Course Project Report

SENIOR CAPSTONE

College Bazaar

Alissa Barnard|Trevor Drozd|Spenser Grzadzinski|James Moir|
Patrick Isho|Kelly Weber

..

Table of Contents

I. Project Description

A. Problem statement

College students have items and services to offer for sale, while others are always looking for items and services to purchase. Other college students are looking for roommates, advice, and/or class help. Our application will be the platform that brings college students together so they can accomplish all of these goals.

B. Objectives

College students often have various items they would like to sell (such as old textbooks, class supplies, etc), and also can offer many services (such as tutoring) to other classmates within their community. Our objective is to create a website where college students can register to and subscribe to their college's portion of the website, and can offer up these goods and services for sale to members of their college. Colleges will register with the website and pay a monthly fee to allow their students access to the system. This will make it easier for their students to connect with each other and adds to their sense of community within the college.

Students shall register with their college email address to join, for purposes of verification, and the system will only accept valid school email addresses. The website's purpose is for students to be able to sell items, purchase items, and to provide and accept services within their college's community. Colleges will be able to register and appoint administrators who will have the authority to approve or deny student requests, whitelist safe domains, blacklist malicious domains, and promote/demote school user accounts (including blocking). Each college shall have group forums of varying subjects that the students and admins are able to post to. Website admins have the authority to approve or deny school requests. The System shall allow payments for goods and services to be made via paypal, and in a secure manner.

C. Target Environment

Currently our target environment are the colleges and their students, located in different colleges across the State of Michigan.

D. Technologies & tools

Programing languages: HTML5, CSS, JAVASCRIPT, JAVA, PHP, MYSQL.
Design software: Photoshop
Coding framework: Ionic, Jhipster
Testing Software: WAMP
Hosting: Google Cloud Platform
Development: Ubuntu (live), Cloud 9 (work), github (work)
Security: Cloudbric
Hardware: MySql server - Google Cloud Platform; Apache2 Webserver - Google Cloud Platform
Group Communication: Google Docs; Slack, Email

E. Risk Management and Mitigation

Risks

1. All of us work full - time and have many other classes. Finding 800 to 1,000 man hours to create a complex project will be challenging.

2. We are each expanding our areas of knowledge into more complex developing and coding areas in a short amount of time.

3. Our schedules all collide and it is difficult to find times when we are all available at the same time.

4. Many group members are not very strong coders.

Mitigation

1. We all have different skills and we will utilize our different strengths to pull this project together in time.

2. We will each document and post research into github so we can share what we are learning to help each other.

3. We will utilize Slack, Google Docs, and plan ahead to make sure that we are communicating effectively in a timely manner.

4. Help others become more familiar with the different languages being used.

II. Requirements

A. Functional requirements

Identifier	Functional Requirement
REQ1	The system shall display a registration page.
REQ2	The system shall provide the ability for the user to become a subscriber to the website after registering.
REQ3	The system should detect emails with the incorrect format required for this website.
REQ4	The system shall display a login area with an option to change username/password.
REQ5	The system shall provide the user the ability to post and remove products/services for sale.
REQ6	The system shall provide the user the ability to post and remove into forums of various subjects.
REQ7	The system shall provide the user the ability to send private email to other users to arrange to purchase products/services listed for sale.
REQ8	The system shall successfully store all data from the website into the database.

B. Non-functional requirements

Identifier	Non-Functional Requirement
REQ9	The system shall secure user's passwords stored in the database with encryption.
REQ10	The system shall provide DDOS protection.
REQ11	The system shall provide website analytics.
REQ12	The system shall have a work environment separate from a live environment.
REQ13	The system shall have terms of use.

III. Sprint Schedule

A. Sprint 1 (9/6-9/13)

- Project requirements
- Project plan
- Tool chain, language, configuration management, server setup, development environment, IDE, framework
- Sprint planning

B. Sprint 2 (9/15-9/27)

- GCP VMs running
- Create database
- Website design and structure
- User account creation, registration, profiles, email verification
- Develop user stories and Use cases
- UI/UX
- Layout plan on framework

- QA tests for user authentication system

C. Sprint 3 (9/28-10/13)

- GCP live environment go live
- Placeholder page uploaded
- DNS configuration for security platform
- Integration of security and analytics tools
- System security, data protection
- Class or Entity Relationship Diagram
- Website design
- Database tables

D. Sprint 4 (10/14-10/25)

- GCP SQL integration into Apache server
- Penetration testing and system hardening on public facing IPs
- Website design
- Database tables
- Mobile app design and structure
- Unit and System Testing, and test data selection

E. Sprint 5 (10/26-11/8)

- GCP script implementation for making updates in the work environment live in a more user friendly manner
- Connecting application and website to database
- Mobile app design and structure
- Secure a domain name and make the website reachable by itv
- Algorithms Design and Development

F. Sprint 6 (11/9-11/22)

- Mobile app design and security
- System fault testing
- Quality Assurance

G. Sprint 7 (11/23-12/6)

- Final adjustments to website
- Final adjustments to database
- System fault testing
- Quality Assurance

H. Sprint 8 (12/7-12/13)

- System deployment and finalization

IV. User Stories

User Story #1. As a user I can register on the website to become a subscriber, and I can trust that my registration information shall be stored in a database.
User Story #2. As a user I can login to the website.
User Story #3. As a user I shall have the ability to post and remove products, services, and comments in forums to the website.
User Story #4. As a user I shall have the ability to send and receive emails to and from other users.
User Story #5. As a user I should have security of my password stored into the database.
User Story #6. As a user I should have the ability to access a live environment that is not affected by the work environment.
User Story #7. As a user I should have access to the legal section and terms of use.

V. User Stories Descriptions

User Story #1
As a user I can register on the website to become a subscriber, and I can trust that my registration information shall be stored in a database.
create a registration template with email, username, password, first name, last name, college choice.
create the portion of the database to store the data from the registration.
create the code that will take the registration information from the website and store it into the database.
create the code that can verify the format of emails match.
create message to be displayed for the customer such as: “Must enter a valid email, i.e. johnsmith@oakland.edu”.
Confirmation:
Success:
User registers with with email, username, password, first name, last name, college choice and becomes a subscriber.
Registration information successfully stores in database.
If a user enters an email with the wrong format, the system replies with: “Must enter a valid email, i.e. johnsmith@oakland.edu.”
Failure:
User is unable to register.
User’s registration data does not store into database.
User is able to register with an email address with the incorrect format.

User Story #2
As a user I can login to the website.
create the code to display a login area using username and password.
create the code that can verify the data entered as login information matches what is stored in the database..
create message to be displayed for the customer such as: “Username or Password is not correct”.
create the code to display the link “change username/password”.
create the code to send email with directions on changing username/password when link is selected.
create the code that will delete the stored username and password and replaces it with the new username and password.
Confirmation:
Success:
User logs in with with username and password and is given access to website.
Log in information successfully matched in the database..
If a user enters incorrect login information, the system replies with: “Username or Password is not correct”.
If user forgets login the user can click on link for change username/password, and they are emailed directions on how to accomplish this.
The user follows the directions to change username/password, and the system deletes what is stored in the database and replaces it with the new username/password.
Failure:
User is unable to login ever.
The matching process for the login information does not work properly..

User is able to register with a bad username and password.
The message “Username or Password is not correct” does not display at the appropriate times.
The system never sends an email with directions on how to change username/password.
The system doesn’t update itself with the new username/password.

User Story #3
As a user I shall have the ability to post and remove products, services, and comments in forums to the website.
create the code that provides the user the ability to post and remove products to different product pages.
create the code that provides the user the ability to post and remove services to different services pages.
create the code that provides the user the ability to post and remove comments to different forum pages.
Confirmation:
Success:
The user has the ability to successfully post items to the pages designated for items.
The user has the ability to successfully post services to the pages designated for services.
The user has the ability to successfully post comments to the pages designated for forums.
Failure:
System fails to recognize what the user is posting, and allows items to be posted to the wrong pages.

The user is unable to post.

User Story #4
As a user I shall have the ability to send and receive emails to and from other users.
create the code for users to send and receive emails between other registered users.
Confirmation:
Success:
User is able to send and receive emails to the email registered on the site.
Failure:
User sends email and it is never received by the user it was sent to.

User Story #5
As a user I should have security of my password stored into the database.
create hash encryption for the user's password that is stored into the database to ensure it is kept confidential.
create code when creating database to prevent DDOS attacks.
Confirmation:
Success:
Passwords turn into hash when stored into the database.
Database is set up to prevent DDOS attacks.
Failure:
Hash encryption fails and the passwords are stored in plain text.
DDOS attack is possible.

User Story #6
As a user I should have the ability to access a live environment that is not affected by the work environment.
setup the live environment.
setup the work environment.
Confirmation:
Success:
The work environment and live environment are setup to be separate and changes in the work environment doesn't affect the live environment.
Failure:
The changes made in the work environment affect the live environment and the users are able to see this.

User Story #7
As a user I should have access to the legal section and terms of use.
research legal sections and terms of use commonly used on websites.
create the legal section and terms of use on a page on the website.
Confirmation:
Success:
The legal section and terms of use is well written and displayed on the website.
Failure:

The legal section and terms of use does not display properly or is poorly written.

VI. Use Case

USE CASE UC-1:	**REGISTERING**
Related Reqs:	**User Story 1**
Initial Actor:	**User**
Actor's Goal:	**To register for the website**
Participating Actors:	**Database, Website**
Preconditions:	**The user has visited the website. The database is ready to store the data. The system displays a registration page.**
Postconditions:	**The user registers and the data is stored in the database.**
Flow of Events	**1. The user visits the website. The user locates the registration page. 2. The user enter data and completes the registration page. 3. The user's data is stored into the database.**

USE CASE UC-2:	**LOGGING IN**
Related Reqs:	**User Story 2**
Initial Actor:	**User**
Actor's Goal:	**To login the website OR to change username/password**
Participating Actors:	**Database, Website**

Preconditions:	**The user has visited the website. The database is ready to store the data. The system displays a login area.**
Postconditions:	**The user logs in and is provided access to the website.**
Flow of Events	**1. The user visits the website. The user locates the login area.** **2. The user enters login data and access is granted.** **OR** **3. The user sees the message "Username/Password incorrect".** **4. The user clicks on the link that says "Change Username/Password."** **5. The system sends an email to the user with directions on changing Username/Password.** **6. The user follows the directions.** **7. The system deletes old username/password from database and updates it with the new username/password.** **8. The user enters in new login data.** **9. The user is given access to the website.**

USE CASE UC-3:	**POSTING AND REMOVING DATA**
Related Reqs:	**User Story 3**
Initial Actor:	**User**
Actor's Goal:	**To post or remove data to the website.**
Participating Actors:	**Database, Website**
Preconditions:	**The user is registered. The user is logged in.**
Postconditions:	**The user posts or removes data to the website.**
Flow of Events	**1. The system recognizes the user as logged in.** **2. The user locates the section of the website for the data they wish to post or remove.** **3. The user posts or removes the data to the website.**

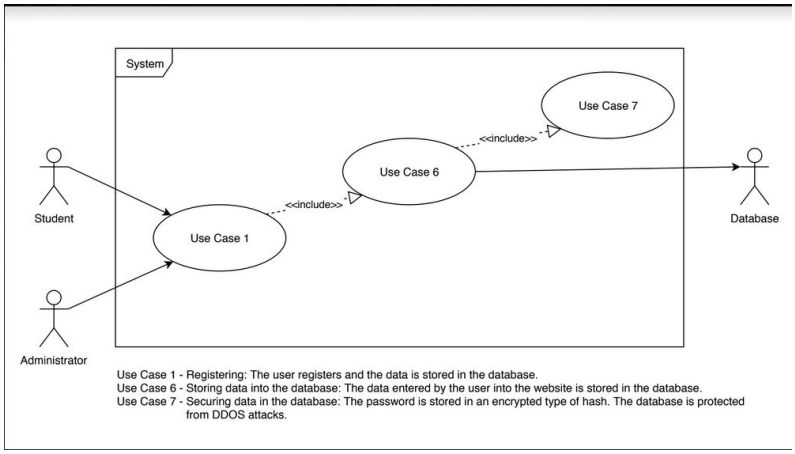
USE CASE UC-4:	SENDING EMAILS
Related Reqs:	User Story 4
Initial Actor:	User
Actor's Goal:	To send email to between users registered with the system.
Participating Actors:	Database, Website
Preconditions:	The user is registered. The user is logged in.
Postconditions:	The emails are sent between the users of the website.
Flow of Events	1. The system recognizes the user is logged in. 2. The system allows the user to transcribe and send an email to another user.

USE CASE UC-5:	SECURING DATA IN DATABASE
Related Reqs:	User Story 5
Initial Actor:	DATABASE
Actor's Goal:	The system shall secure the password with a type of encryption.
Participating Actors:	Administrator
Preconditions:	The database is storing the password of the users. The database is secured for DDOS attacks.
Postconditions:	The password is stored in an encrypted type of hash. The database is protected from DDOS attacks.
Flow of Events	1.The user registers on the website and enters a password. 2. The password is turned into hash before it is stored into the database. 3. The password is stored into the database in hash form. 4. A hacker tries a DDOS attack. 4. The database prevents the DDOS attacks.

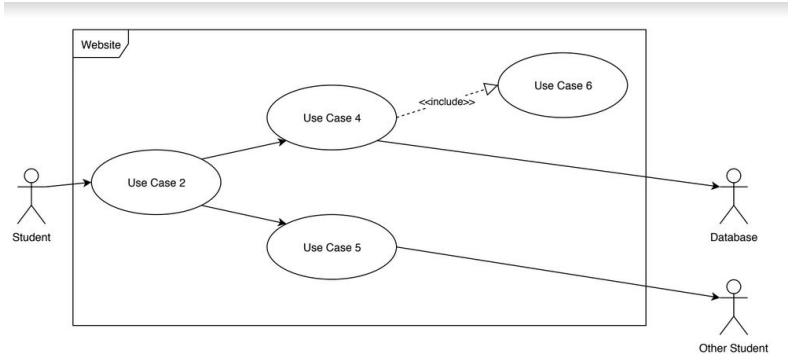
USE CASE UC-6:	SEPARATING THE WORK ENVIRONMENT AND LIVE ENVIRONMENT
Related Reqs:	User Story 6
Initial Actor:	ADMINISTRATOR
Actor's Goal:	The administrator shall build the system in a work environment separate from the live environment.
Participating Actors:	Website
Preconditions:	The administrator is working in a separate work environment when making updates or changes to the website.
Postconditions:	The changes made by the administrator are not visible to users until they are completed and uploaded to the live environment.
Flow of Events	1.The administrator is making updates/changes to the website. 2. The administrator is making these only in the working environment. 3. When the changes are completed, the administrator uploads them to the live environment.

USE CASE UC-7:	PROVIDING A LEGAL SECTION
Related Reqs:	User Story 7
Initial Actor:	USER
Actor's Goal:	The user shall have access to a legal section on the website.
Participating Actors:	Website
Preconditions:	A legal section is designed into the layout of the website.
Postconditions:	The user has the ability to view a legal section that is displayed on the website.
Flow of Events	1.The user starts the registration process. 2. During the process the user is asked to view and accept the terms of use in the legal section. 3. The user view it and accepts it. 4. Registration can be completed.

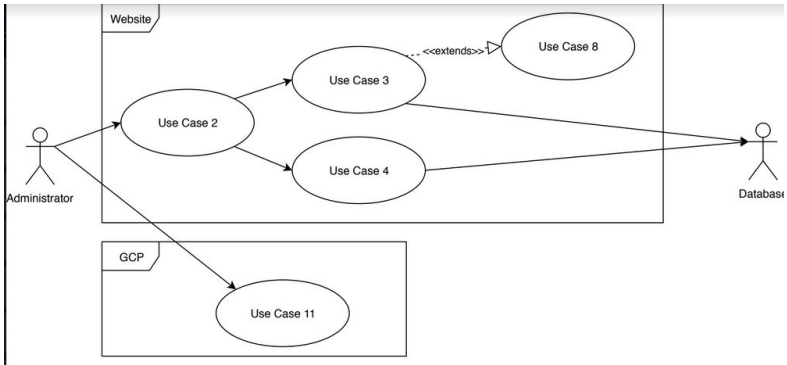
VII. Use Case Diagrams



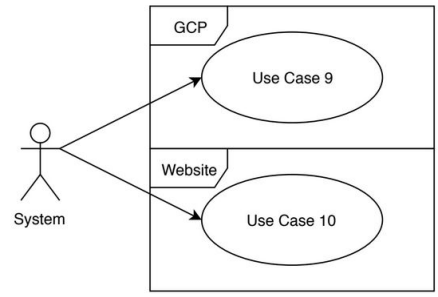
Use Case 1 - Registering: The user registers and the data is stored in the database.
Use Case 6 - Storing data into the database: The data entered by the user into the website is stored in the database.
Use Case 7 - Securing data in the database: The password is stored in an encrypted type of hash. The database is protected from DDOS attacks.



Use Case 2 - Logging in: The user logs in and is provided access to the website.
Use Case 4 - Posting and removing data: The user posts or removes data to the website.
Use Case 5 - Sending emails: The emails are sent between the users of the website.
Use Case 6 - Storing data into the database: The data entered by the user into the website is stored in the database.

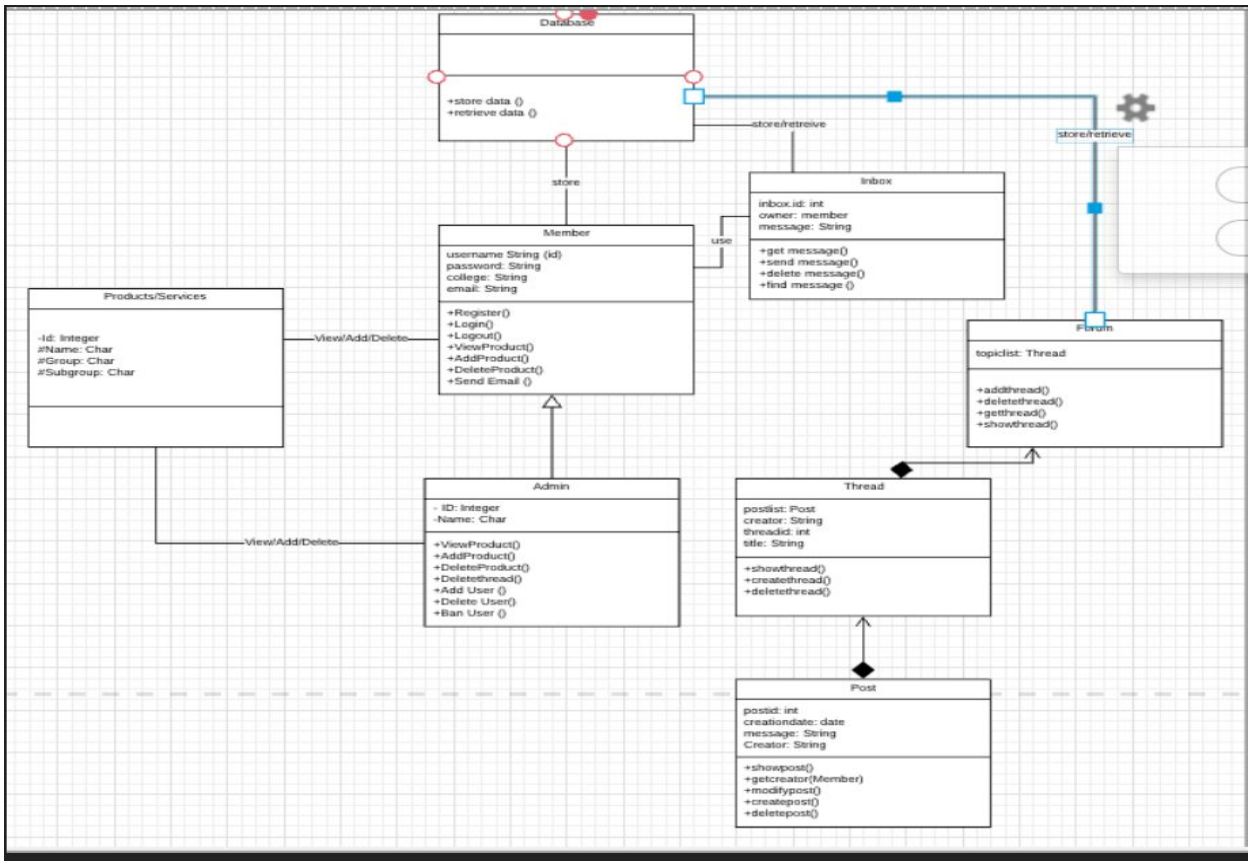


Use Case 2 - Logging in: The user logs in and is provided access to the website.
Use Case 3 - Accessing administrator privileges: The system recognizes the administrator by the username and password and grants the administrator privileges assigned to that administrator.
Use Case 4 - Posting and removing data: The user posts or removes data to the website.
Use Case 8 - Reviewing website analytics: The administrator is able to review the website analytics.
Use Case 11 - Accessing the Google Cloud Platform VMs via console or SSL connection: The GCP Vm's are accessed via console or SSL Connection.



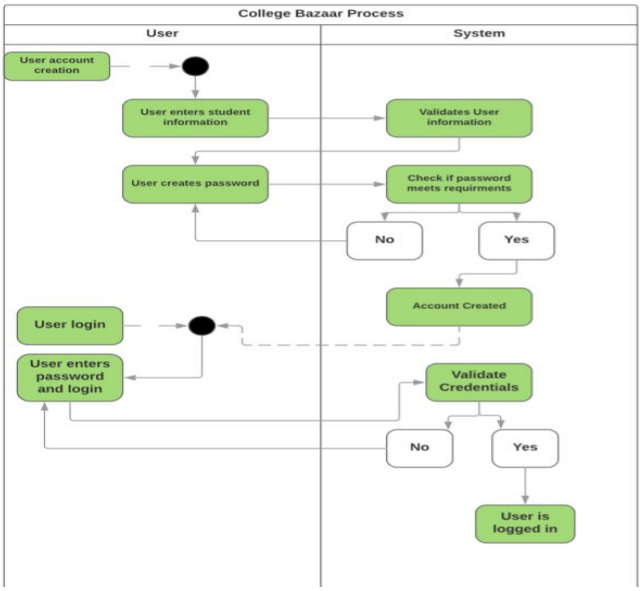
Use Case 9 - Separating work environment from live environment: The changes made by the administrator are not visible to users until they are completed and uploaded to the live environment.
Use Case 10 - Providing a legal section: The administrator provides a legal section that is displayed on the website for the users to view.

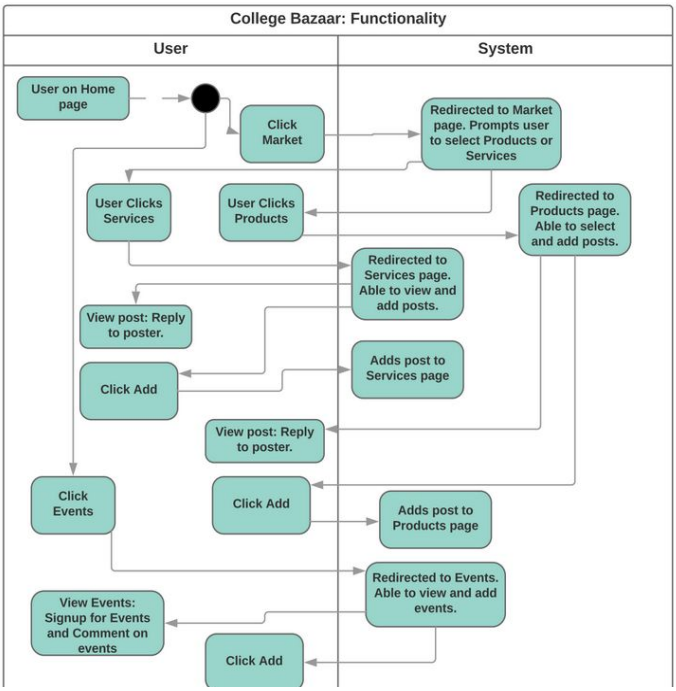
VIII. Class Diagram



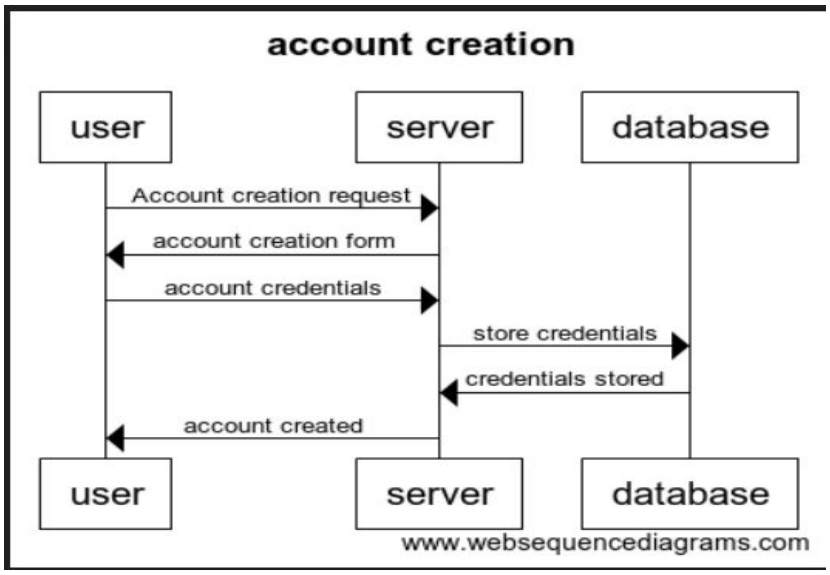
IX. Activity Diagram

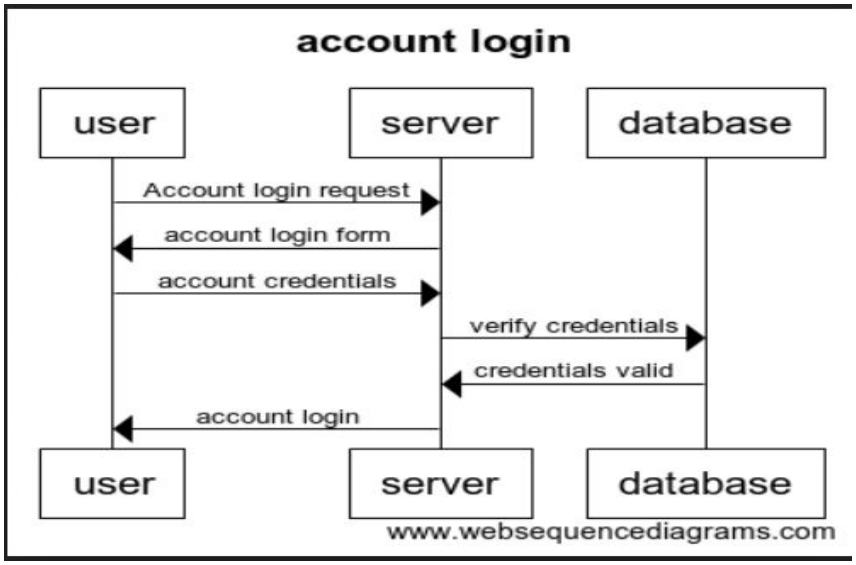
ACTIVITY DIAGRAM

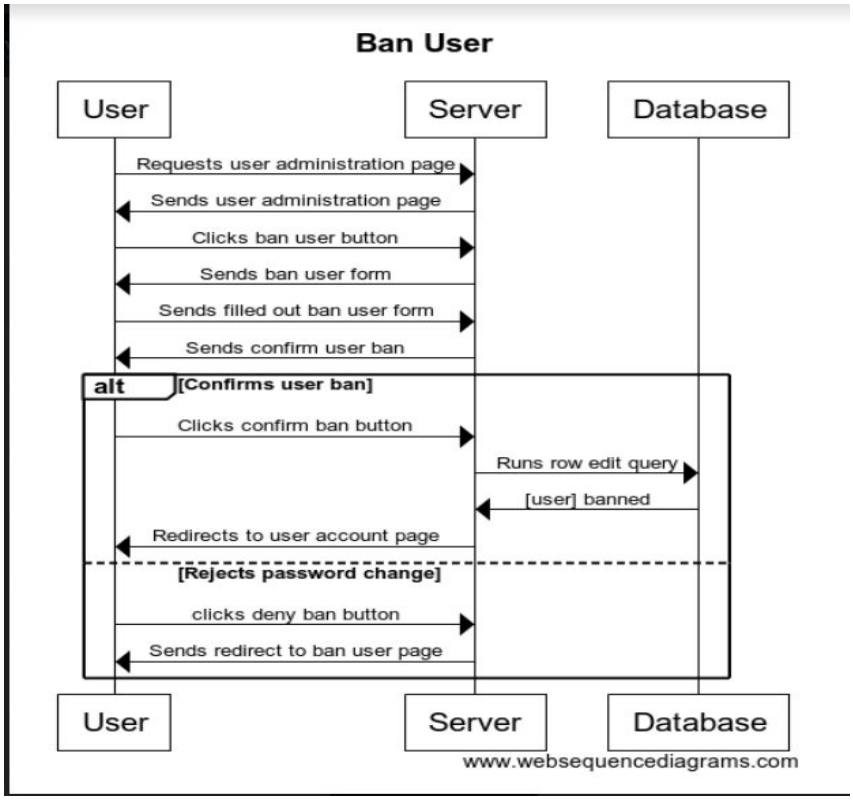


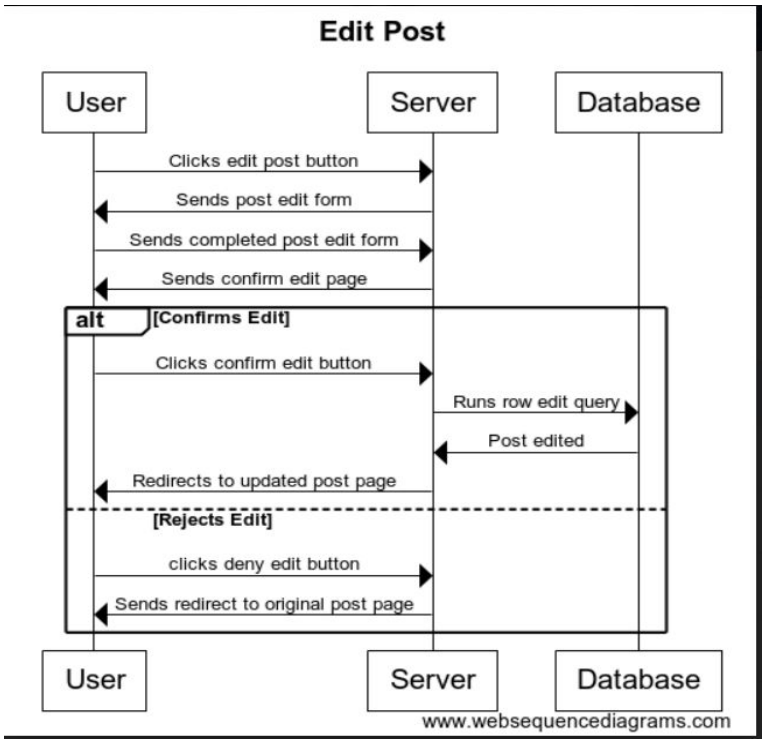


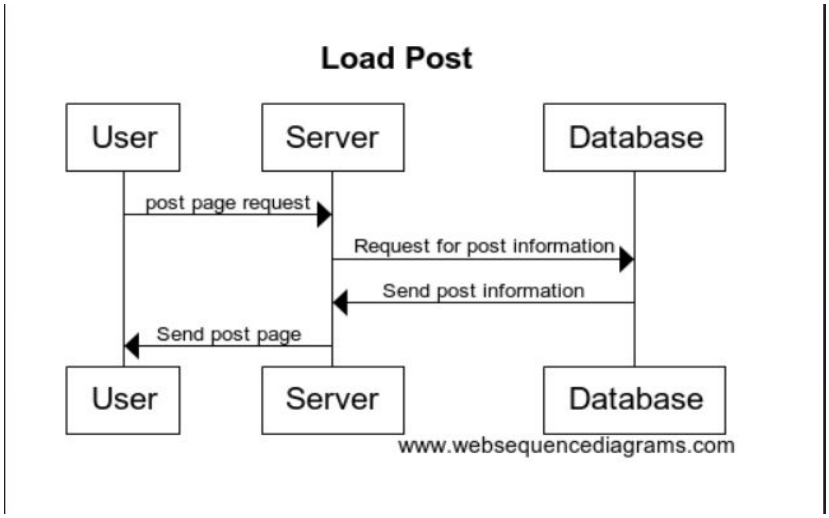
X. Sequence Diagram

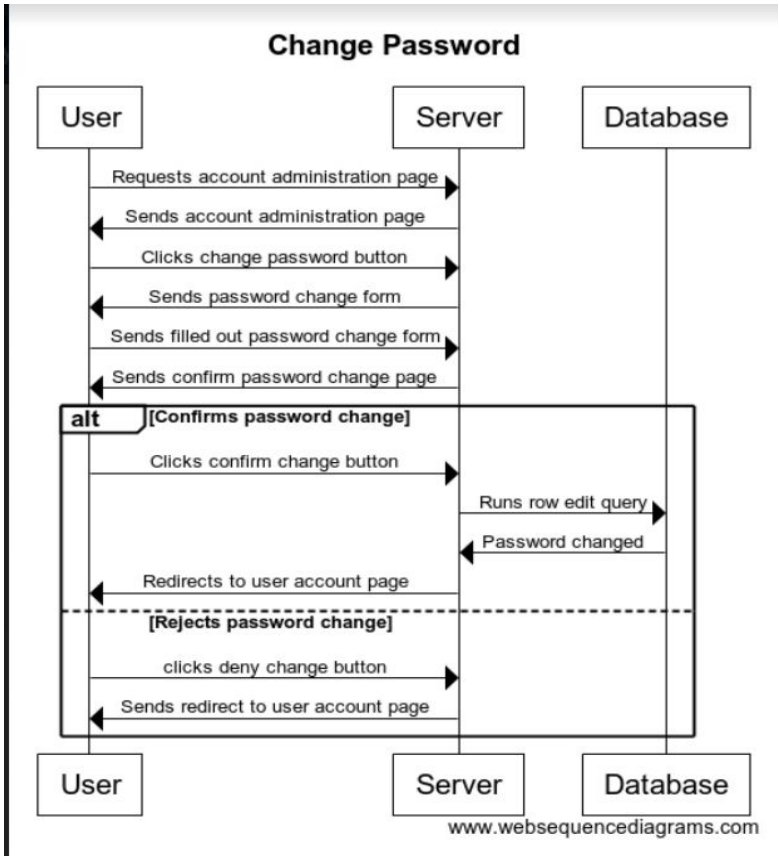


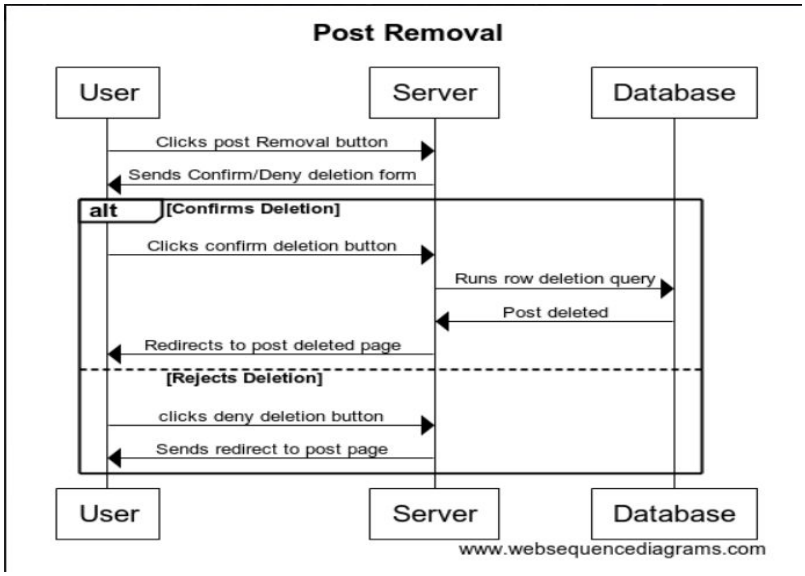


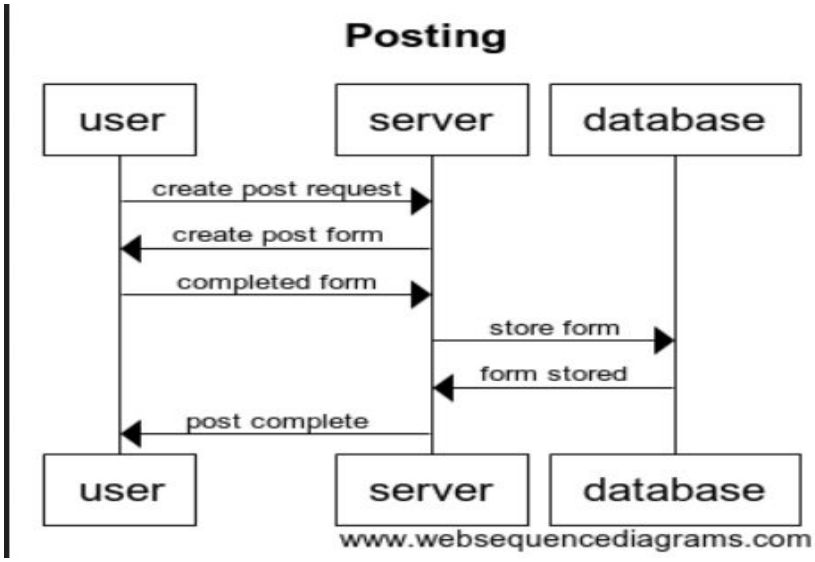

Ban User
User
Server
Database
Requests user administration page
Sends user administration page
Clicks ban user button
Sends ban user form
Sends filled out ban user form
Sends confirm user ban
alt
[Confirms user ban]
Clicks confirm ban button
Runs row edit query
[user] banned
Redirects to user account page
[Rejects password change]
clicks deny ban button
Sends redirect to ban user page
User
Server
Database
www.websequencediagrams.com

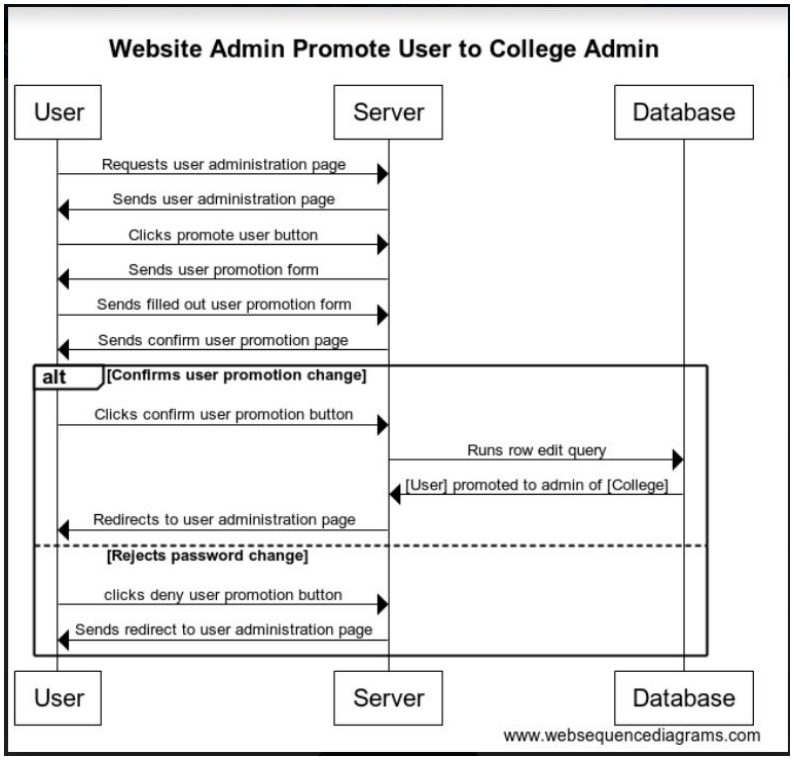

Edit Post
User
Server
Database
Clicks edit post button
Sends post edit form
Sends completed post edit form
Sends confirm edit page
alt
[Confirms Edit]
Clicks confirm edit button
Runs row edit query
Post edited
Redirects to updated post page
[Rejects Edit]
clicks deny edit button
Sends redirect to original post page
User
Server
Database
www.websequencediagrams.com

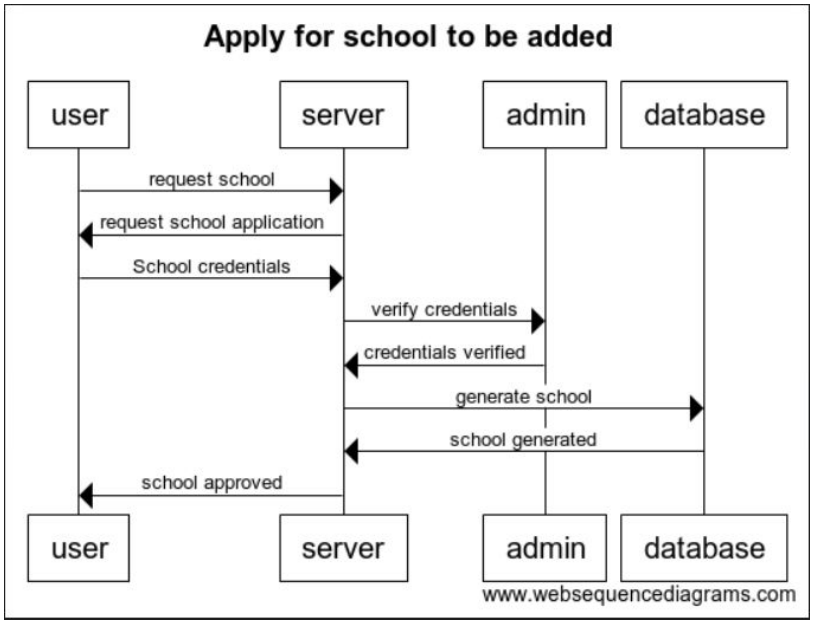

Load Post
User
Server
Database
post page request
Request for post information
Send post information
Send post page
User
Server
Database
www.websequencediagrams.com

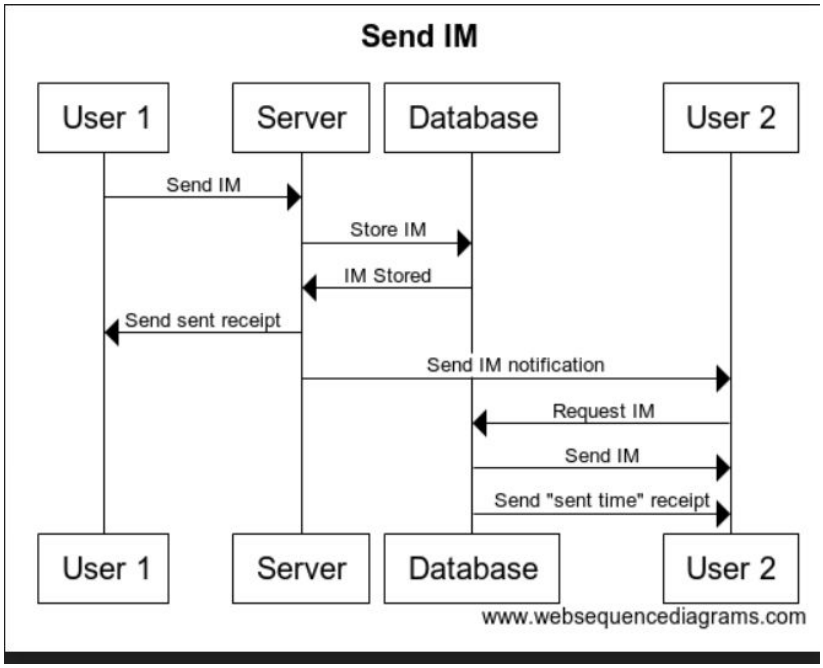

Change Password
User
Server
Database
Requests account administration page
Sends account administration page
Clicks change password button
Sends password change form
Sends filled out password change form
Sends confirm password change page
alt
[Confirms password change]
Clicks confirm change button
Runs row edit query
Password changed
Redirects to user account page
[Rejects password change]
clicks deny change button
Sends redirect to user account page
User
Server
Database
www.websequencediagrams.com

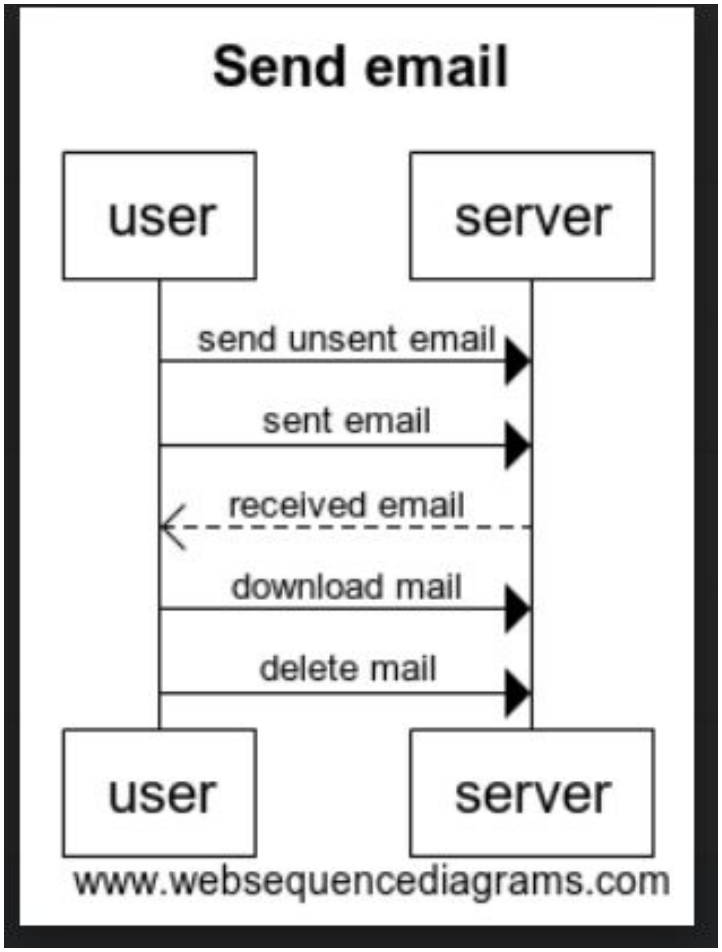

Post Removal
User
Server
Database
Clicks post Removal button
Sends Confirm/Deny deletion form
alt
[Confirms Deletion]
Clicks confirm deletion button
Runs row deletion query
Post deleted
Redirects to post deleted page
[Rejects Deletion]
clicks deny deletion button
Sends redirect to post page
User
Server
Database
www.websequencediagrams.com

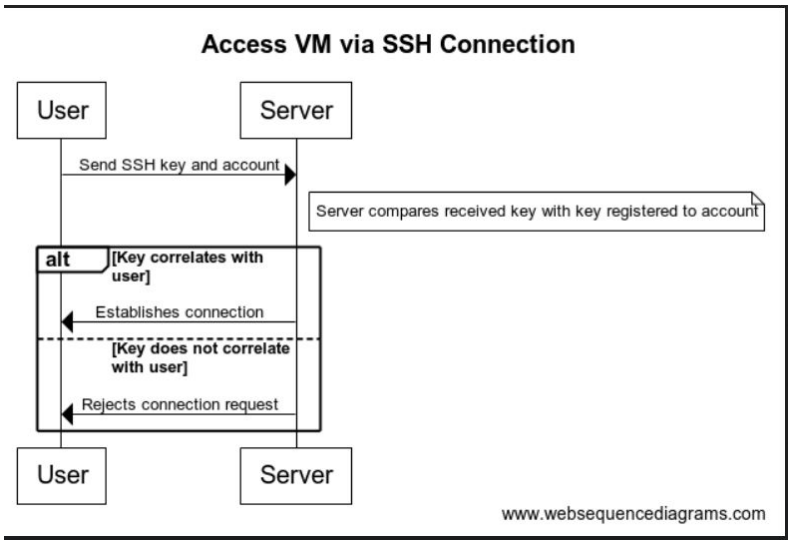

Posting
user
server
database
create post request
create post form
completed form
store form
form stored
post complete
user
server
database
www.websequencediagrams.com

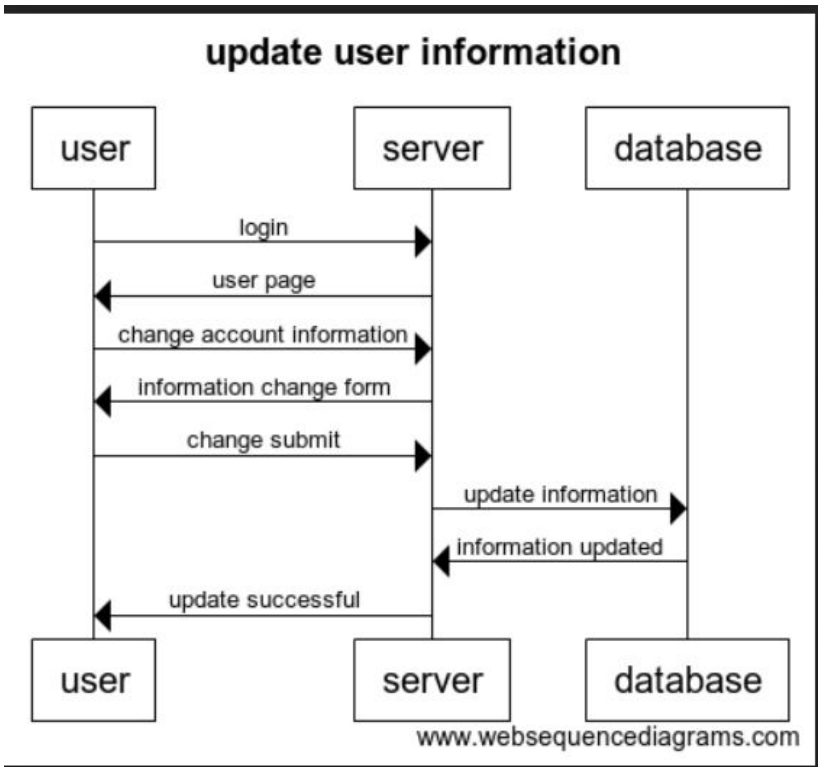

Website Admin Promote User to College Admin
User
Server
Database
Requests user administration page
Sends user administration page
Clicks promote user button
Sends user promotion form
Sends filled out user promotion form
Sends confirm user promotion page
alt
[Confirms user promotion change]
Clicks confirm user promotion button
Runs row edit query
[User] promoted to admin of [College]
Redirects to user administration page
[Rejects password change]
clicks deny user promotion button
Sends redirect to user administration page
User
Server
Database
www.websequencediagrams.com

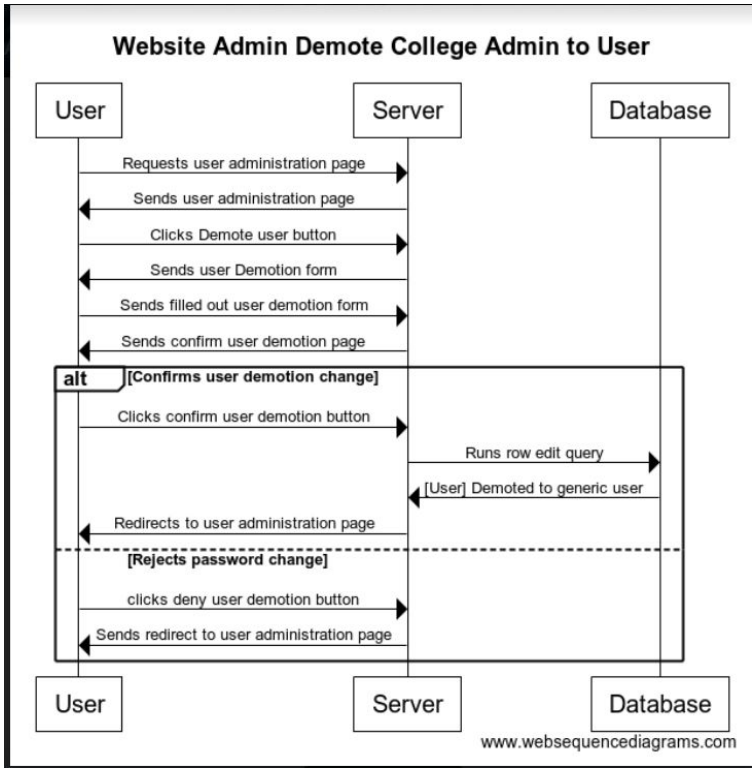

Apply for school to be added
user
server
admin
database
request school
request school application
School credentials
verify credentials
credentials verified
generate school
school generated
school approved
user
server
admin
database
www.websequencediagrams.com


Send IM
User 1
Server
Database
User 2
Send IM
Store IM
IM Stored
Send sent receipt
Send IM notification
Request IM
Send IM
Send "sent time" receipt
User 1
Server
Database
User 2
www.websequencediagrams.com


Send email
user
server
send unsent email
sent email
received email
download mail
delete mail
user
server
www.websequencediagrams.com


Access VM via SSH Connection
User
Server
Send SSH key and account
Server compares received key with key registered to account
alt
[Key correlates with user]
Establishes connection
[Key does not correlate with user]
Rejects connection request
User
Server
www.websequencediagrams.com


update user information
user
server
database
login
user page
change account information
information change form
change submit
update information
information updated
update successful
user
server
database
www.websequencediagrams.com



XI. Microsoft Project

	WBS	Task Name	Duration	Start	Finish	Predecessors	Successors
1	**1**	**Project**	**69 days**	**Mon 9/10/18**	**Thu 12/13/18**		
2	1.1	Project Start	0 days	Mon 9/10/18	Mon 9/10/18		4
3	**1.2**	**Group Meetings**	**68 days**	**Tue 9/11/18**	**Thu 12/13/18**		
4	1.2.1	Group Meetings 1	1 day	Tue 9/11/18	Tue 9/11/18	2	31,33,32,34,
5	1.2.2	Group Meetings 2	1 day	Thu 9/13/18	Thu 9/13/18		
6	1.2.3	Group Meetings 3	1 day	Tue 9/18/18	Tue 9/18/18		
7	1.2.4	Group Meetings 4	1 day	Thu 9/20/18	Thu 9/20/18		
8	1.2.5	Group Meetings 5	1 day	Tue 9/25/18	Tue 9/25/18		
9	1.2.6	Group Meetings 6	1 day	Thu 9/27/18	Thu 9/27/18		
10	1.2.7	Group Meetings 7	1 day	Tue 10/2/18	Tue 10/2/18		
11	1.2.8	Group Meetings 8	1 day	Thu 10/4/18	Thu 10/4/18		
12	1.2.9	Group Meetings 9	1 day	Tue 10/9/18	Tue 10/9/18		
13	1.2.10	Group Meetings 10	1 day	Thu 10/11/18	Thu 10/11/18		
14	1.2.11	Group Meetings 11	1 day	Tue 10/16/18	Tue 10/16/18		
15	1.2.12	Group Meetings 12	1 day	Thu 10/18/18	Thu 10/18/18		
16	1.2.13	Group Meetings 13	1 day	Tue 10/23/18	Tue 10/23/18		
17	1.2.14	Group Meetings 14	1 day	Thu 10/25/18	Thu 10/25/18		
18	1.2.15	Group Meetings 15	1 day	Tue 10/30/18	Tue 10/30/18		

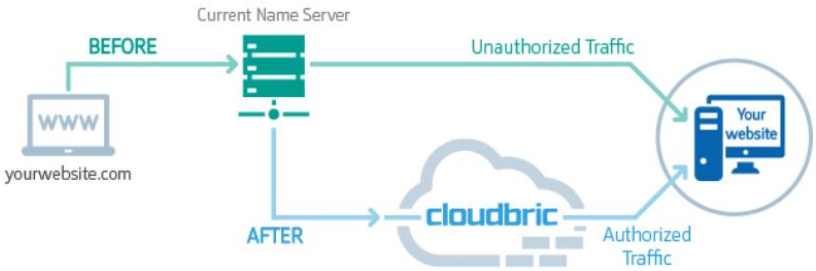
1.2.16	Group Meetings 16	1 day	Thu 11/1/18	Thu 11/1/18		
1.2.17	Group Meetings 17	1 day	Tue 11/6/18	Tue 11/6/18		
1.2.18	Group Meetings 18	1 day	Thu 11/8/18	Thu 11/8/18		
1.2.19	Group Meetings 19	1 day	Tue 11/13/18	Tue 11/13/18		
1.2.20	Group Meetings 20	1 day	Thu 11/15/18	Thu 11/15/18		
1.2.21	Group Meetings 21	1 day	Tue 11/20/18	Tue 11/20/18		
1.2.22	Group Meetings 22	1 day	Thu 11/22/18	Thu 11/22/18		
1.2.23	Group Meetings 23	1 day	Tue 11/27/18	Tue 11/27/18		
1.2.24	Group Meetings 24	1 day	Thu 11/29/18	Thu 11/29/18		
1.2.25	Group Meetings 25	1 day	Tue 12/4/18	Tue 12/4/18		
1.2.26	Group Meetings 26	1 day	Thu 12/6/18	Thu 12/6/18		
1.2.27	Group Meetings 27	1 day	Tue 12/11/18	Tue 12/11/18		
1.2.28	Group Meetings 28	1 day	Thu 12/13/18	Thu 12/13/18	4	
1.3	**Requirements**	**67 days**	**Wed 9/12/18**	**Thu 12/13/18**	**4**	
1.3.1	Project Scope	2 days	Wed 9/12/18	Thu 9/13/18	4	36
1.3.2	Design Requirements	2 days	Wed 9/12/18	Thu 9/13/18	4	36
3.2	Design Requirements	2 days	Wed 9/12/18	Thu 9/13/18	4	36
3.3	Technology	1 day	Thu 9/13/18	Thu 9/13/18	4	36
3.4	Complete Requirements	0 days	Thu 9/13/18	Thu 9/13/18	35,33,34	38,39
4	**Design**	**65 days**	**Fri 9/14/18**	**Thu 12/13/18**		
4.1	User Stories	1 day	Fri 9/14/18	Fri 9/14/18	36	40
4.2	Use Case	1 day	Fri 9/14/18	Fri 9/14/18	36	40,41
4.3	Diagrams	4 days	Mon 9/17/18	Thu 9/20/18	39,38	42
4.4	Project & Risk Management	6 days	Mon 9/17/18	Mon 9/24/18	39	42
4.5	Complete Design	0 days	Mon 9/24/18	Mon 9/24/18	41,40	44,51,50
5.5	**Code**	**59 days**	**Mon 9/24/18**	**Thu 12/13/18**		
5.1	Develop Registration & Login	3 days	Tue 9/25/18	Thu 9/27/18	42	45,46,47
5.2	Develop page layouts	47 days	Fri 9/28/18	Mon 12/3/18	44	48
5.3	Develop HTML & CSS Templates	47 days	Fri 9/28/18	Mon 12/3/18	44	48
5.4	Database Design & Implementatior	47 days	Fri 9/28/18	Mon 12/3/18	44	48

WBS	Task Name	Duration	Start	Finish	Predecessors	Successors
1.5.3	Develop HTML & CSS Templates	47 days	Fri 9/28/18	Mon 12/3/18	44	48
1.5.4	Database Design & Implementatior	47 days	Fri 9/28/18	Mon 12/3/18	44	48
1.5.5	Complete Code	0 days	Mon 12/3/18	Mon 12/3/18	47,45,46	52
1.6	**Testing**	**48 days**	**Thu 9/27/18**	**Mon 12/3/18**		
1.6.1	Validate Code in every Sprint	47 days	Fri 9/28/18	Mon 12/3/18	42	55
1.6.2	Testing & QA	48 days	Thu 9/27/18	Mon 12/3/18	42	55
1.6.3	User test website	4 days	Tue 12/4/18	Fri 12/7/18	48	53
1.6.4	Refine page layouts	1 day	Mon 12/10/18	Mon 12/10/18	52	55
1.7	**Deployment**	**4 days**	**Mon 12/10/18**	**Thu 12/13/18**		
1.7.1	Deploy	2 days	Tue 12/11/18	Wed 12/12/18	53,51,50	56
1.7.2	Present	1 day	Thu 12/13/18	Thu 12/13/18	55	

XII. Cloudbric

1) Comprehensive web security

2) Web application firewall
3) Ddos protection
4) SSL Certification



XIII. Backend Developments

1) Debian 9 VM has been installed and is in the process of configuration.
2) 20 GB dynamic memory and expandable
3) Located in google east servers