**Requirement Engineering**

**Elicitation**

For a course calendar system (Studentable), the requirements engineering technique used was a functional approach. The course calendar system, in general, must be fast, user friendly, and able to handle heavy traffic. The user interacts with a database making system requirements are important; therefore, the functional approach is the optimal choice.

The initial requirements were decided upon through brainstorming sessions. This encouraged the stakeholders to contribute ideas in an informal fashion.

The different type of users are students, professors, and admins. The system is designed for schools therefore must be able to handle many users. Studentable must also be secure since it has access to the user’s personal information.

Students will be able to pick and register for courses. Professors will be able to see which courses they are teaching and the students who registered for their course. Admin will be able to manage the system making changes as needed.

**Specifications**

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

By adhering to completion of a technical specification documentation, *ETPB Software Design****™*** therefore are allowing all project team members to build a stronger idea of the requirement engineering ideology.

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.

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.

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. The system must detect 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.

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.

Students and professors must be able to sign into the system using their emails and self-appointed passwords.

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

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.

**Validation**

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.

The user-interface must be organized and simple for the user to easily travel through.

Only one instance of each login-user may be open at any specific time, otherwise all accounts will be locked out.

The website will be of a consistent color scheme (changeable by the admin, most commonly the school colors).

The user will be allowed to click on their own specific name and change their password or refer to their school-custom email.

A course timetable will be available for students to check their courses by day.

A specific entity will be created for each class, with data such as the teacher, time, location (with the option of AD if required). A student will be able to “register” where a request will be sent to enroll in that specific class.

When all students are enrolled, the class timetable will be published.

The web server must be able to handle requests at a speed not prompting latency to other accounts.

**Negotiation**

All developers through the project are working off a voluntary basis. There are no incentive bonuses, hourly rates or tax deductions. Workers are incentivized to work off the goal of mentorship and honing their craft in the software development field.

Work will be distributed based on the individual’s availability and skills. The team has overlapping skills; however, each individual will overlook a specific stage:

|  |  |
| --- | --- |
| Ethan Garnier | Head Software Development/ Requirement Analysis |
| Tyler Travis | Documentation / Version Control |
| Ben Irvine | Implementation / Testing |
| Pranveer Singh | Software Maintenance / Documentation |

**Negotiation**