***ETPB Software Design™*** *| 212 McAlister Dr, Saint John NB*

*Architecture Draft*

*Studentable: A University Course Registration Management System*

**Architecture Draft**

**Purpose**

The purpose of the architecture draft is to ensure all stakeholders agree on the systems intended purpose, following use-case scenarios. The general description of Studentable will be stated with its features, and use-case scenarios will be made for each type of connection to the software.

**General Description**

The purpose of this project is for *ETPB Software Design****™*** to complete a framework University Registration Management System, formally known as Studentable.

Students and professors will sign into the system using their student emails and self-appointed passwords. This system will be entirely web based, requiring no installation of software on university computers. It will be hosted on the university's public domain, therefore no need for the use of a VPN to access it from home or work.

This system is be able to handle large traffic as in a worst-case scenario: students, professors, and admins can all be accessing it.

Studentable is an organized course catalog for students, instructors, and administration to use. The system allows students at a university to browse a course catalog that provides information on various course offerings they need for their semester. Students will be restricted to only four course offerings per semester, with two alternatives in the case that the other courses are cancelled or filled.

- This system detects when a course should be cancelled (less than 3 students), and when a course is full (more than 10 students).

- The system will send the student's billing information to the university's billing system upon registration completion.

A strict grace period is enforced where students are permitted to change their schedule at the beginning of the semester. During this grace period there is a special student portal that only students have access to using their school credentials. In this portal they can add/drop courses.

A special portal exists through the management system for professors which enables them to see which courses they are instructing, and who are the students enrolled.

Admin accounts are created for school faculty using the logon admin@university-domain.com. This account will allow the admin to perform customizations on the site and modify course information. The admin account is also responsible for the automation of creating new accounts and freeing up old accounts, which will be done through automation.

**Use Case Scenarios**

Student:

* From the viewpoint of a student, they will have a designated login/password authentication process before receiving access to the university registration management system.
* The student will query the registration system for information on the course(s) they wish to select/drop. A student is only able to either select/drop their four classes of choice with two reserved options in the case that the classes have an insufficient number of students or too many. Students may change their schedule as much as they wish until the date designated by the specific university’s registrar.
* After a successful registration in a course, the student will be billed by the registration management system and will then have to pay the course fee to the university’s own billing system. Once this fee is paid, the student will receive a proof of enrollment and payment.
* The student will be able to see the classes they are enrolled in, and also will be issued their bill for the semester.
* Students actions on the webpages will be logged for administrative purposes.

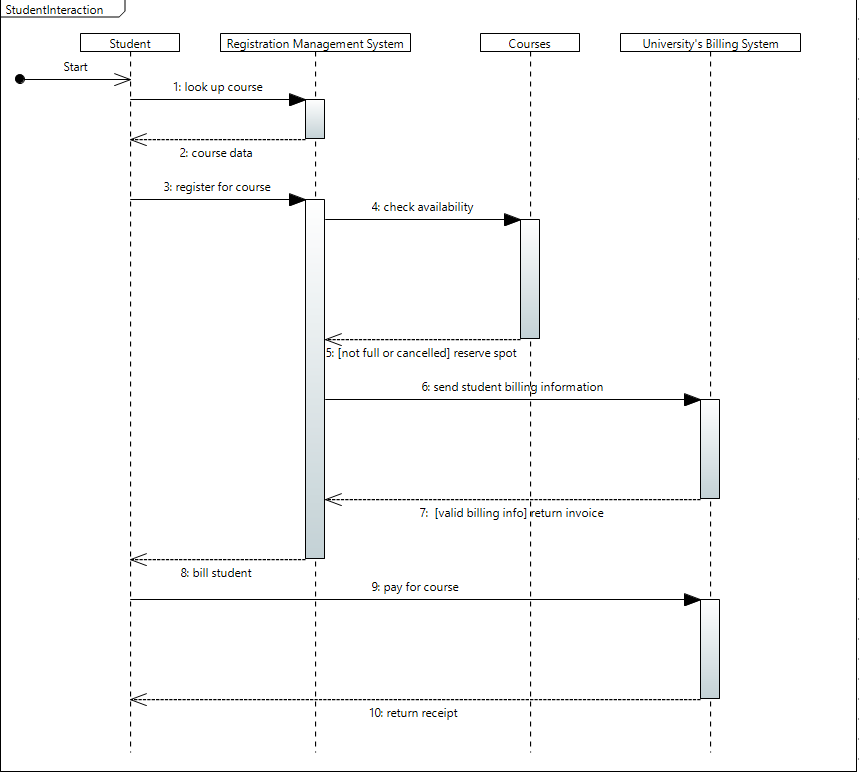
Instructors:

* The instructor will have to login with a designated login/password authentication process before being allowed access.
* When granted access, instructors will be able to query the attendance lists *only* on the courses they are teaching.
* Instructors actions on the webpages will be logged for administrative purposes.

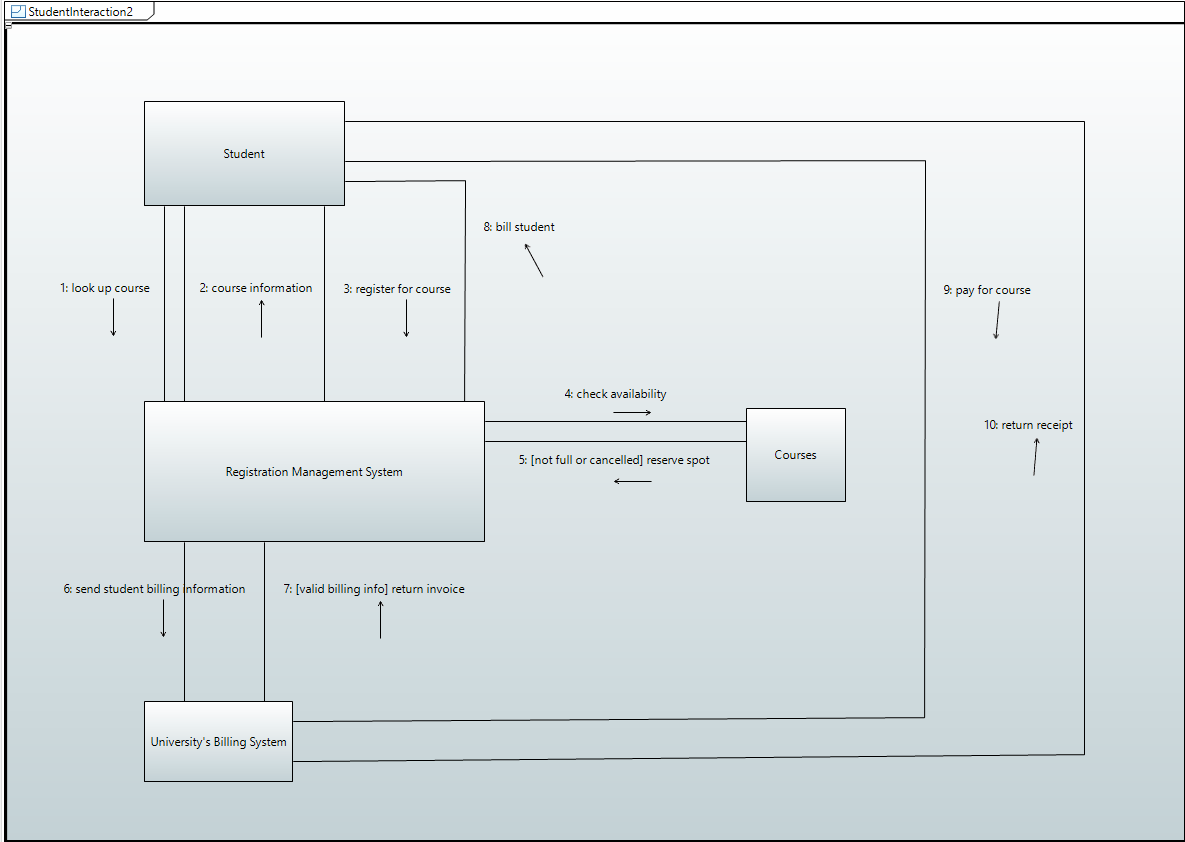
Admin:

* An admin’s perspective of the program begins by their specific login authenticity ([admin@university-domain.com](mailto:admin@university-domain.com)).
* Admins will have access to student view (what a student sees), professor view (what the professor sees), and the admin panel.
* Admins will be able to query for class information, class attendance lists, and student information. As well as access the activity log for the program, which details each action performed by professors, students, and admins in the system.
* Accounts of both instructors and students will automatically be made, however there is an option for the administrator to manually make the account.
* Admin’s can manually enroll or drop a student from a specific course given the student was unable to themselves.
* Admin’s are the only users with access to modify the University’s site page and course pages within the registration system. Modifications include adding new/removing old courses, customization according to University colors/logo, deciding what information is displayed on a course page.

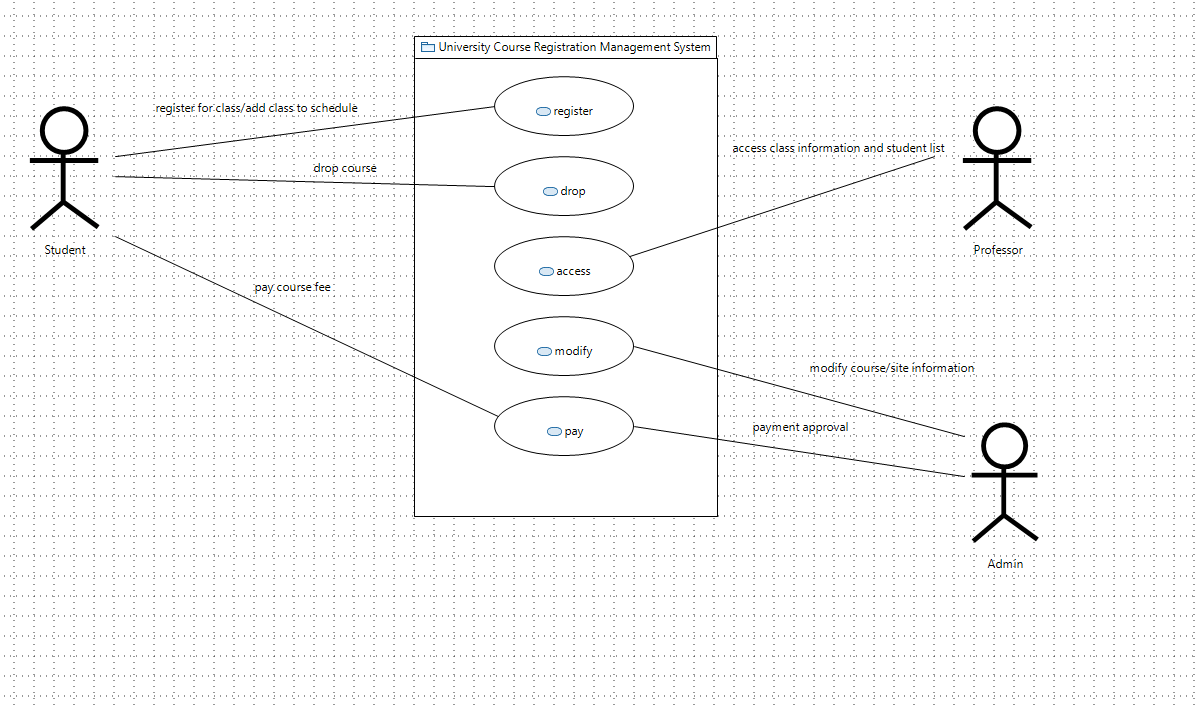
**Sequence Diagram (for student)**



**Communication Diagram (for student)**



**Use Case Diagram**



**Architecture Style**

Chosen Style: 3 – Tier Architecture

**General Purpose:**

* The 3 – Tier Architecture style is a client-server architecture in which the user-interface, the functional process logic and the data storage are maintained as independent modules. This is ideal for web-based applications where a user must interact with a large database through an easy to use web interface.
* **Advantages**:
  + Speed of development: Each module is developed independently of each other; therefore, different layers of the application can be worked on with minimal impact on the others. This allows us to improve on the product with greater speed without worrying about creating bugs in other parts of the application.
  + Team efficiency: Teams can focus on developing one tier of the application at a time, allowing all their efforts to be expended on one thing, improving development efficiency.
  + Scalability: By separating the application into 3 tiers, we can scale each tier independently based on when it is needed.

**Implementation:**

* **User Interface:** Top level of the application. Displays relevant information to the user and allows the user to interact with the data stored in the data tier. This is the only module the user will directly interact with. Will be well presented and easy to look at and traverse.
* **Logic Tier:** Controls the application’s functionality, the business rules of the program. Requests and queries sent by the user in the user interface tier will be parsed in this tier before being sent to the data store. This tier must be robust and not prone to failure as it dictates how the applications functions
* **Data Tier:** Includes persistent data that exists on a sperate service to the rest of the application, whether it be a local data server or a cloud database. Provides an API to the logic tier that allows for methods of accessing and manipulating data without compromising the data store.