Scenarios

These scenarios are examples of different actors working on the student registration system. The actors are the students who will register for classes, the registrars who will approve and manage current class lists and approve additions or drops of classes, and finally the administrator who will be able to edit class lists and perform maintenance on the system. The admin user would have all of the access abilities of the student and the registrar. The registrar will have the abilities of the student as well. The student will only have the abilities of the student user.

Scenario 1:

Students View Point:

Initial Assumption: Jim is a college student at Online University of Practice, and he is enrolled in the registration system when admitted. Before each semester starts, he needs to enroll in the classes for next semester on the online system, and during the semester, he needs to check his class schedule through the same system or drop any unwanted class.

Normal Student Access: When Jim wants to check the class schedule on the system, he only needs to login the system using the username and password given by administrators, and click on the class schedule. If he wants to enroll classes for next semester, he first needs to log in the system with the username and password, and then would be able to see all the classes offered during the next semester and the remaining seats. After seeing the offered classes, he is able to choose classes he wants to enroll in and sends the request to the system which will notify the registrar that a student wants to be added to the course. If he wants to drop a class, he needs to go to his current class schedule, selects the class that he wants to drop, and sends the request to drop the class.

What could go wrong: If the class that Jim chose is already at the maximum occupancy, he will not be able to send the add request. Instead, the system will notify Jim that the class is full, and ask if he wants to send the request to be added on waiting list. If multiple users are being added to the same class at the same time the class addition requests could conflict so we will need to create a way to know what time the process was sent, and make a first come first serve system. This could also be handled by the registrar account to decide who will be allowed to enter the class.

Other Activities: Jim can change the initial password provided by administrators on the system by inputting the old password and then creating a new password via the online system. The password must contain a special character and a capital letter for security reasons.

Scenario 2:

Registrars View Point:

Initial Assumption: Jeff is working for Online University of Practice as a Registrar. His daily work is to confirm the adding/dropping request sent by students and then, enroll or drop the class that is required.

Normal Registrar Access: Jeff uses the username and password provided by the admin to login the system. Once he is in the system, he is able to see all the requests by students and put the class information into the student's' personal account. If the class is not full, he can directly confirm the adding/dropping requests. But if the required class is full at first and gets a dropping request, he would need to remove the student from the class and check if there are students in the waiting list. If so, he needs to add the first student in the waiting list into the class. If one class is on high demand, he also needs to notify the admins to change the size of the class or add a new class.

What could go wrong: If Jeff mistakenly adds a student into a full class, the system should be able to check and notify him. In case that he forgets to add the student in the waiting list into the class after some others drop the class, the system should also notify him when he removes one students from a full class that has the waiting list. The system could instantly add the first in the queue to the class once a student has been removed.

Other Activities: Jeff can change the initial password provided by administrators on the system by input the old password and create a new password via the system.

Scenario 3:

Administrator View Point:

<u>Initial Assumption:</u> Jake is working for the University as a system administrator. His daily job is to maintain the accounts in the school registration system, change the class size, and add or delete the classes in the system.

Normal: Every Time a student is admitted to the school, Jake is notified that a student is admitted, and the system generates a username and password. Jake needs to put the name of the student in the account information, and make sure it is correct with the online form the student created. When the student has graduated, he needs to delete the student account from the system. The system should have a list of students graduating and auto deletes them based on id number. He needs to perform the same procedure for a registrar when hired or fired. Before the enrollment procedure of students starts, he needs to add the classes offered for next semester into the system with the information of class size and class name. If he gets the information from the registrar, he needs to add the new class as requested.

What could go wrong: Jake may accidently delete the wrong accounts that should not be deleted. The system should have one backup database with the information a week ago,. If he deleted the wrong account, he should be able to find the information from the backup database and get back information stored a week ago, the system might also generate duplicate user names and id numbers. Jake must make sure this is corrected.

Requirements

Functional Requirements: a statement of the services the system should provide, and its interactions with individual actors.

- Access to a data base
- Networked (remote access)
- Secure accounts
- password storage
- multiple account levels and permissions
- Username and ID number generation
- duplicate detection (accounts, id numbers, classes)
- Data Editing
- high traffic capabilities (registration day)

Non-Functional Requirements: constraints on the services or functions of the system. apply to the system.

- Dependable on release with full functionality
- Patch support to update and refine the software
- security to prevent access to user accounts or information
- Different UI for different permissions
- Notification System (down times, full classes, accepted requests sent to user E-mail)
- Password reset/ change capabilities
- limited downtime (students do not want their class lists to be down for long during registration)

User Requirements: The service the system provides to the user and the operational constraints

- user friendly UI based on accounts
- Ability to edit or request services from different accounts
- Storage of account information
- notification of request processes
- ability to edit data
- multiple account tiers.
- online application

System Requirements: a description of the system's functions, services, and operational constraints.

- web based application accessed through a standard browser
- access to a data base storing user information
- multiple account creation and generation
- multiple account privileges
- Web based user interface

- notification system for accounts that can send out E-mails.
- security measures for password resets, and access abilities
- duplicate account/ information detection and handling
- error handling
- large traffic capacity and filtering.

Security Questions: a list of basic questions that could be used to enhance the security of a user account, for example if a user wants to change their password they will have to answer these questions the same way they did when first creating an account.

- What is the name of your favorite pet?
- What is the name of the High school you graduated from?
- What is your mother’s maiden name?
- In what city does your furthest sibling live?
- What was the color of your first car?
- What is your eldest sibling’s middle name?
- What is your favorite color?
- What is your favorite resaurant?
- What was the name of you high school mascot?
- What was your first part time job?

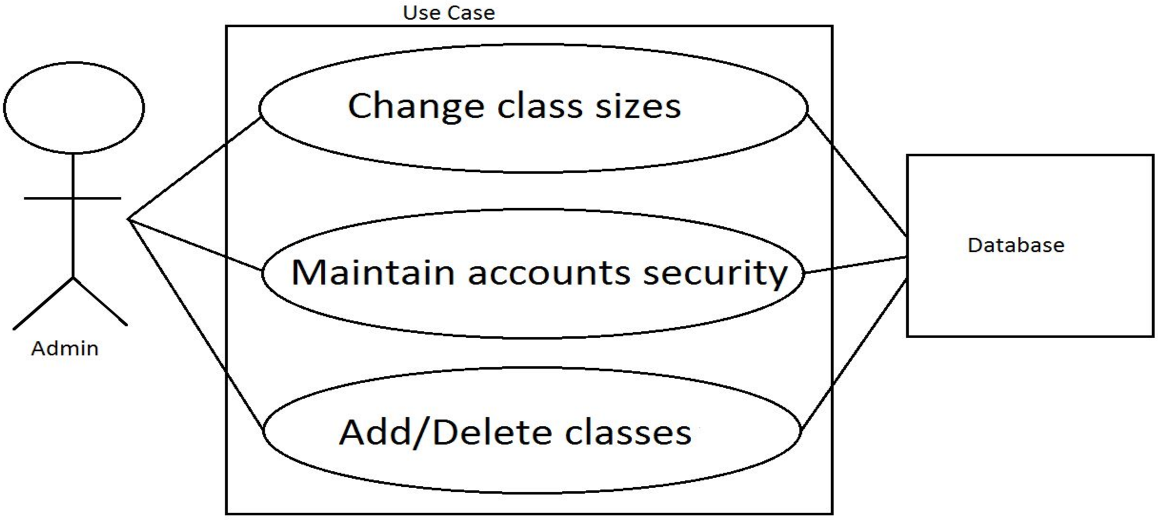
To further secure the system the site must have at least one **backup**, the backup should idealy be off-site and automated for added security

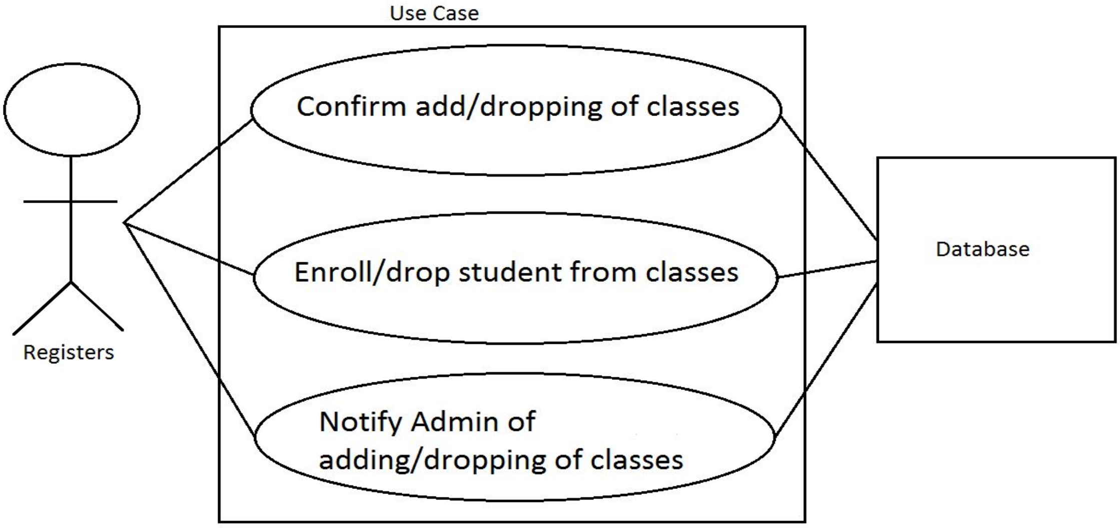
The site should only show the user what the user can see, in other words a student should not be able to access areas that are used by the registar or admins ect.

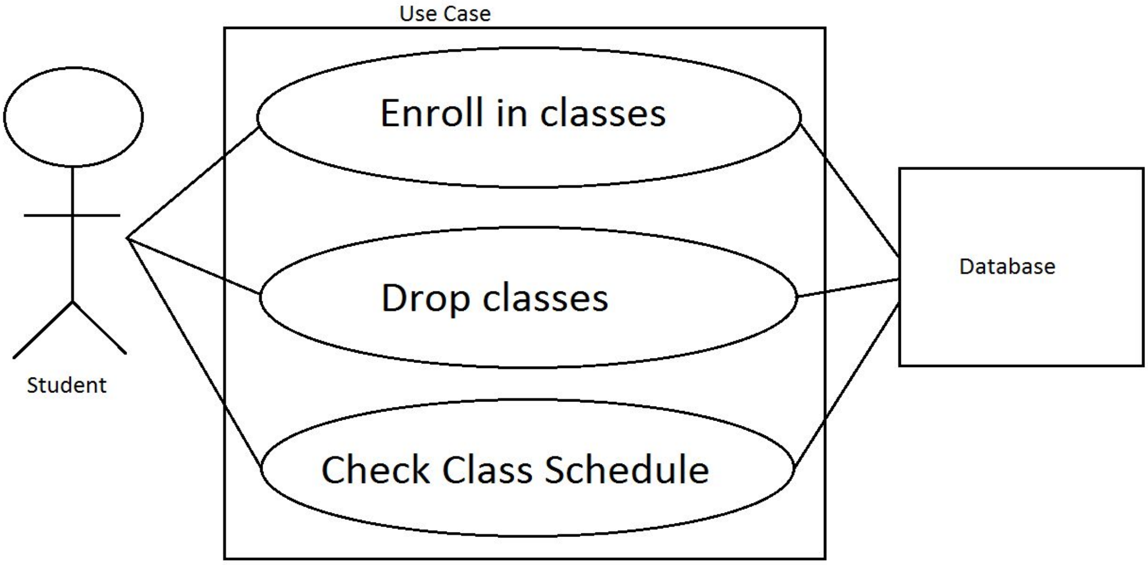
The site should use SSL to prevent private information from leaking and further security

The site should be able to be updated regularly, old data(graduated sudents or retired registars) should be regularly removed from the system. Scripts and other technologies should be updated to fit newer versions. Data should also be monitered to check for unathorized changes, additions, of removals. This would require that the client hire a a team to moniter the systems security.

Scenario Diagrams


Use Case
Admin
Change class sizes
Maintain accounts security
Add/Delete classes
Database


Use Case
Registers
Confirm add/dropping of classes
Enroll/drop student from classes
Notify Admin of adding/dropping of classes
Database


Use Case
Student
Enroll in classes
Drop classes
Check Class Schedule
Database