user presented with the system’s main page interface.
7. The degree warning check continuously runs at the back-end.
8. Scenario name: requirementToWithdrawal

Actor instances: Student, Degree Audit Records and Student Academic Record Units

Flow of events:

1. The system accesses the Degree Audit Records and Student Academic Record Units.
2. Based on the Requirements to Withdrawal in the Student Academic Record Units, if the student falls within this category, they are notified immediately and flagged upon login.
3. Degree Warning Prompts are actively issued to avoid reaching this status.

4. Scenario name: viewTranscript

Actor instances: Student Academic Record Units, Students and Academic Advisors

Flow of events:

1. Upon selection, the system accesses the Student Academic Record Unit for student’s transcript information.
2. This information is used to create a report as a downloadable PDF file.
3. This document is presented to the corresponding user.

5. Scenario name: getDegreeStatus

Actor instances: Students and Student Academic Record Unit

Flow of events:

1. Student Academic Record Unit are accessed for student’s course information.
2. When accessed, a statistical formula is used to calculate a student’s projected GPA score.
3. In addition, the previous semester and overall GPA score information is found.
4. A report is generated using this information as a downloadable PDF file.
5. This report is displayed to the student.

6. Scenario name: getDegreeAudit

Actor instances: Students and Degree Audit Records

Flow of events:

1. The student’s fetched information from the Degree Audit Records is generated as a downloadable PDF file.
2. This report is displayed to the student.

7. Scenario name: communicateWithAdvisor

Actor instances: Students, Academic Advisors and displayRecommendedCourses

Flow of events:

1. An academic advisor (lecturer or tutor) account is associated/assigned to set of students for advising.
2. Upon selection, the assigned advisor information and a confirmation message are presented to the student.
3. Once confirmed, the student can post any comments, questions or queries as it relates to the feedback received from the recommended course selection page on the forum.
4. If the student posts, an acknowledgment is sent to the student that the post has been successfully submitted to their advisor.
5. Any feedback received from the forum is alerted to the student immediately or upon login.

8. Scenario name: adviseStudent

Actor instances: Academic Advisors, Students and viewTranscript

Flow of events:

1. Upon login, users assigned as academic advisors are notified of incoming advising feed to addressed.
2. Upon selection of posts, advisors are directed to the forum.
3. A viewTranscript option is presented where advisors can access that particular student’s transcript
4. This used to help personalize advice and gain better understanding of the student.
5. When the lecturer posts, an acknowledgment is sent to the user that the post has been successfully submitted to the student.
6. Any feedback received from the forum is alerted to the advisor immediately or upon login.

# Use Cases

## Simple Use Cases from User Stories

Actors

* Students
* Academic Advisors
* Deans and Head of Departments
* Student Academic Record Units
* Degree Audit Records

Use Cases:

* Get Academic Advising
* Data Report
* Access Forum
* Login

## Use Case Diagram

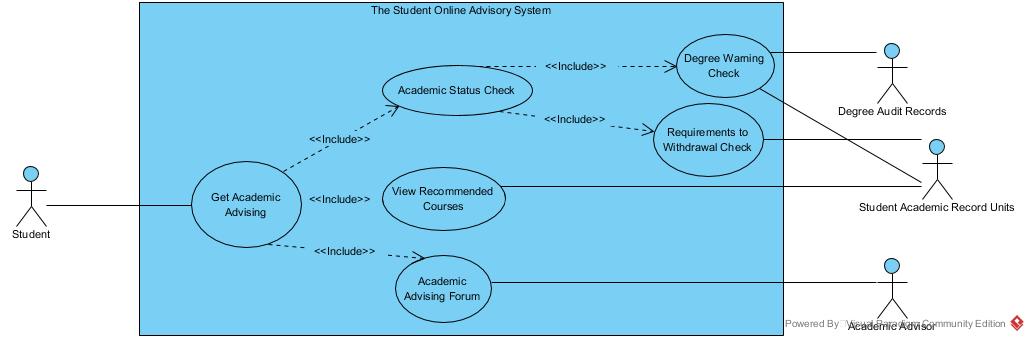
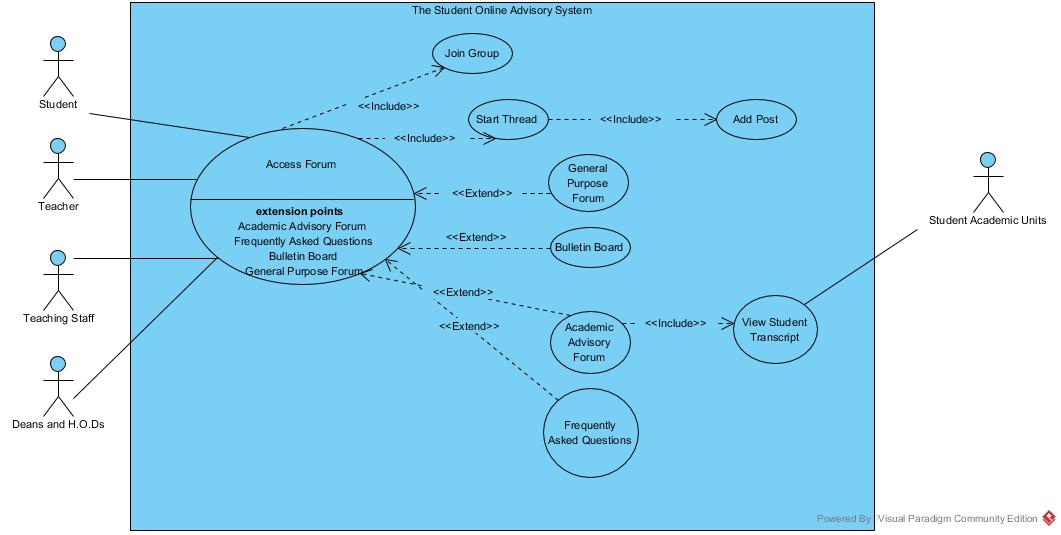


Figure1.1:Use Case for Get Academic Advising

Figure1.2: Use Case for Access Forum

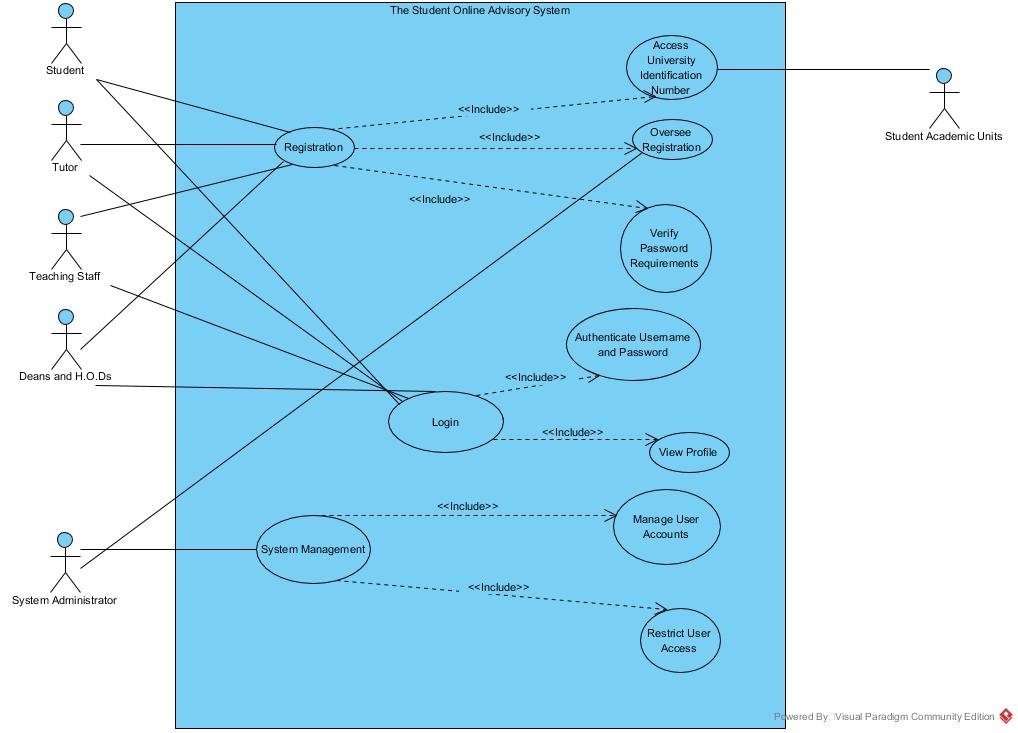


Figure1.3:Use Case for Login

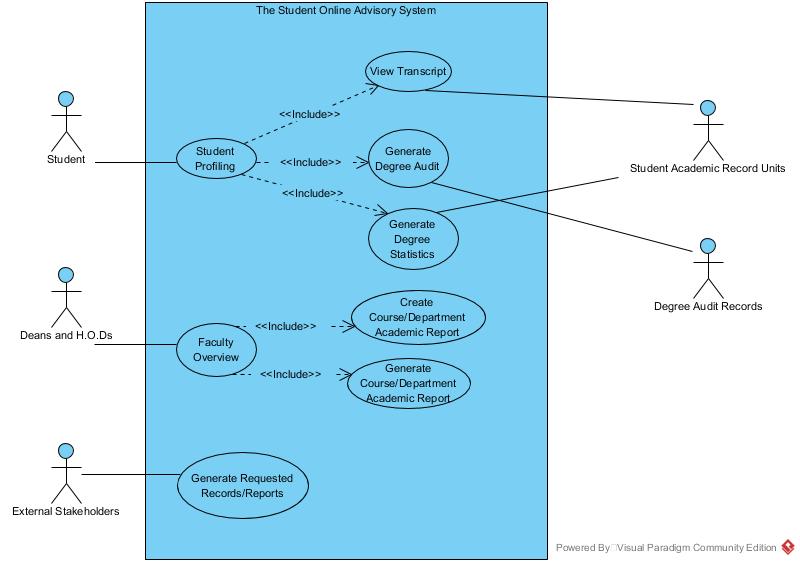


Figure1.4:Use Case for Generate a Report

## Ranking of Use Cases

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Use-Case Name | Ranking Criteria (1-5) | | | | | | Total Score | Priority |
| 1 | 2 | 3 | 4 | 5 | 6 |
| Registration | 3 | 4 | 4 | 4 | 5 | 1 | 21 | High |
| System Management | 4 | 1 | 5 | 2 | 3 | 3 | 18 | Medium |
| Access Forum | 5 | 2 | 3 | 3 | 4 | 3 | 20 | High |
| Student Profiling | 3 | 4 | 3 | 2 | 3 | 1 | 16 | Medium |
| Generate requested reports/records | 1 | 1 | 2 | 1 | 2 | 3 | 10 | Low |
| Faculty Overview | 2 | 2 | 2 | 1 | 2 | 2 | 11 | Low |
| Get Academic Advising | 5 | 2 | 4 | 5 | 5 | 4 | 25 | High |

Table1: Ranking of Use Cases

## Expanded Use Cases

|  |  |  |
| --- | --- | --- |
| **Use-Case Name:** | Registration | |
| **Use-Case ID:** | SOAS-R1 | |
| **User-Case Type Business Requirements:** | 1. Users should only be able to interact with their authorized forums. 2. There is to be a forum dedicated to the academic advising process 3. Each teaching staff assigned as an academic advisor can view student’s transcripts upon request. | |
| **Priority:** | High | |
| **Source:** | 1. Main navigation page selection 2. Selection within the academic advising forum | |
| **Primary Business Actor:** | System User (Student, Tutor, Teaching Staff, Deans and H. O. Ds) | |
| **Other Participating Actors:** | Student Academic Record Units | |
| **Other Interested Stakeholders:** | System administrators – interested in the proper administration of access levels to authorized forums for all users. | |
| **Description:** | This use case describes the entails the event of users accessing the forums provided by the system. There are two scenarios where users can access forums which are: 1. The general purpose forums within the main navigation page and 2. the dedicated academic advising forum. Each forum is broken up into a collection of threads. For the academic advisor view of the advising forum, there is a feature to generate their assigned student’s transcript. | |
| **Preconditions**: | The user must be successfully logged onto the system and allocated to the required forums by the system administrator. | |
| **Trigger**: | Upon selection of the forum feature on the main navigation page or within the get academic advising feature. | |
| **Typical Course of Events:** | **Actor Actions** | **System Actions** |
| **Step 1:** Upon successful login, the user selects the forum feature on the main navigation page or student academic advising forum.  **Step 6:** User can access various threads to view new messages, add posts or start a new thread(if permitted) | **Step 2:** The system does a lookup on the account based on the access specifications outlined by the system administrator.  **Step 3:** The system obtains all messages (read and unread).  **Step 4:** Using a pre-defined template for the forums and threads, the system renders the information for the user’s display.  **Step 5:** If there are any new messages, the system generates notifications alongside the thread it belongs to. |
| **Alternative Courses:** |  |  |
| **Conclusion:** | This use case concludes when the user is given a report that they have been registered. | |
| **Post-conditions:** | Each user is presented only with authorized forum information and details as outlined by the system administrator. | |
| **Business rules:** | 1. Forum interactions include starting threads, adding threads and adding/deleting posts. 2. Users who attempt to use the site to disrupt others or cause harm can be temporarily banned from the site | |
| **Implementation constraints and specifications:** | 1. One user to one account 2. No duplicate emails allowed for registering 3. Unique user login id’s given to all users. | |
| **Assumptions:** | N/A | |
| **Open Issues:** | N/A | |

Table2.1: Expanded Use case on Registration

|  |  |  |
| --- | --- | --- |
| **Use-Case Name:** | Access Forum | |
| **Use-Case ID:** | SOAS-R2 | |
| **User-Case Type Business Requirements:** | 1. Users should only be able to interact with their authorized forums. 2. There is to be a forum dedicated to the academic advising process 3. Each teaching staff assigned as an academic advisor can view student’s transcripts upon request. | |
| **Priority:** | High | |
| **Source:** | 1. Main navigation page selection 2. Selection within the academic advising forum | |
| **Primary Business Actor:** | System User (Student, Tutor, Teaching Staff, Deans and H. O. Ds) | |
| **Other Participating Actors:** | Student Academic Record Units | |
| **Other Interested Stakeholders:** | System administrators – interested in the proper administration of access levels to authorized forums for all users. | |
| **Description:** | This use case describes the entails the event of users accessing the forums provided by the system. There are two scenarios where users can access forums which are: 1. The general purpose forums within the main navigation page and 2. the dedicated academic advising forum. Each forum is broken up into a collection of threads. For the academic advisor view of the advising forum, there is a feature to generate their assigned student’s transcript. | |
| **Preconditions**: | The user must be successfully logged onto the system and allocated to the required forums by the system administrator. | |
| **Trigger**: | Upon selection of the forum feature on the main navigation page or within the get academic advising feature. | |
| **Typical Course of Events:** | **Actor Actions** | **System Actions** |
| **Step 1:** Upon successful login, the user selects the forum feature on the main navigation page or student academic advising forum.  **Step 6:** User can access various threads to view new messages, add posts or start a new thread(if permitted) | **Step 2:** The system does a lookup on the account based on the access specifications outlined by the system administrator.  **Step 3:** The system obtains all messages (read and unread).  **Step 4:** Using a pre-defined template for the forums and threads, the system renders the information for the user’s display.  **Step 5:** If there are any new messages, the system generates notifications alongside the thread it belongs to. |
| **Alternative Courses:** | **Alt-Step 1:** If an academic advisor (other than a tutor) is responding to their assigned student within the academic advising forum and selects the feature to generate a student transcript.  **Alt-Step 6:** The user is provided with a downloadable file for the transcript document. | **Alt-Step 2:** The system captures the user account’s username and runs a query in the Student Academic Record Units to obtain the student’s transcript information.  **Alt-Step 3:** The information is provided to the user as a downloadable PDF File. |
| **Conclusion:** | This use case concludes when the presentation of forum information and features to the user. | |
| **Post-conditions:** | Each user is presented only with authorized forum information and details as outlined by the system administrator. | |
| **Business rules:** | 1. Forum interactions include starting threads, adding threads and adding/deleting posts. 2. Posts/Threads should be in reverse chronological order. 3. Users need to be notified of new messages in a timely manner. | |
| **Implementation constraints and specifications:** | 1. Web screen interface for users (presentation and navigation) within a pre-defined template. 2. Back-end connection to the Student Academic Record Units 3. Each thread provides a drop-down menu to visit other threads in the forum. 4. Important threads and bulletin boards are fixed at the top of the forum 5. Users should be notified of new messages upon login or in no longer than 30 seconds if already logged in. | |
| **Assumptions:** |  | |
| **Open Issues:** | 1. Need to determine how academic advisors are assigned to students. 2. Need to determine how to establish secure and reliable connection to Student Academic Record Units. | |

Table2.2: Expanded Use Case on Access Forum

|  |  |  |
| --- | --- | --- |
| **Use-Case Name:** | Get Academic Advising | |
| **Use-Case ID:** | SOAS-R3 | |
| **User-Case Type Business Requirements:** | 1. Users should be able to get advising and suggestions from teaching staff. 2. Users should be suggested what subjects would be best for them to take on during the semester. 3. Users interacting with the system can only do so during specific periods in time at the beginning of the semester | |
| **Priority:** | High | |
| **Source:** | 1. Semester Sign up 2. Registration 3. Subject Selection | |
| **Primary Business Actor:** | System User (Student, Teaching Staff) | |
| **Other Participating Actors:** | N/A | |
| **Other Interested Stakeholders:** | Deans and H. O. Ds – assuring that there is some structure to the lecturer’s advising by providing suggestions for general cases. | |
| **Description:** | This use case describes the advising process after the fact that a user has registered for the semester. Following the registration process, are to select the courses they wish to pursue throughout the upcoming semester and once that has been done, are required to receive academic advising from the teaching staff as to if their choices are recommended for the respective user at the time, taking into consideration the users, GPA, number of courses selected, current semester etc. | |
| **Preconditions**: | The user must have an account on the system, be successfully logged in, begun the registration process and selected courses to pursue during the semester. | |
| **Trigger**: | After choosing the subjects the users wish to do, the users are directed to the academic advising forum. | |
| **Typical Course of Events:** | **Actor Actions** | **System Actions** |
| **Step 1:** After choosing the subjects the users wish to do, the users are directed to the academic advising forum.  **Step 4:** The student can create a thread asking a specific question directed to the advising staff.  **Step 5:** Advising staff can respond to the specific question to the user and provide suggestions.  **Step 6:** Advising staff confirms that the student completed the advising process. | **Step 2:** The system alerts the teaching staff responsible for advising of the new student for academic advising.  **Step 3:** The system produces a list of frequently asked questions for the user to look through.  **Step 7:** The system saves a record of the user as “Completed Academic Advising.” |
| **Alternative Courses:** | **Alt-Step 1:** User can request a meeting with the advisor.  **Alt-Step 2:** Advisor accepts request and organizes a meeting day & time and enters it into the system.  **Alt-Step 4:** Advising staff meets with the user and answers and queries the user may have.  **Alt-Step 5:** Advising staff after finishing their meeting with the user marks down the user as “Completed Academic Advising” and inputs it into the system. | **Alt-Step 3:** System records the date & time for the meeting between the user and advising staff  **Alt-Step 6:** The system saves a record of the user as “Completed Academic Advising.” |
| **Conclusion:** | This use case concludes when the user is saved as “Completed Academic Advising”. | |
| **Post-conditions:** | User is given an alert that they are clear to do and view their courses. | |
| **Business rules:** | 1. A user is entitled to a meeting if they request one. 2. Advising staff need to be alerted timely about new students for advising. 3. Users should be given the standard help during academic advising. | |
| **Implementation constraints and specifications:** |  | |
| **Assumptions:** | N/A | |
| **Open Issues:** | 1. Need to determine how academic advisors are assigned to students. | |

Table 2.3: Expanded Use Case on Get Academic Advising

# Sequence Diagrams

## Top use Case

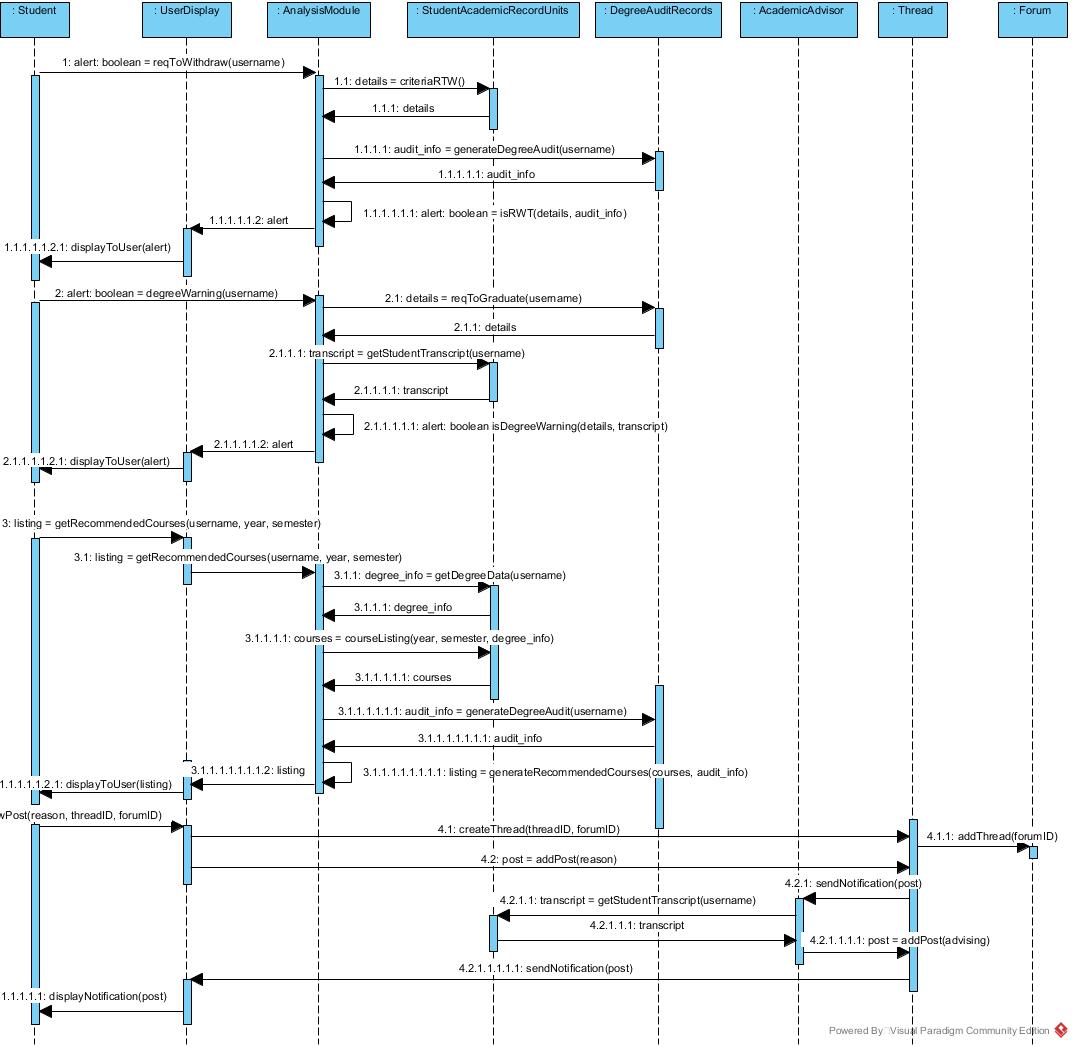


Figure2: Sequence Diagram for Get Academic Advising

# Class Diagrams

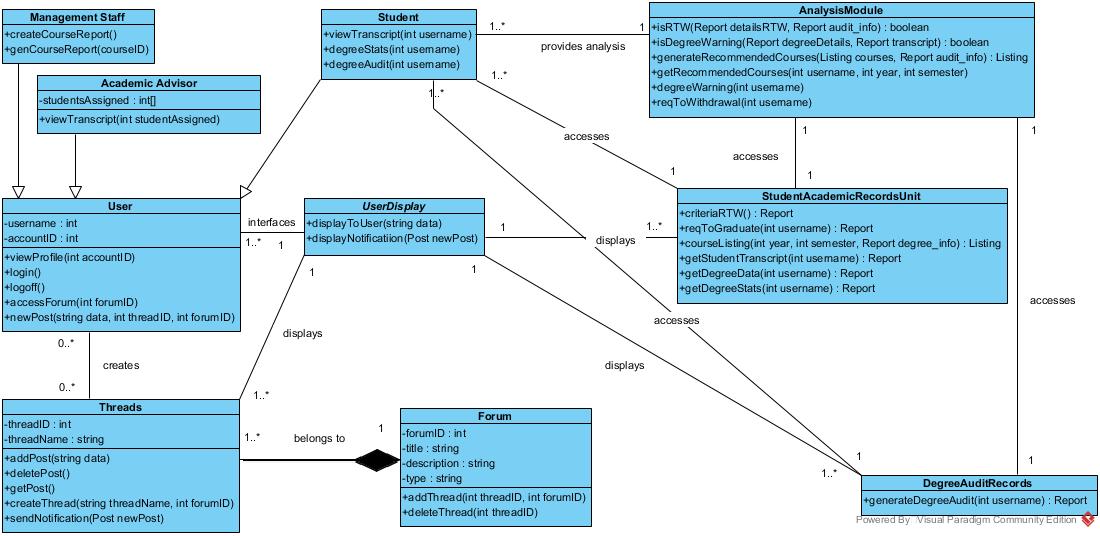


Figure3: Class Diagram for Student Online Advisory System

# Testing Plans

## Method Cases:

* newPost(string data, int threadID, int forumID)
* createThread(string threadName, int forumID)
* courseListing(int year, int semester, Report degree\_info): Listing
* getStudentTranscript(int username): Report
* generateRecommendedCourses(Listing courses, Report audit\_info): Listing

1. newPost(string data, int threadID, int forumID)

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Valid** | **Invalid** | **Boundary** |
| data | data.length > 0 | data.length == 0  data == null | 0 < data.length < = 1000 |
| threadID | threadID > 0  threadID instanceof int  Thread contains threadID | threadID < = 0  threadID == null | 0 < threadID < = 1000 |
| forumID | forumID > 0  forumID instanceof int  Forum contains forumID | forumID < = 0  forumID == null | 0 < forumID < = 10 |

1. createThread(string threadName, int forumID)

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Valid** | **Invalid** | **Boundary** |
| threadName | threadName.length > 0 | threadName.length == 0  threadName == null | 0 < threadName.length < =1000 |
| forumID | forumID > 0  forumID instanceof int  Forum contains forumID | forumID < = 0  forumID == null | 0 < forumID < = 10 |

1. courseListing(int year, int semester, Report degree\_info): Listing

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Valid** | **Invalid** | **Boundary** |
| year | 1,2,3,4,5  year instanceof int | anything else  year == null | 1 < = year < = 5 |
| semester | 1,2,3  semester instanceof int | anything else  semester == null | 1 < = semester < = 3 |
| degree\_info | degree\_info instanceof Report | degree\_info == null | - |

1. getStudentTranscript(int username): Report

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Valid** | **Invalid** | **Boundary** |
| username | username instanceof int  StudentAcademicRecordsUnit contains username | username == null | - |

1. generateRecommendedCourses(Listing courses, Report audit\_info): Listing

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Valid** | **Invalid** | **Boundary** |
| username | courses instanceof Listing | courses == null | - |
| audit\_info | audit\_info instanceof Report | audit\_info == null | - |

# Risk Management

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Risk** | **Probability** | **Effects** | **Affects** | **Strategy Category** | **Strategies** |
| Experienced staff leaving the project before completion | Moderate | Serious | Project | People | Contracts can be implemented for the project duration |
| There is a change of organisational management with different priorities | Low | Catastrophic | Project | Organizational | Prepare a briefing document for senior management showing the importance of the project to the business |
| Essential hardware isn’t delivered on time | Moderate | Serious | Project | Technology | A briefing can be done for the senior management and an extension can be requested |
| If there are more requirements to change than expected | High | Serious | Project and Product | Requirements | Derive traceability information to assess requirements change impact; maximize information hiding in the design |
| Specifications of essential interfaces are not available on time | Moderate | Serious | Project and Product | Requirements | A meeting with senior management can be done to discuss the necessary specifications for the project |
| The size of the system has been underestimated | Low | Catastrophic | Project and Product | Estimation | Certain features that are not essential to the operation of the business can be removed |
| CASE tools which support the system do not perform as anticipated | Low | Tolerable | Product | Tools | New methods or CASE tools can be implemented which can better help the system |
| The underlying technology on which the system is built is replace with new technology | Moderate | Serious | Business | Technology | Hiring skilled staff within the business who are able to cope with technological change or have staff complete the required training to use the system |
| A competitive product is marketed before the system is complete | Low | Tolerable | Business | Organizational | The product can be analysed and the new system can perform more efficiently and be more cost-effective |
| Organizational financial cannot complete the full project | Moderate | Catastrophic | Project | Organizational | Preparing a briefing document for senior management showing the projects contribution to the business and present reasons why more funds are needed |
| Expert staff gets an illness | Moderate | Serious | Project | People | Reorganize team so that there are more overlap of work and people therefore understand each other’s jobs. |
| Components needed maybe defective | Low | Serious | Project | Technology | Replace potentially defective components with bought-in components of known reliability |
| Database performance is not as efficient as wanted | Low | Serious | Product | Technology | Investigate the possibility of buying a higher-performance database. |
| Hiring the required personnel for the project is difficult | Moderate | Serious | Project | People | Alert customer to potential difficulties and the possibility of delays; investigate buying-in components. |
| The scope of the project increases | Moderate | Serious | Business | Organizational | The database size and software speed should be increased to fit the new needs of the system |

# Cost Estimation

The **Co**nstructive **Co**st **Mo**del II is a model geared towards estimating the cost, effort and schedule when planning a new software development activity. It consists of three models called the Applications Composition, Early Design and Post-architecture which are listing in increasing fidelity provided.

This model aids with financial decisions involving a software development effort, setting project budgets and schedules and negotiating trade-offs among software cost, schedule, functionality, performance and quality together with implementing a process improvement strategy.

# Interfaces

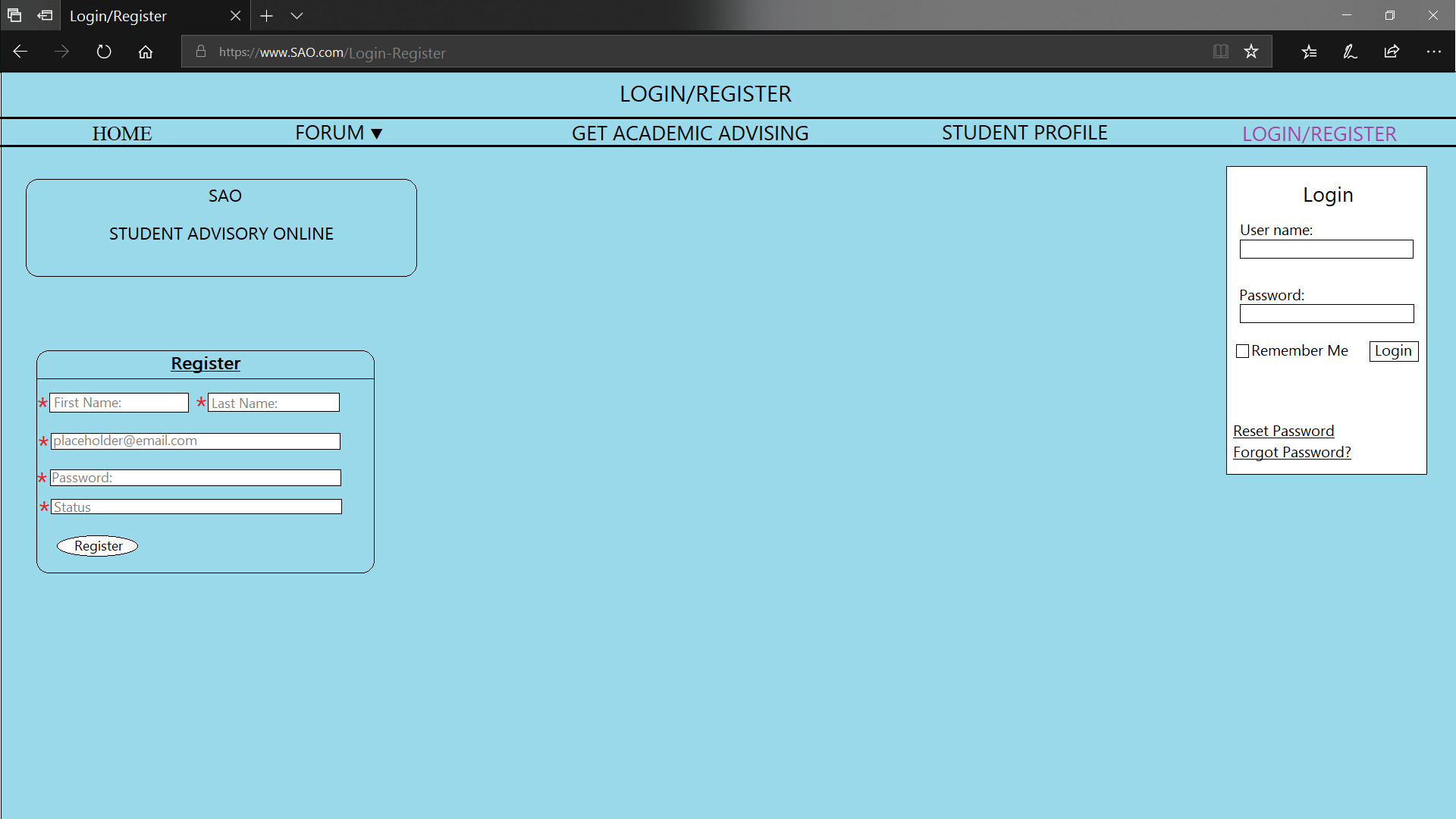


Figure4.1: Login Interface

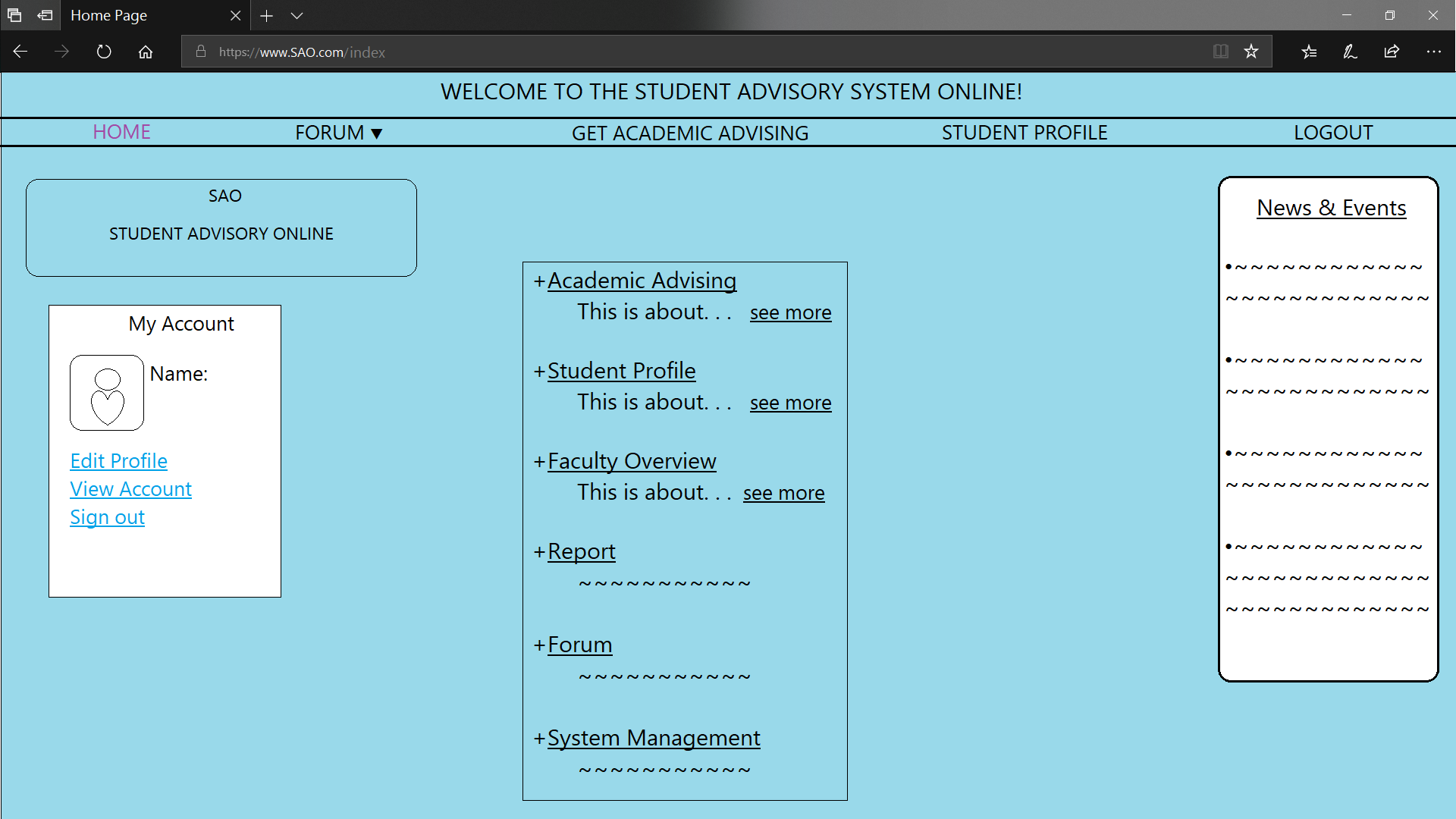


Figure4.2: Home page Interface

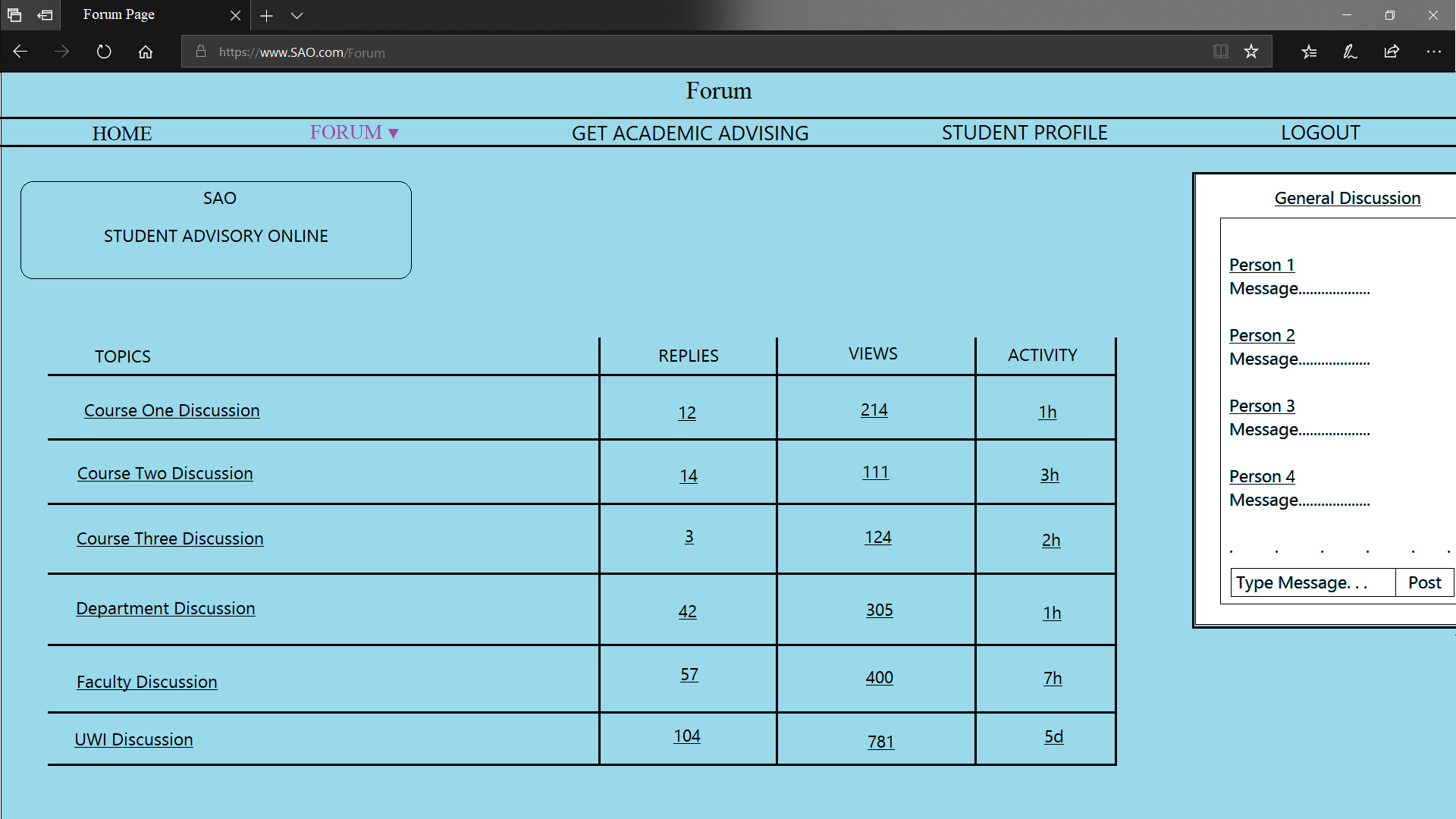


Figure4.3: Forum interface

# References

# Glossary of Terms