

Final Project for SW Engineering Class
CSC648-848 Spring 2019

Team 204

GatorDater

Tejas Vajjhala (tvajjhala@gmail.com): Team Lead, Github Master

Sanil Rijal (srijal0509@gmail.com): Front-end Lead

Thomas Zhang(t.zhang70@gmail.com): Front-end assist

Aaron Li (ali16@mail.sfsu.edu): Back-End Lead, Database Master

URL of Demo:

<http://35.203.131.184/>

12/19/2019

Product Summary

Name of Product: GatorDater

Current project URL: http://35.203.131.184/m3/Index_Page.html

All Major Committed Functions:

- Users shall be able to make an account
- Users shall be able to login
- Users shall be able to access the office hour scheduling page
- Users shall be able to reserve a time slot with a professor
- Users shall be able to cancel their time slot with a professor
- Users shall be able to search for professors
- Professors shall be able to cancel their appointment with a student
- Professors shall be able to block office hour days and time
- Professors shall be able to specify their free time slot
- Users shall be able to update their name
- Users shall be able to update their password
- Users shall be able to declare whether they are a student or professor
- Users shall be able to contact the developers for any issue
- Professors shall be able to view their appointments with students on calendar
- Professors shall be able to update their name
- Professors shall be able to update their password
- Professors shall be able to contact the developer for any issues

What is unique about GatorDater:

Currently, there is nothing like GatorDater that has been implemented for SFSU. For scheduling office hours, students need to either email the professor and try to come up with a time to meet, or sign up via a paper list posted outside the teachers room. This is very inconvenient as its difficult to sync up with professors and stay organized.

GatorDater allows users and professors to schedule appointments via an easy to use web interface and displays upcoming appointments, along with the ability to cancel, and reschedule them seamlessly.

Revision #: 2

Date: October 14th, 2019

SW Engineering CSC 648 / 848 Fall 2019

Milestone 1:

Use cases, High Level Requirements and Architecture

GatorDater

Team 204:

Tejas Vajjhala: Team Lead, Github Master

Sanil Rijal: Front-end Lead

Thomas Zhang: Front-end assist

Aaron Li: Back-End Lead, Database Master

Milestone	Version	Date Submitted For Review
Milestone 1	2	10/14/2019
Milestone 1	1	10/03/2019

Table of Contents

Table of Contents	2
Executive Summary	3
Main Use Cases	4
Diagram	6
Main Data Items and Entities	9
List of Functional Requirements	10
List of Non-Functional Requirements	11
Competitive Analysis	12
High-Level System Architecture and Technology	13
Checklist	14
Team	15

Executive Summary

GatorDater is an online application that facilitates the scheduling of office hours for students and professors. Previously, students would need to schedule with professors via a paper list located outside the professor's office room, or send an email to the professor which makes it hectic to synchronize with each other. This causes students to fall behind in class, and be unable to even ask for help as they are not able to get in touch with the professor face to face outside of class.

By using GatorDater, office hours are easily scheduled through the interface and professors do not need to wait in their office for a student that may or may not show up during an unscheduled period. By scheduling online, both students and professors have a clear calendar of their appointments and can ensure that they have an allocated time to discuss with each other. With the addition of the calendar, both students and professors are able to view and update their current status regarding time slots and availability. This allows for more fluidity and ensures less stress on both students and professors alike.

Main Use Cases

Use Cases : Student

- 1) A San Francisco State University student, Wesley, wants to schedule an appointment with his Math professor, Valerie for follow up questions. After reviewing his schedule, he comes to the conclusion that he is available after 3pm. He finds his professor's name and an associated time slot after 3pm. He then makes the reservation under Wesley.
- 2) A San Francisco State University Student Wesley, has reserved an appointment with programming Professor Shawn for tomorrow from 3:30 pm to 4:30 pm. But, Wesley realizes that he may not be able to attend the meeting because of his grad assignment that is due tomorrow. So, he logs in and cancel the appointment; making the professor office slot available for other students.

Use Cases: Professor

- 1) Professor Jake wants to view his upcoming office hours. He checks his calendar and sees his upcoming scheduled appointments. He realizes that he will not be available during one of the booked hours, and cancels the scheduled meeting. He then realizes this will be a recurrent issue and sets a block which makes students unable to book office hours during the specified hours/days.
- 2) A San Francisco State University professor, Jeff, just started teaching and wanted to host office hours. However, he did not want to attend school on his off days at his office, if no students were to show up. Therefore, he posted his time slot online for students to reserve. Once, a time slot is occupied, Jeff will arrive to school and host his office hours accordingly.
- 3) A San Francisco State University professor, Valerie, logs in and shall see the reservations for the upcoming office hours. She sees that one of her office hour's time slots at 5pm is reserved by Wesley. On the day of, Valerie waits for Wesley at her office to do her office hours.
- 4) A San Francisco State University Professor Jacob, logs in and see that a student has booked an appointment with him from 3:00 pm to 4:00 pm. But, the professor has an emergency and realizes that he will not be able to wait for the student. So, he cancels the appointment and sends the possible alternative office hours to the student.

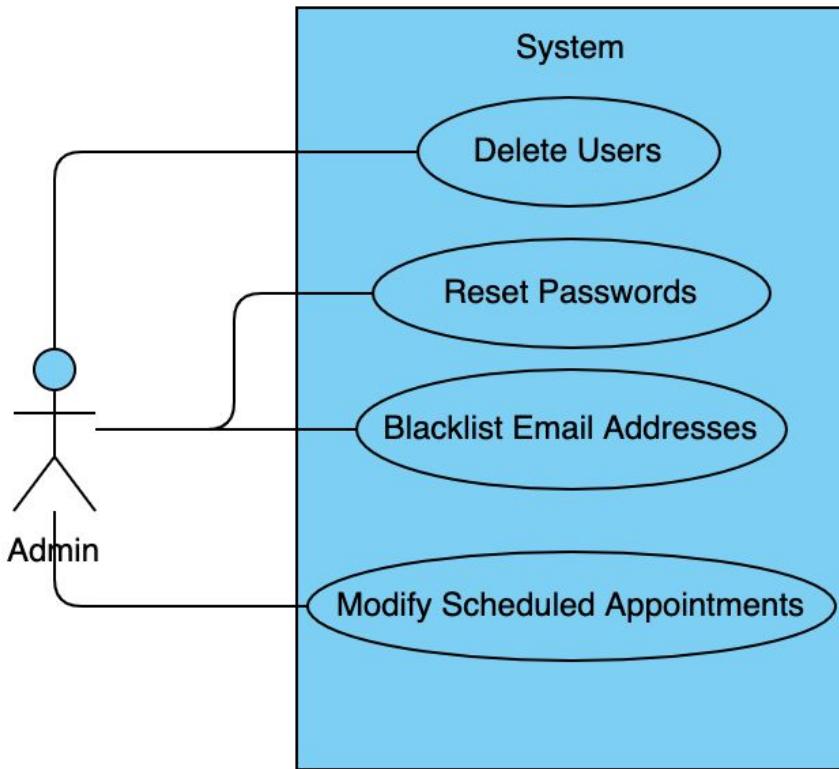
Use Cases: Unregistered User

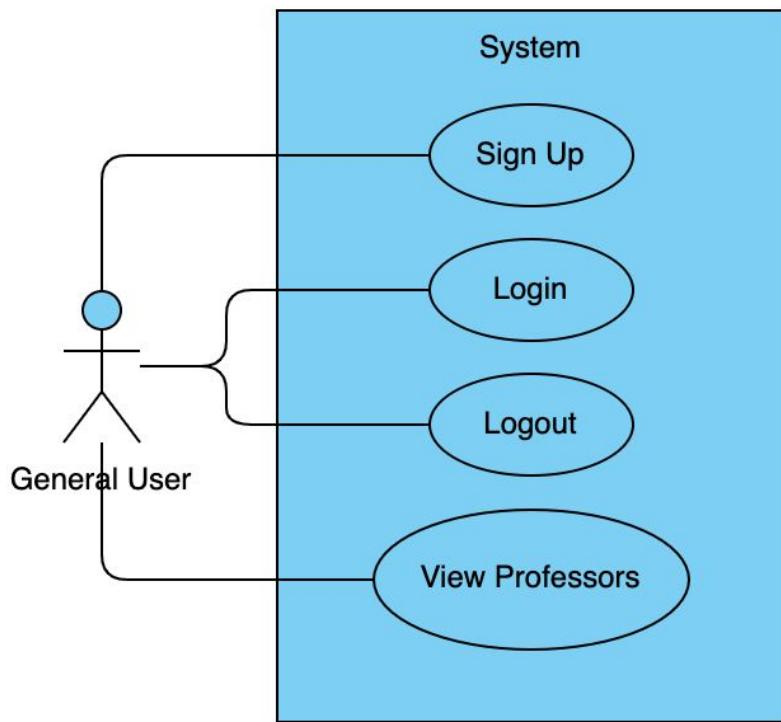
- 1) Jenny wants to schedule an office hour appointment with her professor. She has never used GatorDater before. She loads the website, signs up for the service, then logs in. She searches for her professor, and views the free slots for office hours. She then books a time slot with the professor and has it saved on her personal calendar.

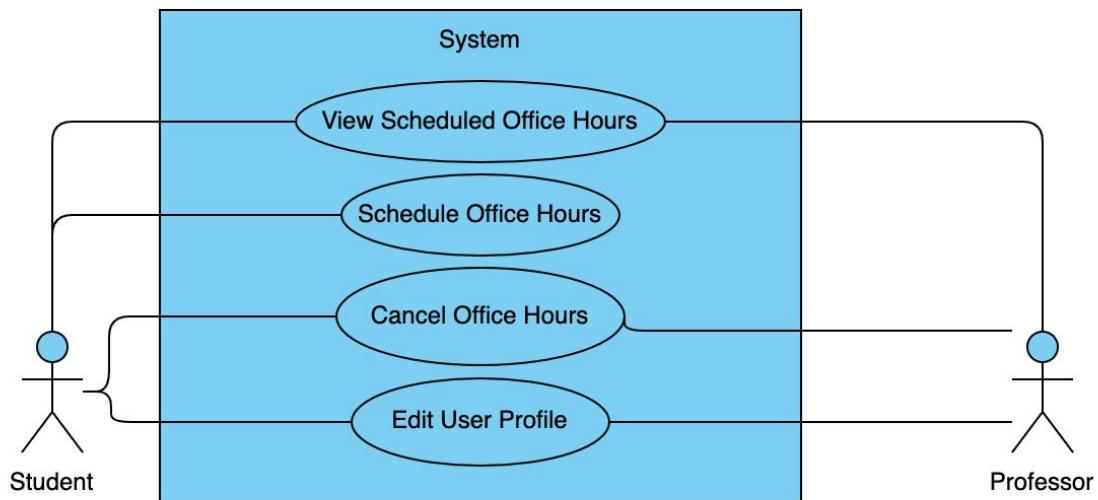
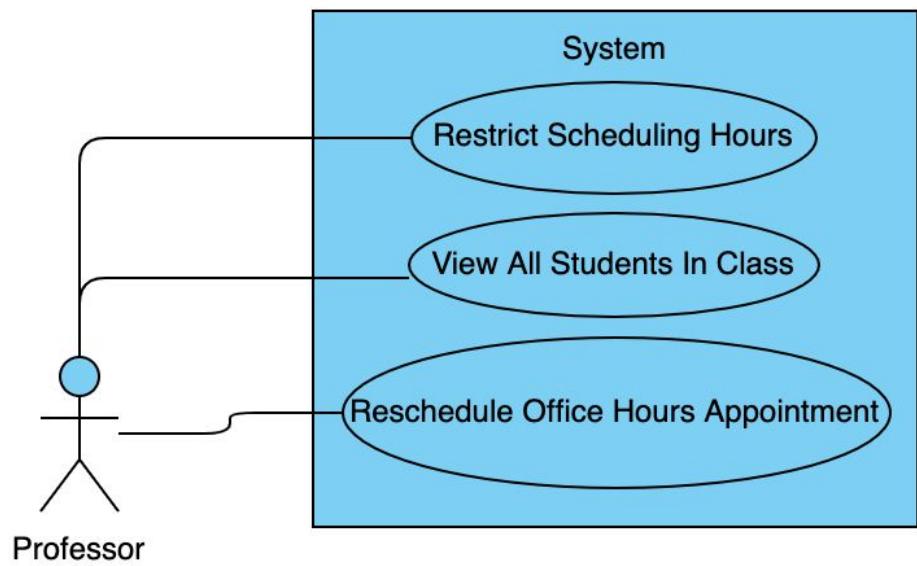
Use Cases: Admin

- 1) A user has been spamming professors and blocking all of their office hours. In order to free up these sections for other users, and prevent this user from blocking off hours, admin Joe deletes this user's account and blacklists his email address.

Diagrams:







List of main data items and entities

Users:

- 1) Registered User who has an account
- 2) Unregistered User who doesn't have an account

Student:

- 1) Registered User who can schedule office hours with a professor
- 2) Registered User who can cancel their appointment
- 3) Registered User who can view the calendar with their respected hours

Professor:

- 1) Registered User who can post and accept/decline hours
- 2) Registered User who can view the calendar with all the respected students hours

Admin:

- 1) Registered User who can access all data and content and modify the database
- 2) Registered User who can schedule office hours with professor
- 3) Registered User who can post and accept/decline hours

Application:

- 1) Calendar to display scheduled office hours associated with students and professors
- 2) User profiles for students and professors to provide a bio
- 3) Search bar to display multiple professors

Initial list of functional requirements

For Registered Users:

- 1) Users shall be able to make an account
- 2) Users shall be able to login
- 3) Users shall be able to access the office hour page
- 4) Users shall be able to cancel their scheduled office hours appointment
- 5) Users should be able to update their profile
- 6) Users shall be able to specify which class they belong to
- 7) Users shall be able to specify whether they are professor or student
- 8) Users shall be able to specify which major they are in
- 9) User shall be able to contact the creators for help

For Registered Users That Are Students:

- 1) Users shall be able to search for their professors
- 2) Users shall be able to reserve a time slot
- 3) Users shall be able to cancel their appointment
- 4) Users shall be able to schedule multiple hours
- 5) Users shall be able to monitor respected hours on the calendar
- 6) Users shall be able to have multiple appointments

For Registered Users That Are Professors:

- 1) Professor shall be able to post their available time slots.
- 2) Professor shall be able to restrict hours from being scheduled
- 3) Professor shall be able to specify which class is allowed to schedule office hours
- 4) Professor shall be able to block a whole day from hosting office hours
- 5) Professor shall be able to monitor schedules hours on the calendar

System Tasks:

- 1) System shall make reserved time slots unavailable
- 2) System shall send an email when a user signs up
- 3) System shall add scheduled office hours to personal calendar
- 4) System shall check professor's calendar for free slots
- 5) System shall add selected office hours to professor's calendar
- 6) System shall send email to student and professor when office hours scheduled
- 7) System shall send email to student/professor when meeting cancelled.

List of non-functional requirements

Availability:

- 1) Scheduling Office Hours System shall be available for use 24/7
- 2) Register/Login System shall be available for use 24/7
- 3) In the event that a system is non-operational, there will be a notification to inform the users about its unavailability

Documentation:

- 1) All documents of the system must be embedded in source code

Capacity:

- 1) All system should be able to handle at least 10 users at once
- 2) All system should be able to handle at least 20 scheduling at once

Interoperability:

- 1) The system shall be able to work on any browser (Chrome, IE, Firefox, etc)
- 2) Should be compatible with mobile usage

Maintainability:

- 1) Any system shall not be down for maintenance for more than 24 hours

Performance:

- 1) Load time shall not take more than 3 seconds

Security:

- 1) Access will be granted to only legitimate users
- 2) Access to only registered users
- 3) The system will store passwords as hashed
- 4) Every unsuccessful attempt by user to access the system will be recorded

Competitive analysis

	Outlook	iLearn	Google Calendar	GaterDater
Calendar	+	+	+	++
Messages	+	+	-	+
Reservation	-	-	-	++
Search/Filter	+	-	-	+
Reminder	-	-	+	+

In terms of scheduling, all the applications listed above can perform some type of scheduling. Like the analysis above, all competitors contain a calendar like Google Calendar to notify the users of the current month and day. However, GaterDater brought it up to another level. Where the users are able to not only view but also reserve a time slot from the calendar. Which solves iLearn and Outlook's problems of both students and professors being clueless on the status of office hours. Referring back to reserving time slots, the competitors doesn't provide a reservation function. Where as GaterDater allows users to reserve or unreserve their desired time slot with their professor. In addition, users are able to search for their respective professor to schedule hours for more simplicity. Unlike most of the other applications, it will also become color coordinated to indicate the current status regarding hours. Therefore, GaterDater will be the next big thing.

High-level system architecture and technologies used

Server Host: Google Compute Engine 1vCPU 614MB RAM (Free tier for now)

Operating System: Ubuntu 18.04 Server

Database: MySQL 8.0.17

Web Server: Apache 2.4.41

Server-Side Language: Node.js, PHP

Additional Technologies: PHPMyAdmin, maven, bootstrap

IDE: IntelliJ

Web Analytics: Google Analytics

SSL Cert: Lets Encrypt (Cert Bot)

SASS: 3.5.5

Team

Tejas Vajjhala: Team Lead, Github Master

Sanil Rijal: Front-end Lead

Thomas Zhang: Front-end assist

Aaron Li: Back-End Lead, Database Master

Checklist

Team found a time slot to meet outside of the classroom: DONE

Github master chosen: DONE

Team decided and agreed together on using the listed SW tools and deployment server:
DONE

Team ready and able to use the chosen back and front end frameworks and those who
need to learn are working on learning and practicing: DONE

Team lead ensured that all team members read the final M1 and agree/ understand it
before submission: DONE

Github organized as discussed in class (e.g. master branch, development branch, folder
for milestone documents etc.) DONE

Milestone 2:

GatorDater

Team 204 (local):

Tejas Vajjhala (tvajjhala@gmail.com): Team Lead, Github Master

Sanil Rijal: Front-end Lead

Thomas Zhang: Front-end assist

Aaron Li: Back-End Lead, Database Master

Milestone	Version	Date Submitted For Review
Milestone 2	2	11/8/2019
Milestone 2	1	10/24/2019
Milestone 1	2	10/14/2019
Milestone 1	1	10/03/2019

Table of Contents

Table of Contents	2
Data Definition	3
Functional Requirements	4
Priority	5
UI Mockups and Storyboards	6
High Level Database Architecture & Organization	16
ERD	17
Database Model	18
High Level APIs and Main Algorithms	19
High Level UML Diagrams	20
High Level Application Network and Deployment Diagrams	21
Current Risks	22
Project Management	22

Data Definitions V2

Users: User can be registered or unregistered.

1. **Registered User - Student:** Registered user who is able to browse classes and schedule office hours with a professor.
2. **Registered User - Professor:** Registered user who is able to allocate hours for office hour scheduling and manage his/her appointments.
3. **Unregistered User - Guest:** Unregistered user who has the ability to search for professors and view their available office hours.
4. **Administrator:** A registered user who has the ability to ban users, create and delete appointments, and blacklist email addresses.

Application:

1. **Calendar:** Calendar to display scheduled office hours associated with students and professors
 - a. **Office Hour:** record of the date and time with professors name

Functional Requirements V2

For Registered Users:

- 1) Users shall be able to make an account
- 2) Users shall be able to login
- 3) Users shall be able to access the office hour page
- 4) Users shall be able to cancel their scheduled office hours appointment
- 5) Users should be able to edit profile
- 6) Users shall be able to specify which class they belong to

For Registered Users That Are Students:

- 1) Users shall be able to search for their professors
- 2) Users shall be able to reserve a time slot

For Registered Users That Are Professors:

- 1) Professor shall be able to post their available time slots.
- 2) Professor shall be able to restrict hours from being scheduled
- 3) Professor shall be able to specify which class is allowed to schedule office hours

System Tasks:

- 1) System shall make reserved time slots unavailable
- 2) System shall send an email when a user signs up
- 3) System shall add scheduled office hours to personal calendar
- 4) System shall check professor's calendar for free slots
- 5) System shall add selected office hours to professor's calendar
- 6) System shall send email to student and professor when office hours scheduled
- 7) System shall send email to student/professor when meeting cancelled.

Priorities

Priority 1:

- 1) User shall be able to make an account
- 2) User shall be able to login
- 3) User shall be able to logout
- 4) User shall be able to access the office hour scheduling page
- 5) User shall be able to search for professors
- 6) Users shall be able to update their name
- 7) Users shall be able to update their password
- 8) Users shall be able to declare whether they are a student or professor
- 9) Users shall be able to contact the developers for any issues
- 10) Users shall be able to view a professor's open time slots
- 11) Students shall be able to reserve a time slot with a professor
- 12) Students shall be able to cancel their time slot with a professor
- 13) Professors shall be able to cancel their appointment with a student
- 14) Professors shall be able to block office hour days and time
- 15) Professors shall be able to specify their free time slot
- 16) Professors shall be able to update their name
- 17) Professors shall be able to update their password
- 18) Professors shall be able to contact the developers for any issues
- 19) Professors shall be able to view their appointments with students on calendar

Priority 2:

- 1) Registered users can message other registered users
- 2) Registered users can update their profiles
- 3) Registered users can add their courses
- 4) Professors shall be verified by an admin

Priority 3:

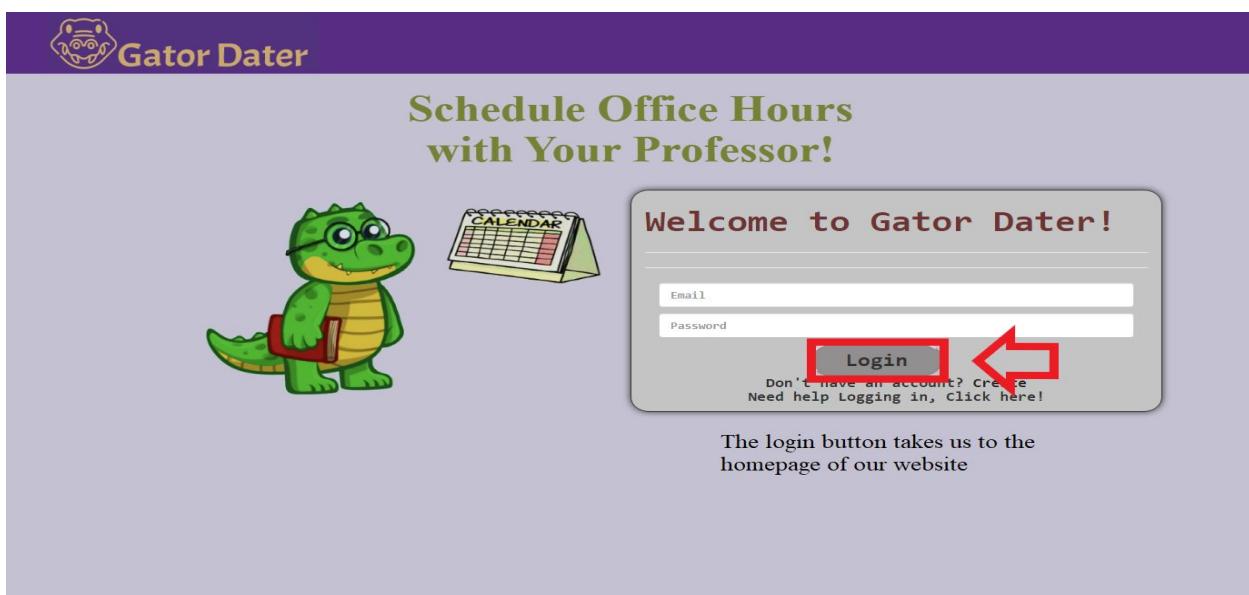
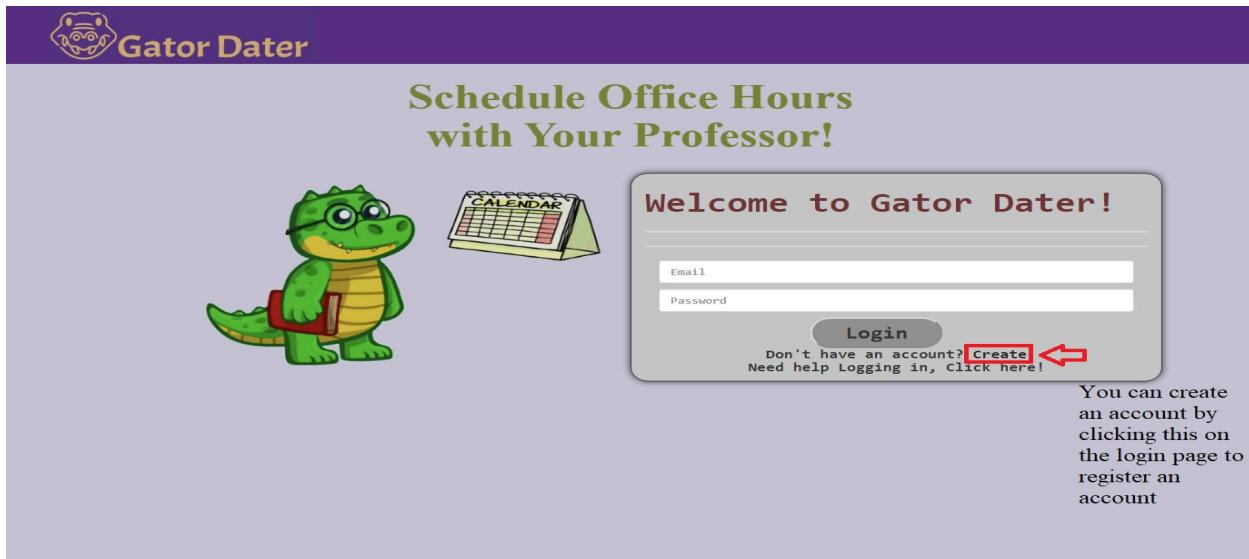
- 1) Upon reservation/cancellation, notifications will be sent towards both registered users
- 2) Registered users can input events into calendar
- 3) Users can receive a text message notification on their phone
- 4) Professor can get an alert notifying the end of an appointment

UI Mockups and Storyboards

Navbar:

- Home
- Update/Courses (depending on Professor/Student status)
- Calendar
- Inbox
- Help
- **Register/Login/Logout:** Popup in the top right corner.

Login Page:



Register Page:

- Asks if the user has registered or not. If they have not, register them immediately after they enter in their information
- Option to take them to the home page if they have registered or skip to browse

Create Account

Name

Email

Are You

Major

Password

Confirm Password

Submit

A drop-down box to choose if you're a student or professor

After you are done entering in your credentials, you click the submit button to create your account.

User Home Page/Professor Search:

The screenshot shows the Gator Dater user home page. At the top, there is a purple header bar with the "Gator Dater" logo and a welcome message "Welcome Daniel Tomasevich, Logout". On the left, there is a sidebar with icons for Account, Update, Calendar, Inbox, and Help. The main area has a search bar with the placeholder "Search for your Professor". A red arrow points to the search results below the bar, which show a single result: "jose" (highlighted in a red box) and "Jose Ortiz-Costa". Below the search results, there is a descriptive text: "On the home page, we have the search function for our professors. After searching and clicking on the professor, you get transferred to the website where you can schedule with them."

Firstname	Lastname	Class	About
Jose	Ortiz-Costa	CSC 648	I have not set an about me text yet.

Currently, this is the page that you get directed to, but it will change in the future



User Scheduling Page:

First, we see a calendar with open time slots after clicking on a professor's name. Then we select the time slot a student wants and then get to book that time slot.

The screenshot shows a monthly calendar for January 2020. The days of the week are labeled at the top: SUN, MON, TUES, WED, THURS, FRI, SAT. The dates are arranged in a grid. Days with available time slots are highlighted with green boxes labeled "Open". Days without available time slots are highlighted with red boxes labeled "Close".

Daniel Tomasevich Hours						
Today				January 2020		
SUN	MON	TUES	WED	THURS	FRI	SAT
		1	2	3	4	5
6	7 Open	8	9 Open	10	11	12
13	14 Close	15	16 Open	17	18	19
20	21 Close	22	23 Open	24	25	26
27	28 Close	29	30 Open	31	1	2

The screenshot shows a detailed view for Monday, Jan 7, 2020. It lists four time slots: 3:00pm, 3:15pm, 3:30pm, and 3:45pm. Each slot is labeled "Open" and has a "Add to your Calendar" button next to it.

Jan 7, 2020 - Monday			
	3:00pm	Open	Add to your Calendar
	3:15pm	Close	
	3:30pm	Open	Add to your Calendar
	3:45pm	Open	Add to your Calendar

User Confirming Appointment:

The screenshot shows the Gator Dater application interface. At the top, there is a dark purple header bar with a yellow gator logo on the left, the text "Gator Dater" in white, and "Jacob Pluto, Logout" on the right. Below the header is a sidebar on the left containing icons and text for "Account", "Courses", "Calendar", "Inbox", and "Help". The main area is titled "Confirmation" and contains a message: "Jan 7, 2020 - 3:30pm Appointment with Daniel Tomasevich has been added to your Calendar."

User Cancelling Appointment:

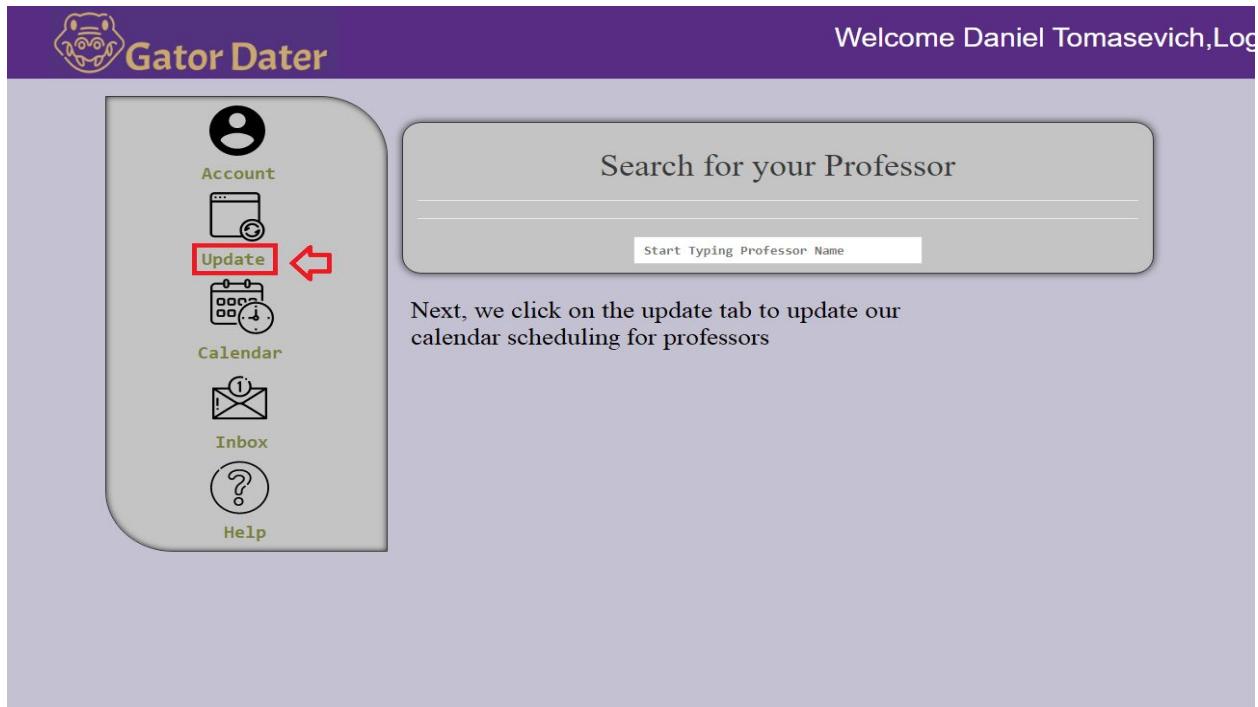
The screenshot shows the Gator Dater application interface. At the top, there is a dark purple header bar with a yellow gator logo on the left, the text "Gator Dater" in white, and "Jacob Pluto, Logout" on the right. Below the header is a sidebar on the left containing icons and text for "Account", "Courses", "Calendar", "Inbox", and "Help". The main area is titled "Cancel" and contains a message: "Jan 7, 2020 - 3:30pm with Daniel Tomasevich has been Cancelled."

Professor Cancelling Appointment:

The screenshot shows the Gator Dater application interface. At the top, there is a dark purple header bar with a yellow gator logo icon on the left, followed by the text "Gator Dater". On the right side of the header, it says "Daniel Tomasevich, Logout". Below the header is a sidebar on the left containing icons and labels for "Account", "Update", "Calendar", "Inbox", and "Help". The main area is titled "Confirmation" and contains a message box. The message box displays the date and time "Jan 7, 2020 - 3:30pm" and asks "Are you sure you want to Cancel this appointment? Please Confirm." It has two buttons at the bottom: "Yes" (in green) and "No" (in red).

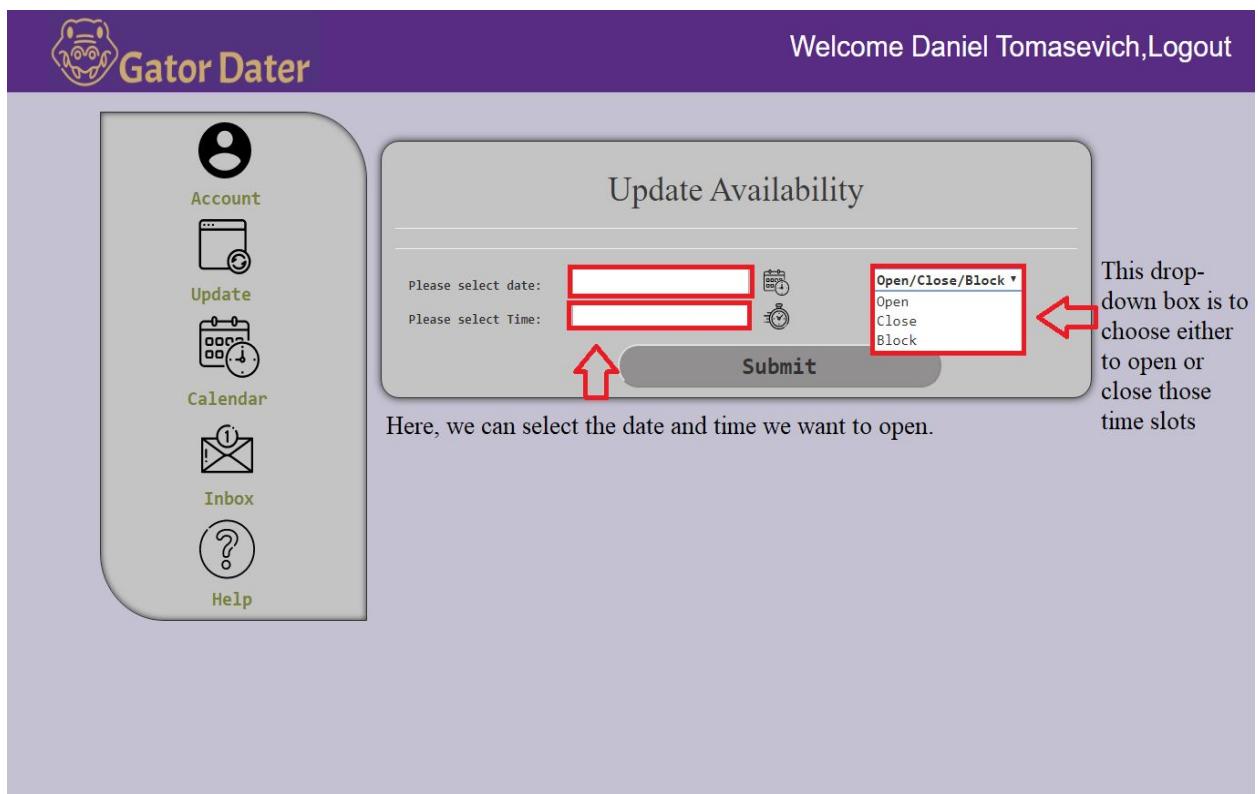
This screenshot shows the Gator Dater application after a cancellation. The interface is similar to the previous one, with the dark purple header, yellow gator logo, and "Gator Dater" text. The user name "Daniel Tomasevich, Logout" is also present. The sidebar on the left includes "Account", "Update", "Calendar", "Inbox", and "Help". The main area is titled "Cancel" and contains a message box. The message box displays the date and time "Jan 7, 2020 - 3:30pm" and the text "Appointment with Jacob Pluto has been Cancelled." in red.

Professor Update Page:



The screenshot shows the Gator Dater homepage. At the top right, it says "Welcome Daniel Tomasevich,Logout". On the left, there's a sidebar with icons for Account, Update (which is highlighted with a red box and a red arrow pointing to it), Calendar, Inbox, and Help. The main area has a search bar with "Search for your Professor" and "Start Typing Professor Name".

Next, we click on the update tab to update our calendar scheduling for professors



The screenshot shows the "Update Availability" page. At the top right, it says "Welcome Daniel Tomasevich,Logout". On the left, there's a sidebar with icons for Account, Update (highlighted with a red box and a red arrow pointing to it), Calendar, Inbox, and Help. The main area has fields for "Please select date:" and "Please select Time:", both with red boxes around them. Below these is a dropdown menu labeled "Open/Close/Block" with options "Open", "Close", and "Block", also with a red box and a red arrow pointing to it. A red arrow points up to the "Submit" button.

Here, we can select the date and time we want to open.

This drop-down box is to choose either to open or close those time slots

Professor Calendar:

The screenshot shows the Gator Dater homepage. On the left, there is a vertical navigation menu with icons for Account, Update, Calendar (which is highlighted with a red box), Inbox, and Help. A red arrow points from the text below to the 'Calendar' icon. On the right, there is a search bar with the placeholder 'Start Typing Professor Name'. The top right corner displays the welcome message 'Welcome Daniel Tomasevich'.

Next, we click on the calendar to go to our monthly calendar.

The screenshot shows the monthly calendar for January 2020. The days of the week are labeled at the top: SUN, MON, TUES, WED, THURS, FRI, SAT. The dates are arranged in a grid. Several dates have annotations: January 7 has a green box labeled 'Meeting at 3:30pm' with 'Cancel' and 'Open' buttons; January 8 has a purple box labeled 'Blocked' with 'Open' and 'Cancel' buttons; January 14 and January 15 both have red arrows pointing to them, with a red text overlay stating 'Here, we can see blocked dates and meetings that have been scheduled'.

SUN	MON	TUES	WED	THURS	FRI	SAT
		1	2	3	4	5
6	7 Meeting at 3:30pm Cancel Open	8 Blocked Open	9	10	11	12
13	14 ↑ Here, we can see blocked dates and meetings that have been scheduled	15 ↑	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2

Student/Professor Inbox:

Welcome Daniel Tomasevich,Logout

Search for your Professor

Start Typing Professor Name

Account

Update

Calendar

Inbox

Help

The inbox will take you to your mailbox where you have messages from professors or notifications in general.

Jacob Pluto, Logout

Account

Courses

Calendar

Inbox

Help

Inbox

- Your appointment has been accepted.
- You cancelled your appointment with Daniel Tomasevich.

Professor/Student Help

The screenshot shows the Gator Dater website interface. At the top, there is a purple header bar with the text "Welcome Daniel Tomasevich,Logout". On the left, a sidebar contains icons for Account, Update, Calendar, Inbox, and Help. The "Help" icon is highlighted with a red box and a red arrow points to it from the bottom right. To the right of the sidebar is a search bar with the placeholder "Search for your Professor" and a text input field labeled "Start Typing Professor Name". Below the search bar, there is some descriptive text: "The help button leads you to where you can message the admins if a problem appears."

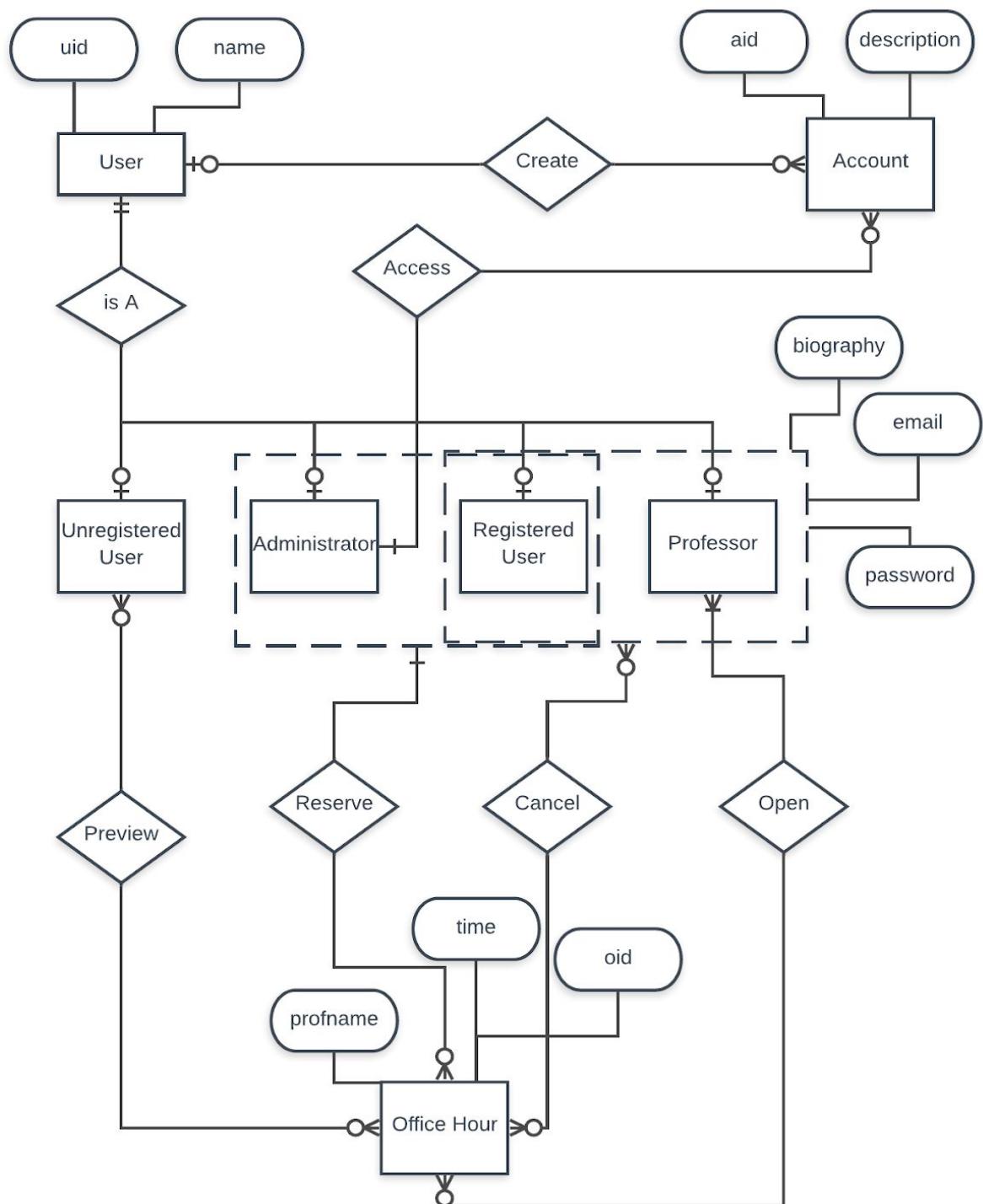
The screenshot shows the "Help" page of the Gator Dater website. The top navigation bar includes the Gator Dater logo, the name "Daniel Tomasevich", and a "Logout" link. The left sidebar is identical to the one on the previous page, featuring icons for Account, Update, Calendar, Inbox, and Help. The main content area is titled "Help" and contains fields for "Email:" and "Title:", both represented by empty input boxes. Below these fields is a large text area labeled "Please type your message:". At the bottom right of the form is a "Submit" button.

High Level Database Architecture and Organization

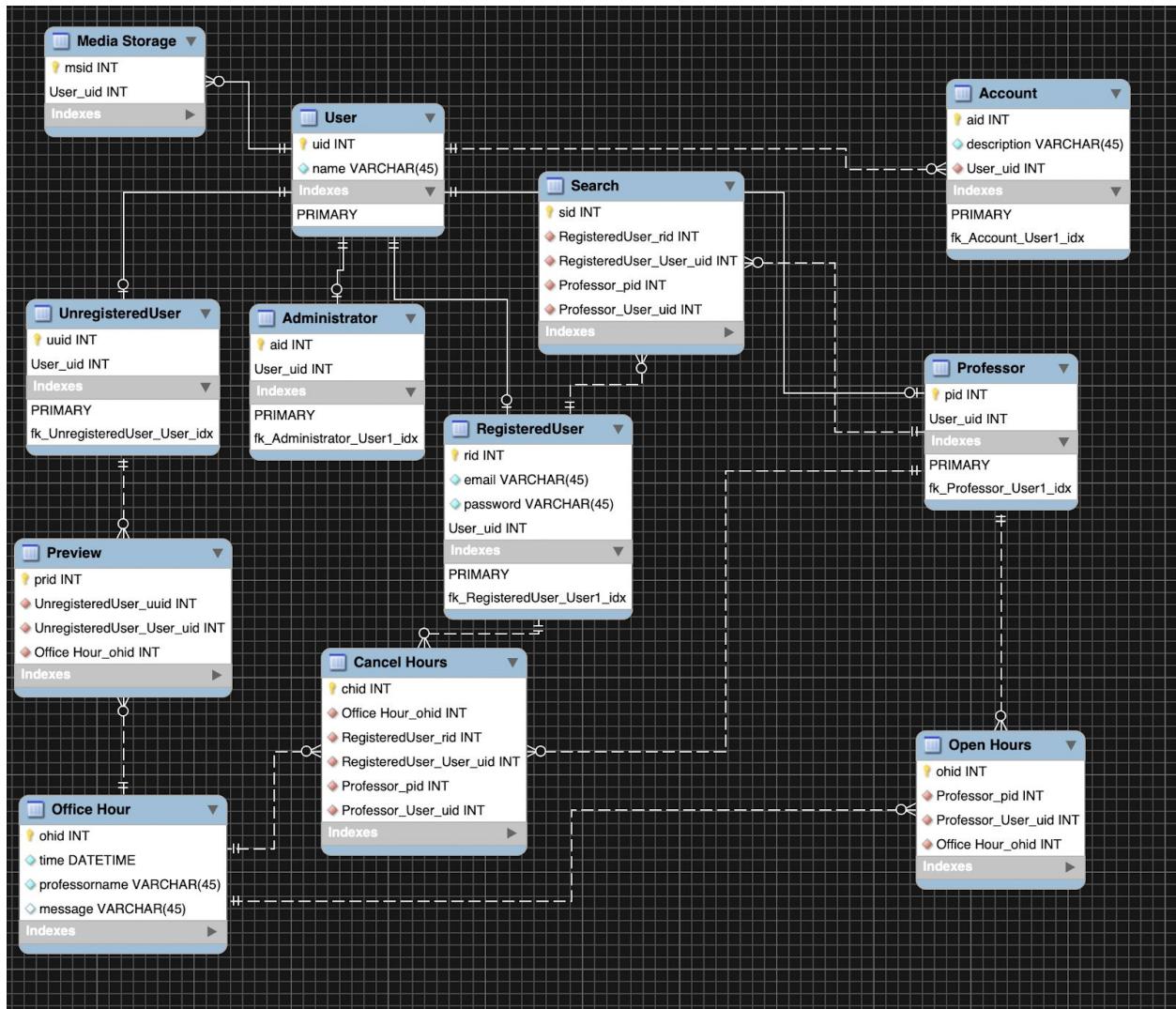
Business Rules :

1. Users can create multiple accounts
2. A registered user can reserve unlimited office hours
3. A professor can open multiple office hours
4. Each office hour is reserved by only one registered user
5. Unregistered users can preview unlimited office hours
6. A registered user can cancel unlimited office hours
7. A professor can cancel multiple office hours

ERD:



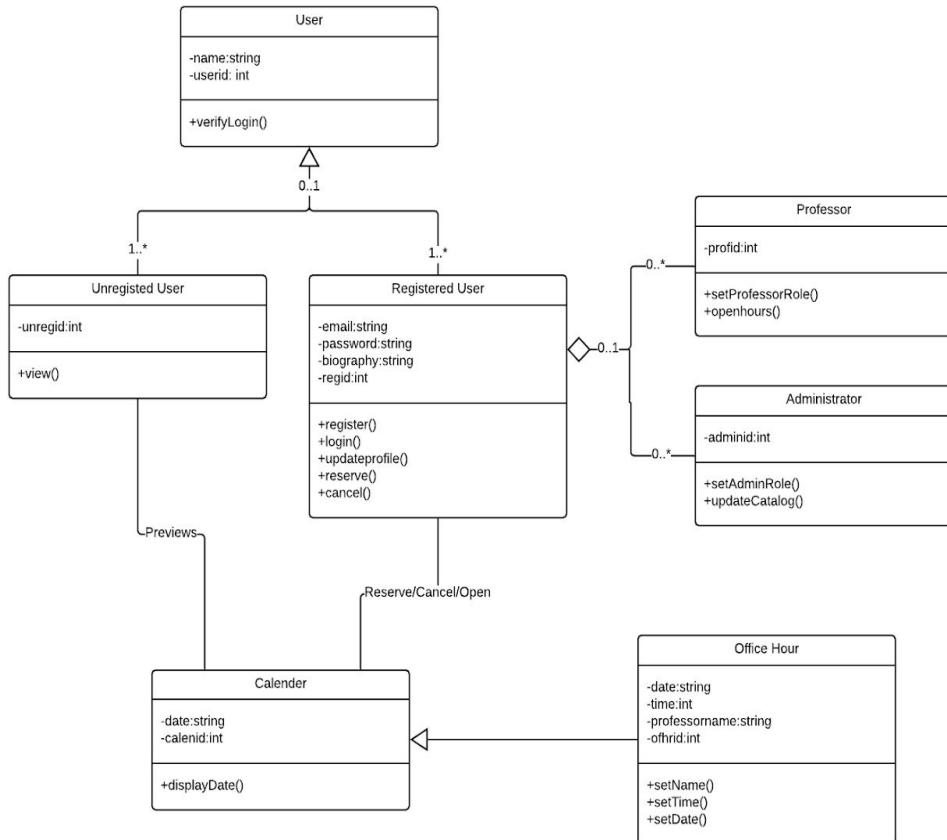
DB Model



High Level APIs and Main Algorithms

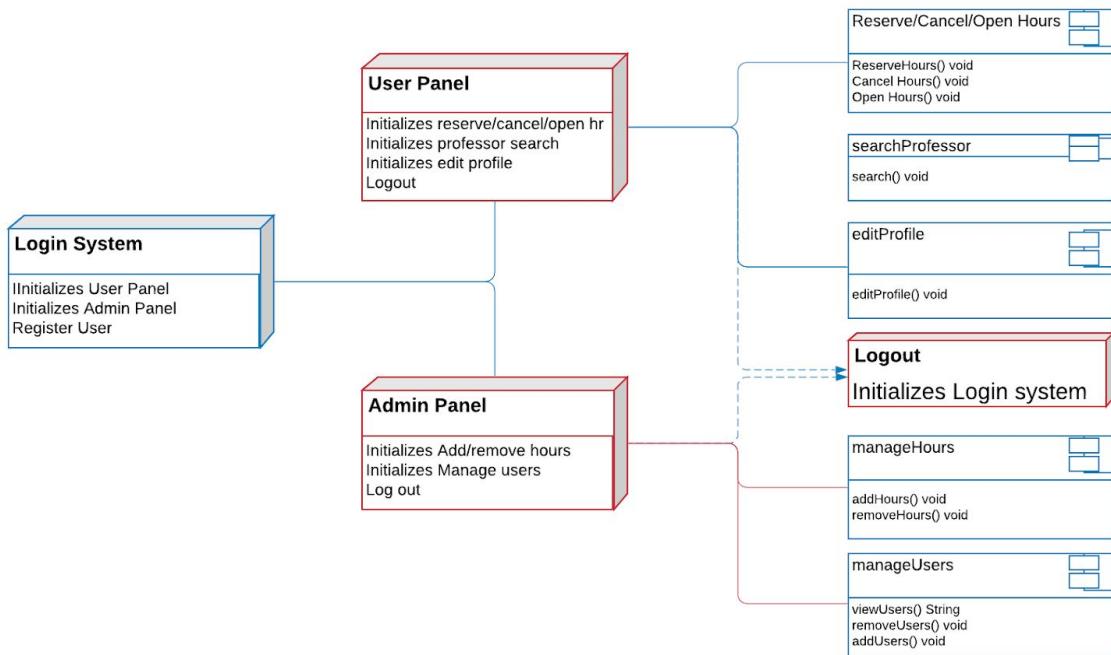
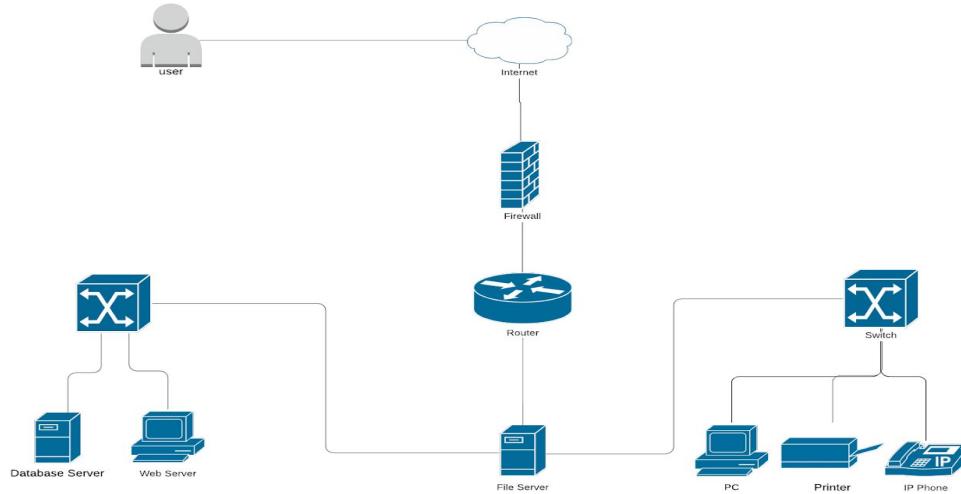
No real APIs are currently implemented or foreseen. Most of the backend communication will be via PHP -> MySQL connect and functions.

High Level UML Diagrams



The central class is the **User**. Associated with the **User** are an **Unregistered User**, **Registered User**, **Professors** and **Administrator**. **Unregistered User** are associated with only previewing the **Calender**. While **Registered User**, **Professor**, and **Administrator** are able to reserve/cancel/open to a **Calender**. The **Calender** contains date each with associated office hour.

High Level Application Network and Deployment Diagrams



Current Risks

Skills risks: The team as a whole possesses the skills to complete all aspects of the project. Some members need to be helped out/brought up to speed on how to implement certain features, but as a whole we do have the knowledge to do so and educate everyone to bring them to the same level.

Schedule risks: So far, we have been able to meet all the deadlines and have our prototypes meet all of the requirements on time. No issues with the delivery schedule.

Technical risks: None so far.

Teamwork risks: None so far, each team member is putting in effort and doing their best.

Legal/content risks: None so far.

Project Management

For M2, I split the load of the front end and back end to their respective teams through the use of trello. Where I was able to notify the team on what needed to be worked on. My front end team worked on the UI Mockups/Storyboards and my back end team worked on the high level database architecture and organization. I helped both sub-teams with their tasks. I also put together the vertical prototype. Not much help was needed from the front end or back end team for this part as it was pretty simple for a single team member to put together. What we really need to do moving forward is to bring the whole team up to speed on the latest UI mockup tools such as Figma, which would have made our UI mockups look a lot nicer. Right now the mockups were built using paint and word. I'll be sure to have a meeting specifically for teaching team members about other tools we can use rather than solely focusing on getting the work done and distributing load.

Revision #: 2

Date: November 23, 2019

SW Engineering CSC 648 / 848 Fall 2019

Milestone 3:

Review Functionality, UI, SW and planning for final product delivery

GatorDater

Team 204:

Tejas Vajjhala: Team Lead, Github Master

Sanil Rijal: Front-end Lead

Thomas Zhang: Front-end assist

Aaron Li: Back-End Lead, Database Master

Milestone	Version	Date Submitted For Review
Milestone 3	2	11/23/2019
Milestone 3	1	11/21/2019
Milestone 2	2	11/8/2019
Milestone 2	1	10/24/2019
Milestone 1	2	10/14/2019
Milestone 1	1	10/03/2019

Table of Contents

Table of Contents	2
Data Definition	3
Functional Requirements	4
Priority	5
UI Mockups and Storyboards	6
High Level Database Architecture & Organization	16
ERD	17
Database Model	18
High Level APIs and Main Algorithms	19
High Level UML Diagrams	20
High Level Application Network and Deployment Diagrams	21
Current Risks	22
Project Management	23
UI and Functionality Feedback	24
Offline Review	25

Data Definitions V2

Users: User can be registered or unregistered.

1. **Registered User - Student:** Registered user who is able to browse classes and schedule office hours with a professor.
2. **Registered User - Professor:** Registered user who is able to allocate hours for office hour scheduling and manage his/her appointments.
3. **Unregistered User - Guest:** Unregistered user who has the ability to search for professors and view their available office hours.
4. **Administrator:** A registered user who has the ability to ban users, create and delete appointments, and blacklist email addresses.

Application:

1. **Calendar:** Calendar to display scheduled office hours associated with students and professors
 - a. **Office Hour:** record of the date and time with professors name

Functional Requirements V2

For Registered Users:

- 1) Users shall be able to make an account
- 2) Users shall be able to login
- 3) Users shall be able to access the office hour page
- 4) Users shall be able to cancel their scheduled office hours appointment
- 5) Users should be able to edit profile
- 6) Users shall be able to specify which class they belong to

For Registered Users That Are Students:

- 1) Users shall be able to search for their professors
- 2) Users shall be able to reserve a time slot

For Registered Users That Are Professors:

- 1) Professor shall be able to post their available time slots.
- 2) Professor shall be able to restrict hours from being scheduled
- 3) Professor shall be able to specify which class is allowed to schedule office hours

System Tasks:

- 1) System shall make reserved time slots unavailable
- 2) System shall send an email when a user signs up
- 3) System shall add scheduled office hours to personal calendar
- 4) System shall check professor's calendar for free slots
- 5) System shall add selected office hours to professor's calendar
- 6) System shall send email to student and professor when office hours scheduled
- 7) System shall send email to student/professor when meeting cancelled.

Priorities

Priority 1:

- 1) User shall be able to make an account
- 2) User shall be able to login
- 3) User shall be able to logout
- 4) User shall be able to access the office hour scheduling page
- 5) User shall be able to search for professors
- 6) Users shall be able to update their name
- 7) Users shall be able to update their password
- 8) Users shall be able to declare whether they are a student or professor
- 9) Users shall be able to contact the developers for any issues
- 10) Users shall be able to view a professor's open time slots
- 11) Students shall be able to reserve a time slot with a professor
- 12) Students shall be able to cancel their time slot with a professor
- 13) Professors shall be able to cancel their appointment with a student
- 14) Professors shall be able to block office hour days and time
- 15) Professors shall be able to specify their free time slot
- 16) Professors shall be able to update their name
- 17) Professors shall be able to update their password
- 18) Professors shall be able to contact the developers for any issues
- 19) Professors shall be able to view their appointments with students on calendar

Priority 2:

- 1) Registered users can send messages to other registered users
- 2) Registered users can update their profiles
- 3) Registered users can add their courses
- 4) Professors shall be verified by an admin

Priority 3:

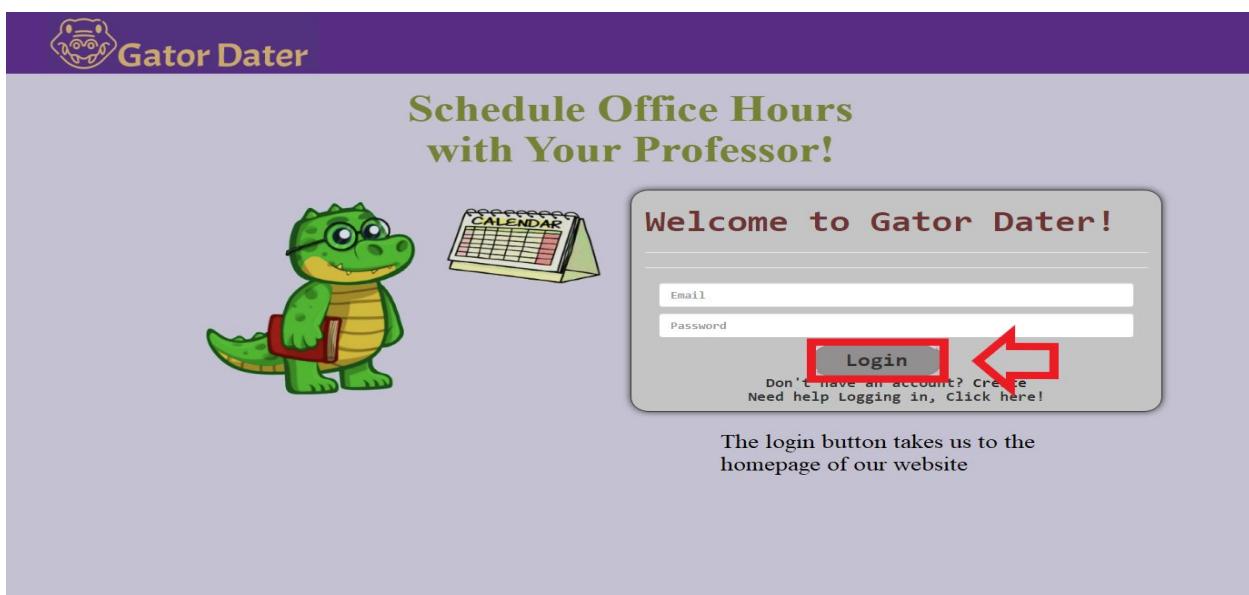
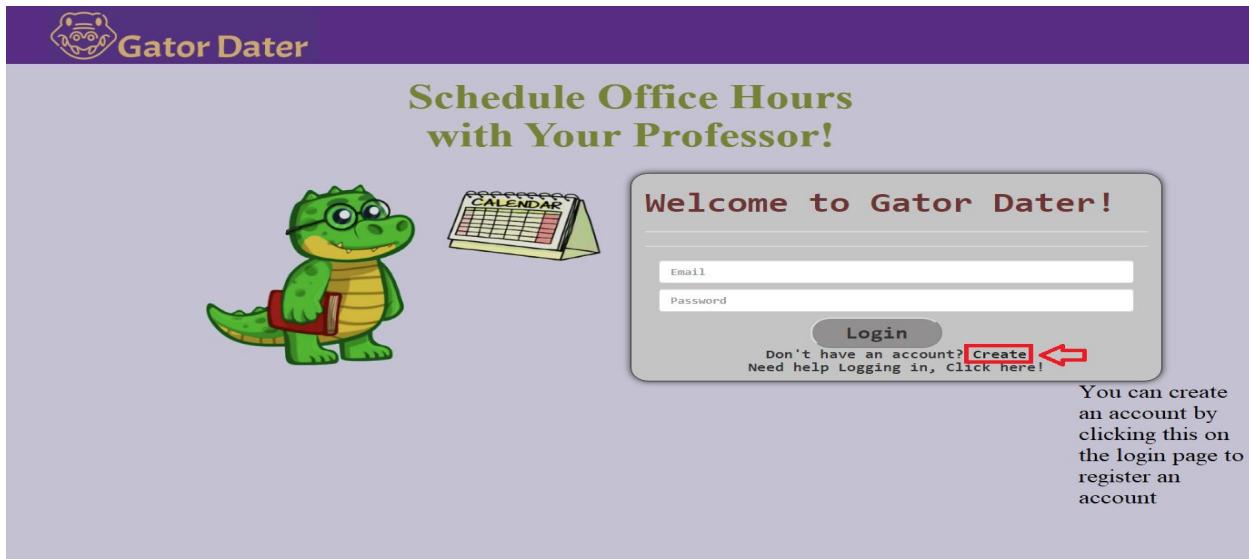
- 1) Upon reservation/cancellation, notifications will be sent towards both registered users
- 2) Registered users can input events into calendar
- 3) Users can receive a text message notification on their phone
- 4) Professor shall get an alert notifying the end of an appointment

UI Mockups and Storyboards

Navbar:

- Home
- Update/Courses (depending on Professor/Student status)
- Calendar
- Inbox
- Help
- **Register/Login/Logout:** Popup in the top right corner.

Login Page:



Register Page:

- Asks if the user has registered or not. If they have not, register them immediately after they enter in their information
- Option to take them to the home page if they have registered or skip to browse

Create Account

Name

Email

Are You

Major

Password

Confirm Password

Submit

A drop-down box to choose if you're a student or professor

After you are done entering in your credentials, you click the submit button to create your account.

User Home Page/Professor Search:

The screenshot shows the Gator Dater user home page. At the top, there is a purple header bar with a cartoon gator icon and the text "Gator Dater". On the right side of the header, it says "Welcome Daniel Tomasevich, Logout". Below the header is a sidebar on the left containing icons for Account, Update, Calendar, Inbox, and Help. To the right of the sidebar is a search interface with a text input field containing "jose". A red arrow points from the text input field towards the explanatory text below. The explanatory text reads: "On the home page, we have the search function for our professors. After searching and clicking on the professor, you get transferred to the website where you can schedule with them."

Firstname	Lastname	Class	About
Jose	Ortiz-Costa	CSC 648	I have not set an about me text yet.

Currently, this is the page that you get directed to, but it will change in the future



User Scheduling Page:

First, we see a calendar with open time slots after clicking on a professor's name. Then we select the time slot a student wants and then get to book that time slot.

The screenshot shows a monthly calendar for January 2020. The days of the week are labeled at the top: SUN, MON, TUES, WED, THURS, FRI, SAT. The dates are arranged in a grid. Days with available time slots are highlighted with green boxes labeled "Open". Days without available time slots are highlighted with red boxes labeled "Close".

Daniel Tomasevich Hours						
Today				January 2020		
SUN	MON	TUES	WED	THURS	FRI	SAT
		1	2	3	4	5
6	7 Open	8	9 Open	10	11	12
13	14 Close	15	16 Open	17	18	19
20	21 Close	22	23 Open	24	25	26
27	28 Close	29	30 Open	31	1	2

The screenshot shows a detailed view for Monday, Jan 7, 2020. It lists four time slots: 3:00pm, 3:15pm, 3:30pm, and 3:45pm. Each slot is labeled "Open" and has a "Add to your Calendar" button next to it.

Jan 7, 2020 - Monday			
	3:00pm	Open	Add to your Calendar
	3:15pm	Close	
	3:30pm	Open	Add to your Calendar
	3:45pm	Open	Add to your Calendar

User Confirming Appointment:

The screenshot shows the Gator Dater application interface. At the top, there is a dark purple header bar with a yellow gator logo on the left, the text "Gator Dater" in white, and "Jacob Pluto, Logout" on the right. Below the header is a sidebar on the left containing icons and text for "Account", "Courses", "Calendar", "Inbox", and "Help". The main area is titled "Confirmation" and contains a message: "Jan 7, 2020 - 3:30pm Appointment with Daniel Tomasevich has been added to your Calendar."

User Cancelling Appointment:

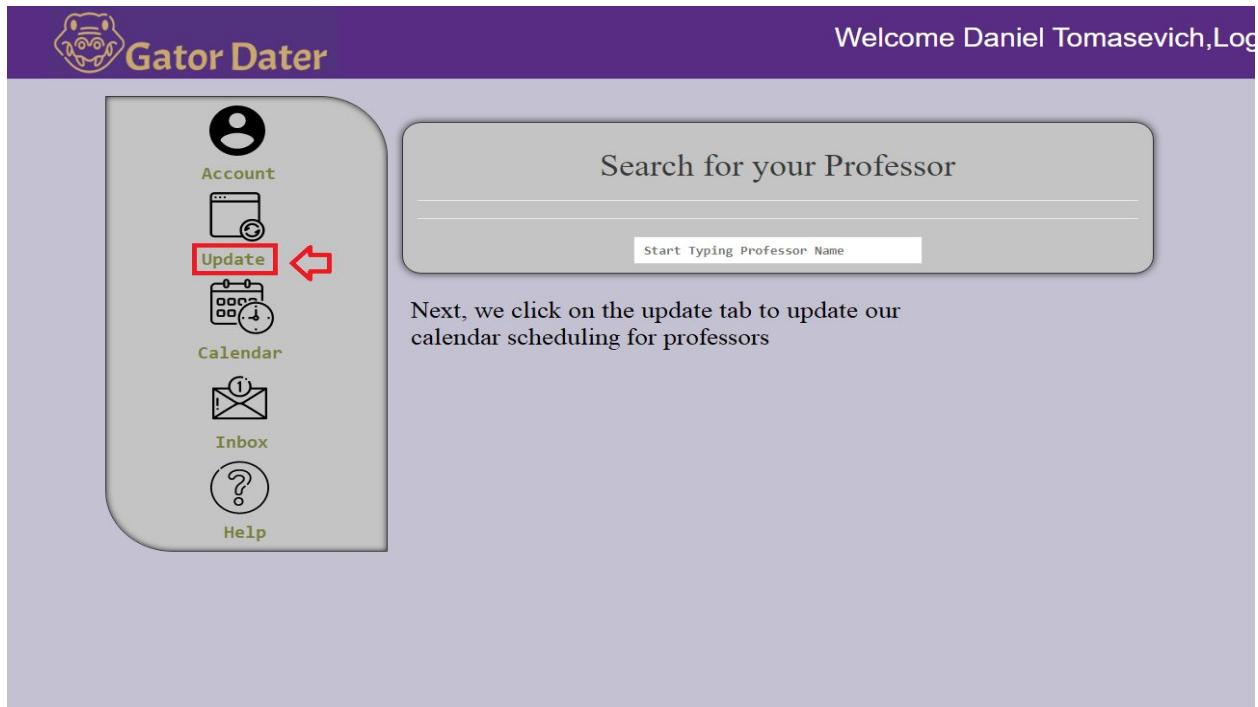
The screenshot shows the Gator Dater application interface. At the top, there is a dark purple header bar with a yellow gator logo on the left, the text "Gator Dater" in white, and "Jacob Pluto, Logout" on the right. Below the header is a sidebar on the left containing icons and text for "Account", "Courses", "Calendar", "Inbox", and "Help". The main area is titled "Cancel" and contains a message: "Jan 7, 2020 - 3:30pm with Daniel Tomasevich has been Cancelled."

Professor Cancelling Appointment:

The screenshot shows the Gator Dater application interface. At the top, there is a dark purple header bar with a yellow gator logo icon on the left, followed by the text "Gator Dater". On the right side of the header, it says "Daniel Tomasevich, Logout". Below the header is a sidebar on the left containing icons and labels for "Account", "Update", "Calendar", "Inbox", and "Help". The main area is titled "Confirmation" and contains a message box. The message box displays the date and time "Jan 7, 2020 - 3:30pm" and asks "Are you sure you want to Cancel this appointment? Please Confirm." It has two buttons at the bottom: "Yes" (in green) and "No" (in red).

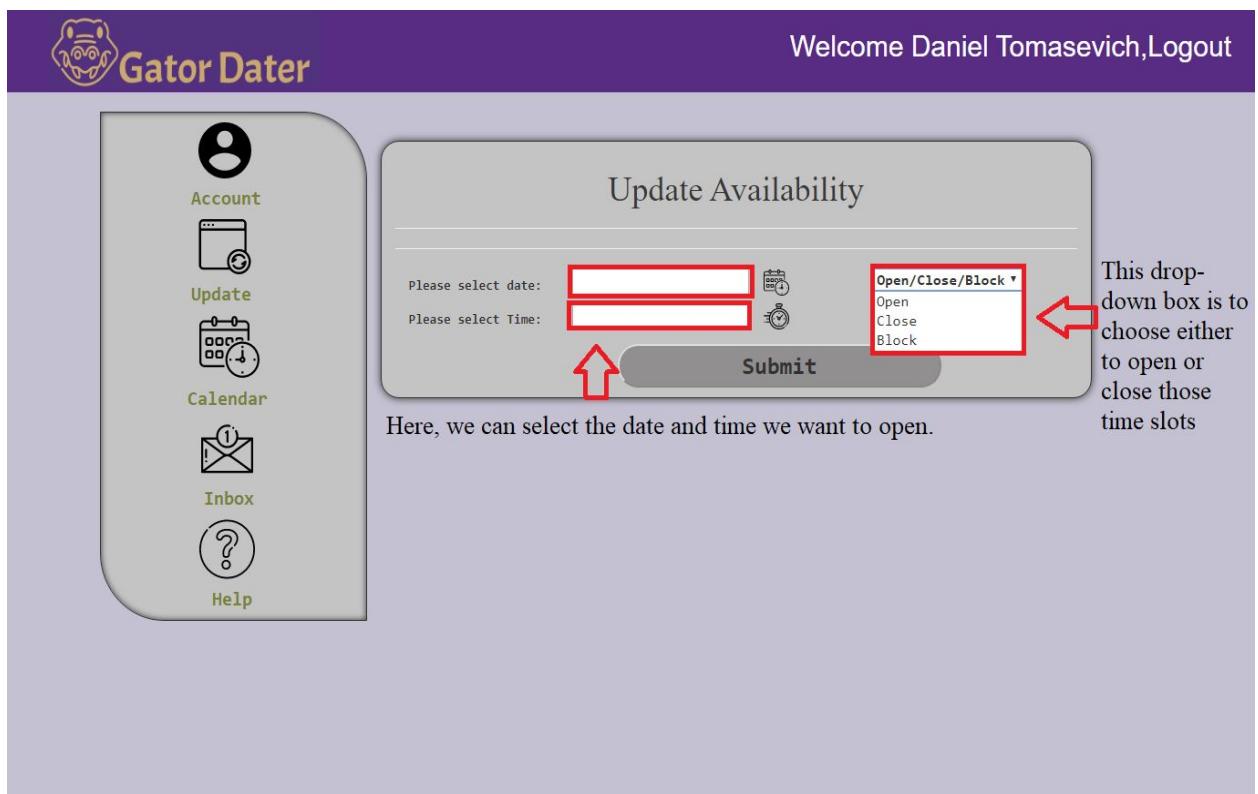
This screenshot shows the Gator Dater application after a cancellation. The interface is similar to the previous one, with the dark purple header, yellow gator logo, and "Gator Dater" text. The user name "Daniel Tomasevich, Logout" is also present. The sidebar on the left includes "Account", "Update", "Calendar", "Inbox", and "Help". The main area is titled "Cancel" and contains a message box. The message box displays the date and time "Jan 7, 2020 - 3:30pm" and the text "Appointment with Jacob Pluto has been Cancelled." in red.

Professor Update Page:



The screenshot shows the Gator Dater homepage. At the top right, it says "Welcome Daniel Tomasevich,Logout". On the left, there's a sidebar with icons for Account, Update (which is highlighted with a red box and a red arrow pointing to it), Calendar, Inbox, and Help. The main area has a search bar with "Search for your Professor" and "Start Typing Professor Name".

Next, we click on the update tab to update our calendar scheduling for professors



The screenshot shows the "Update Availability" page. At the top right, it says "Welcome Daniel Tomasevich,Logout". On the left, there's a sidebar with icons for Account, Update (highlighted with a red box and a red arrow pointing to it), Calendar, Inbox, and Help. The main area has fields for "Please select date:" and "Please select Time:", both with red boxes around them. Below these is a dropdown menu labeled "Open/Close/Block" with options "Open", "Close", and "Block", also with a red box and a red arrow pointing to it. A red arrow points up to the "Submit" button.

Here, we can select the date and time we want to open.

This drop-down box is to choose either to open or close those time slots

Professor Calendar:

The screenshot shows the Gator Dater homepage. On the left, there is a vertical navigation menu with icons for Account, Update, Calendar (which is highlighted with a red box), Inbox, and Help. A red arrow points from the text below to the 'Calendar' icon. On the right, there is a search bar with the placeholder 'Start Typing Professor Name'. The top right corner displays the welcome message 'Welcome Daniel Tomasevich'.

Next, we click on the calendar to go to our monthly calendar.

The screenshot shows the monthly calendar for January 2020. The days of the week are labeled from Sunday to Saturday. The calendar grid shows the following details:

SUN	MON	TUES	WED	THURS	FRI	SAT
		1	2	3	4	5
6	7 Meeting at 3:30pm Cancel	8 Blocked Open	9	10	11	12
13	14 ↑ Here, we can see blocked dates and meetings that have been scheduled	15 ↑	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2

A red arrow points to the 'Meeting at 3:30pm' entry on January 7th, and another red arrow points to the 'Blocked' status on January 8th. A red box highlights the text 'Here, we can see blocked dates and meetings that have been scheduled'.

Student/Professor Inbox:

The screenshot shows the Gator Dater application interface. At the top, there is a purple header bar with a cartoon gator logo on the left, the text "Gator Dater" in the center, and "Welcome Daniel Tomasevich, Logout" on the right. Below the header is a sidebar on the left containing icons for Account, Update, Calendar, and a red-bordered "Inbox" icon, which is highlighted with a red arrow. To the right of the sidebar is a large search box with the placeholder "Search for your Professor" and a smaller input field below it labeled "Start Typing Professor Name".

The screenshot shows the Gator Dater application interface. At the top, there is a dark purple header bar with a cartoon gator logo on the left, the text "Gator Dater" in the center, and "Jacob Pluto, Logout" on the right. Below the header is a sidebar on the left containing icons for Account, Courses, Calendar, and a red-bordered "Inbox" icon. To the right of the sidebar is a large "Inbox" section with the title "Inbox" at the top. Inside the inbox, there are two red circular bullet points with corresponding messages: "Your appointment has been accepted." and "You cancelled your appointment with Daniel Tomasevich."

Professor/Student Help

The screenshot shows the Gator Dater website interface. At the top, there is a purple header bar with the text "Welcome Daniel Tomasevich,Logout". On the left, a sidebar contains icons for Account, Update, Calendar, Inbox, and Help. The "Help" icon is highlighted with a red box and a red arrow pointing to it. To the right of the sidebar is a search bar labeled "Search for your Professor" with a placeholder "Start Typing Professor Name". Below the search bar, a message states: "The help button leads you to where you can message the admins if a problem appears."

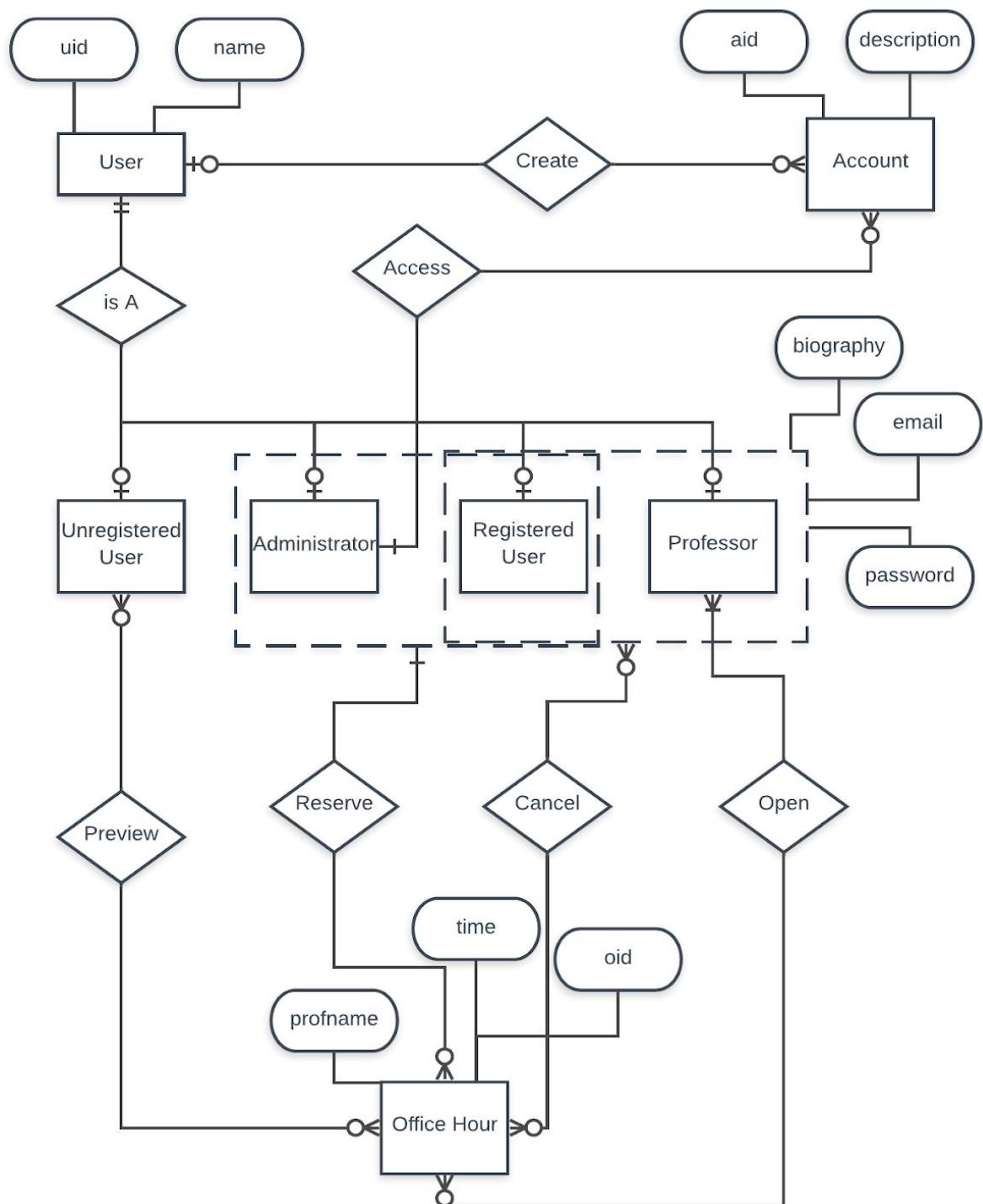
The screenshot shows the "Help" page of the Gator Dater website. The top navigation bar includes the Gator Dater logo, the name "Daniel Tomasevich", and a "Logout" link. The left sidebar is identical to the one on the previous page, featuring icons for Account, Update, Calendar, Inbox, and Help. The main content area is titled "Help" and contains fields for "Email:" and "Title:", both represented by empty input boxes. Below these fields is a large text area labeled "Please type your message:". At the bottom right of the form is a "Submit" button.

High Level Database Architecture and Organization

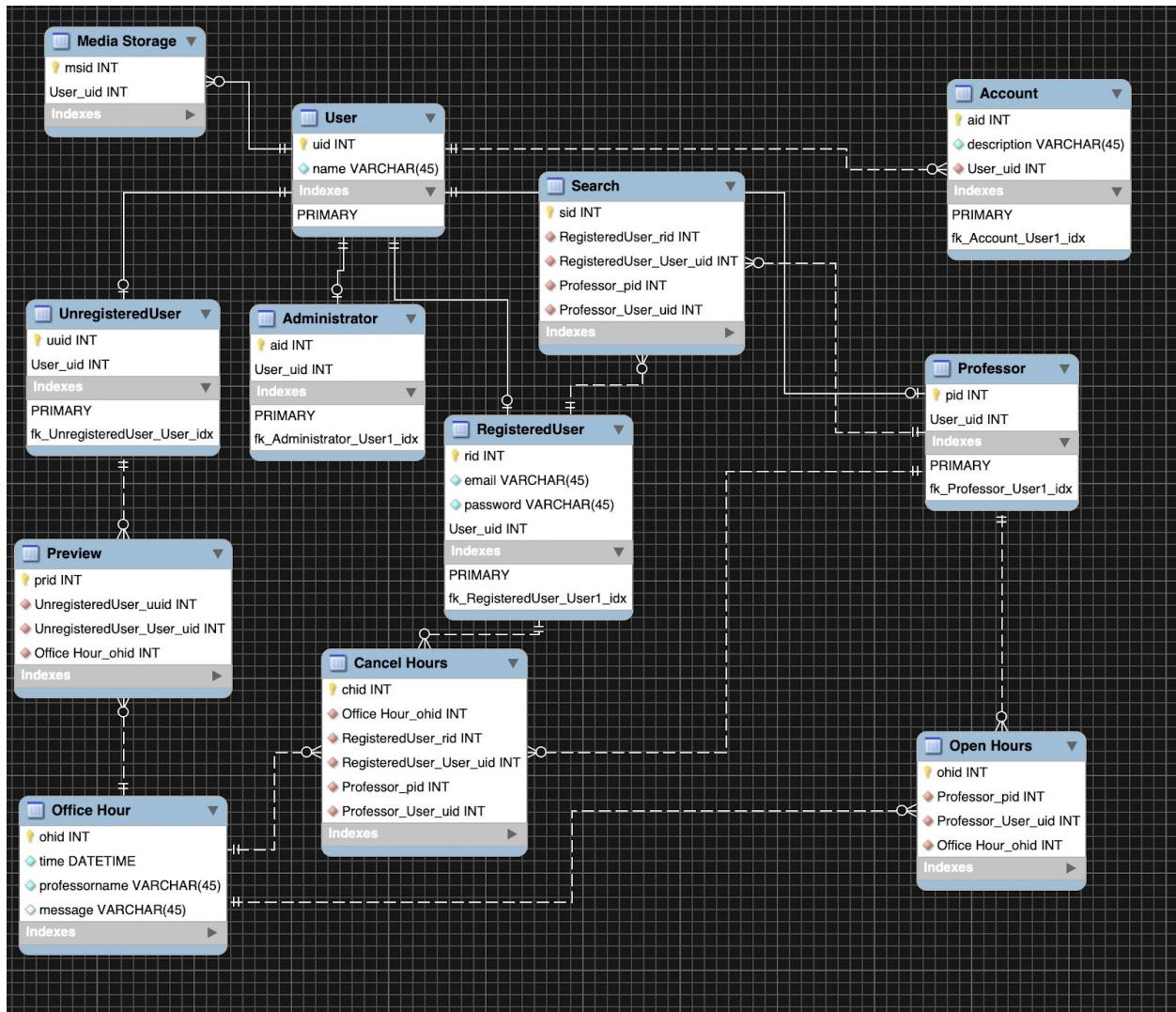
Business Rules :

1. Users can create multiple accounts
2. A registered user can reserve unlimited office hours
3. A professor can open multiple office hours
4. Each office hour is reserved by only one registered user
5. Unregistered users can preview unlimited office hours
6. A registered user can cancel unlimited office hours
7. A professor can cancel multiple office hours

ERD:



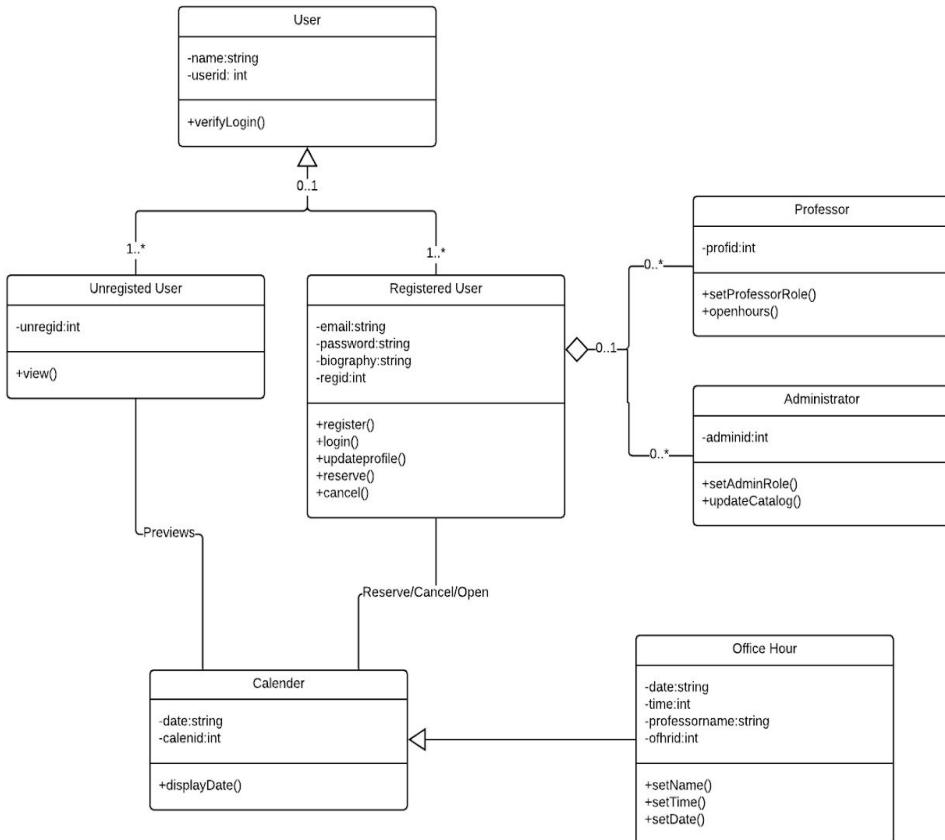
DB Model



High Level APIs and Main Algorithms

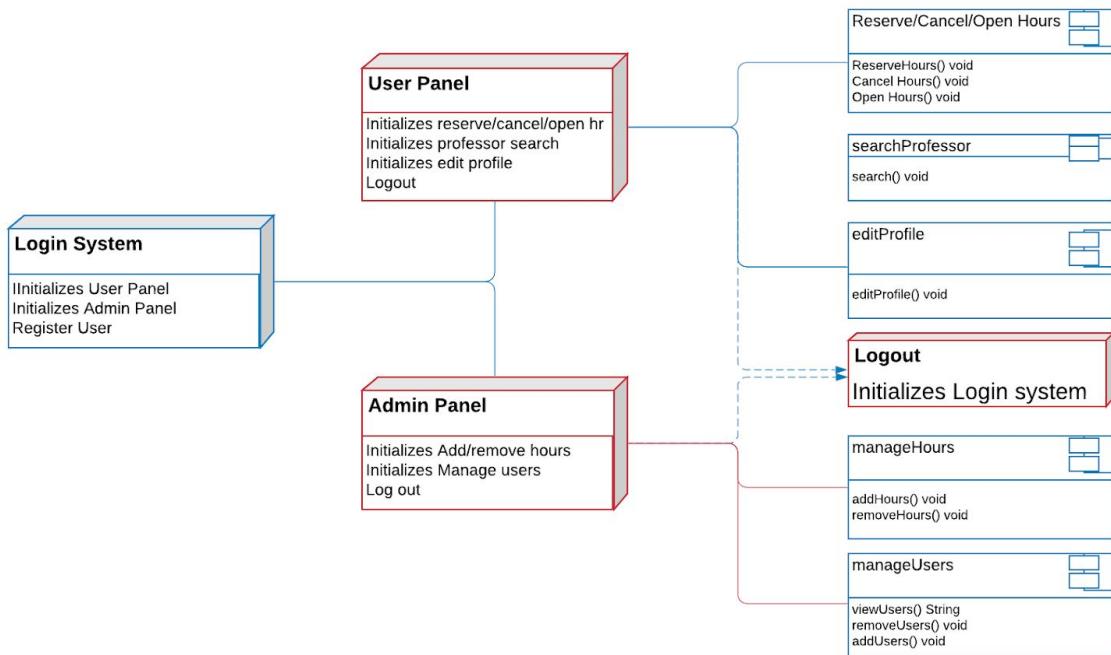
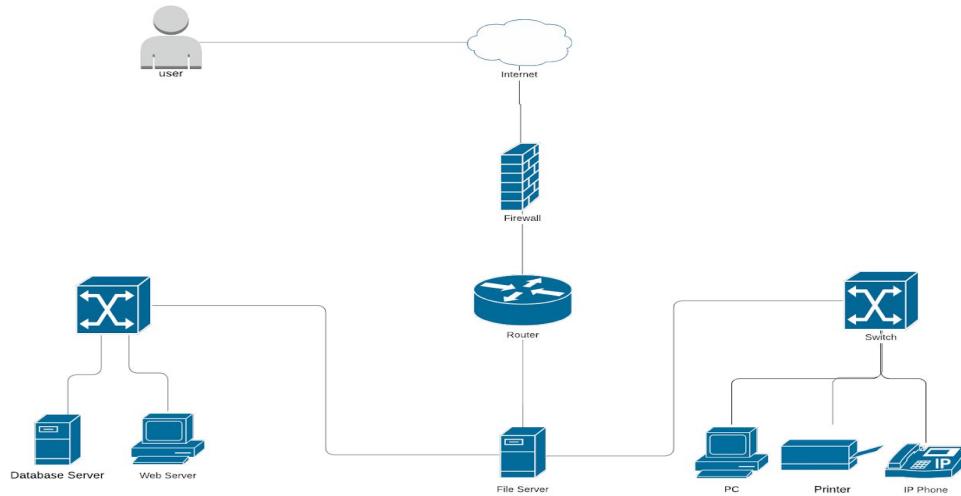
No real APIs are currently implemented or foreseen. Most of the backend communication will be via PHP -> MySQL connect and functions.

High Level UML Diagrams



The central class is the **User**. Associated with the **User** are an **Unregistered User**, **Registered User**, **Professors** and **Administrator**. **Unregistered User** are associated with only previewing the **Calender**. While **Registered User**, **Professor**, and **Administrator** are able to reserve/cancel/open to a **Calender**. The **Calender** contains date each with associated **Office Hour**.

High Level Application Network and Deployment Diagrams



Current Risks

Skills risks: The team as a whole possesses the skills to complete all aspects of the project. Some members need to be helped out/brought up to speed on how to implement certain features, but as a whole we do have the knowledge to do so and educate everyone to bring them to the same level.

Schedule risks: So far, we have been able to meet all the deadlines and have our prototypes meet all of the requirements on time. No issues with the delivery schedule.

Technical risks: None so far.

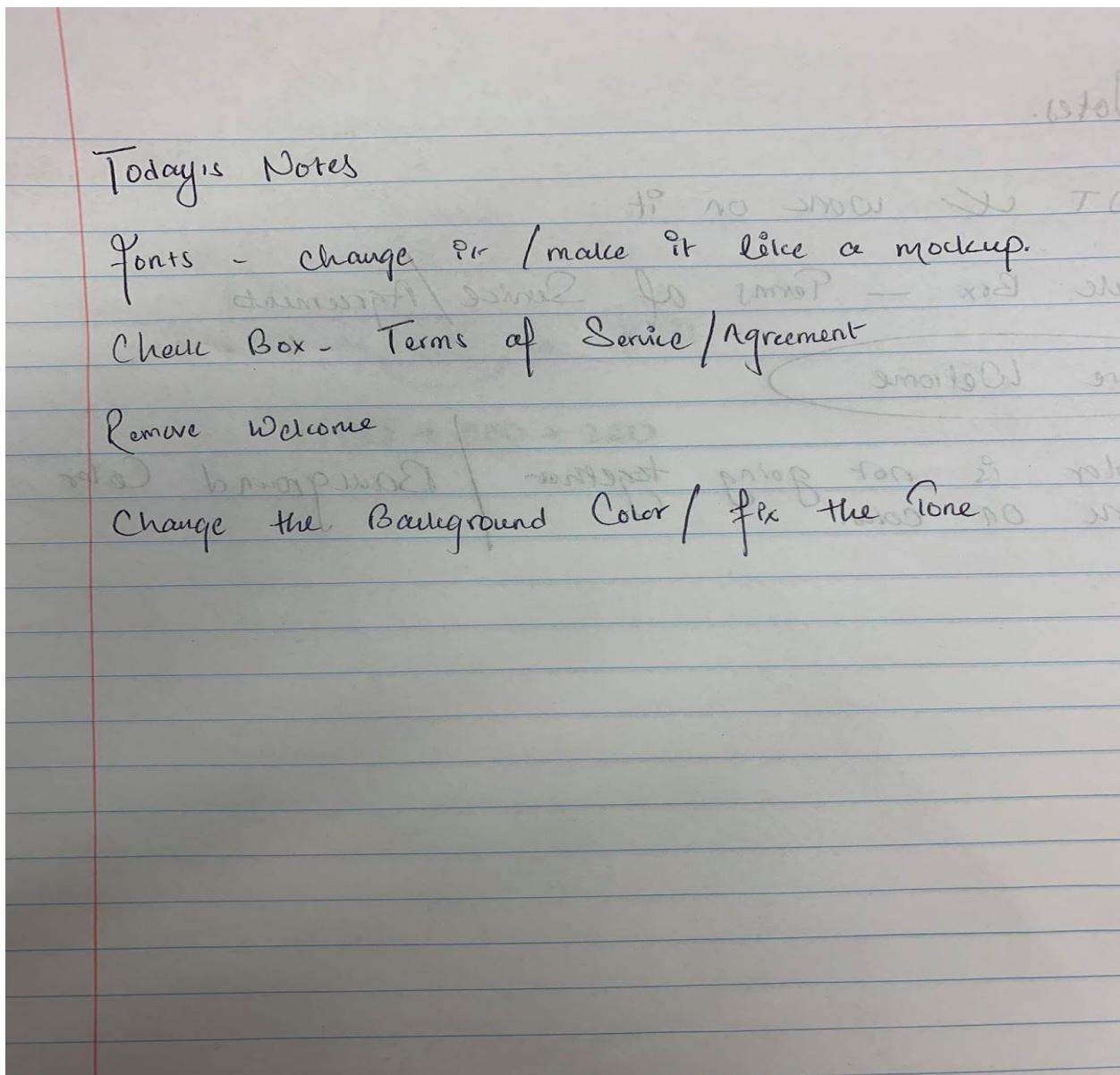
Teamwork risks: None so far, each team member is putting in effort and doing their best.

Legal/content risks: None so far.

Project Management

For M2, I split the load of the front end and back end to their respective teams through the use of trello. Where I was able to notify the team on what needed to be worked on. My front end team worked on the UI Mockups/Storyboards and my back end team worked on the high level database architecture and organization. I helped both sub-teams with their tasks. I also put together the vertical prototype. Not much help was needed from the front end or back end team for this part as it was pretty simple for a single team member to put together. What we really need to do moving forward is to bring the whole team up to speed on the latest UI mockup tools such as Figma, which would have made our UI mockups look a lot nicer. Right now the mockups were built using paint and word. I'll be sure to have a meeting specifically for teaching team members about other tools we can use rather than solely focusing on getting the work done and distributing load.

UI and Functionality feedback



The major takeaway from the UI review was the change of color scheme and fonts. As some fonts were considered to be hard to read or inconsistent from page to page. In addition, towards the UI, the background color was suggested to be changed to something lighter and easy on the users eyes. As we take Jose's feedback and implement it into the new and improved UI, we have fonts that are all consistent and easy to read, white color background, and Terms of Service/Agreement for the registration page.

Offline Review

CSC648-848 Software Engineering: Offline M3 Check

JC

Jose Ortiz Costa (via ilearn) <jortizco@sfsu.edu>

Fri 11/22/2019 5:50 PM

Aaron Li ✉



Hi all,

I finished the offline check for M3. There are some minor problems that need to be solved in your repositories, but in general everything seems in order. I was going to send individual reports to each team lead, but I decided to write a general email, instead, because most of those problems are generalized across all the teams.

Some common problems in your repositories that need your attention (if applicable)

1. Some teams haven't updated the main README file yet. That was a M0 task. The main README file should contain your names, roles, name of your project, a small description of the services and functionalities provided by your project (1 or 2 sentences max), and the url to your site. Fix this ASAP before M4 (if needed)
2. I took a quick look into your V2 documents of each milestones, and I saw that a few teams are not addressing the problems stated in V1 of those milestones. I must remind you that V2 of each milestone is the one that will be graded,
3. Contributions to the repo are not balanced. I see students with a lot of commits done, and I see other students from the same team with little commits. Again, number, content, and quality of work in commits is also graded. In addition, quality of the commits comments are graded as well. In the software industry, an excellent comment in a commit can save a lot of headaches and money when a bug needs to be fixed.

General comments about your horizontal prototype:

1. I told you this many times in class: simplicity is not equal to simple design. Simplicity in SW engineering is related to the concepts of a product being easy to use, and usable. Take that into consideration when you are doing your UI refactoring for M4 and M5. This is not a SW development class where you can do a simple project and you don't look back to it anymore. This is a SW engineering class and the quality of your project is part of your grade. So, keep that in mind. Also, your home page should state clearly what services your product is going to offer in a few words.
2. Yesterday, many teams did not implement all the storyboards. In addition, there were teams which horizontal prototype was a totally different version than the storyboards they brought to me. This is not acceptable in this class. This prototype was not graded, but make sure that your final UI is similar to your storyboards. Storyboards and mockups are there for a reason, and they represent the vision from the client about their needs in the product. If you change this, then you are putting your needs before the needs of the client.
3. I got many reports from team leads that students are committing to the repository or sending work at the last minute, and in some specific cases, just not sending any work at all. This is not acceptable in this class. Every time you commit or submit work to your team leads, they need time to review the work to make sure that everything is in order, and the quality of the work the one expected.

Please if some of the above apply to your team fix them ASAP.

Happy thanksgiving to everyone, and I'll see you in next class.

Jose.

Based off the offline review and in-person meeting, our team doesn't fall within those criterias/problems mentioned within the general mail.

Milestone 4:

GatorDater

Team 204 (local):

Tejas Vajjhala (tvajjhala@gmail.com): Team Lead, Github Master

Sanil Rijal (srijal0509@gmail.com): Front-end Lead

Thomas Zhang(t.zhang70@gmail.com): Front-end assist

Aaron Li (ali16@mail.sfsu.edu): Back-End Lead, Database Master

Milestone	Version	Date Submitted For Review
Milestone 4	2	12/10/2019
Milestone 4	1	12/05/2019
Milestone 3	2	11/23/2019
Milestone 3	1	11/21/2019
Milestone 2	2	11/8/2019
Milestone 2	1	10/24/2019
Milestone 1	2	10/14/2019
Milestone 1	1	10/03/2019

Table of Contents

Table of Contents	1
Product Summary	2
Usability Test Plan	3
QA Test Plan	5
Code Review	6
External Code Review	11
Self Check: Best Practices for Security	16
Self Check: Adherence to Original Non-Functional Specs	18

Product Summary

Name of Product: GatorDater

Current project URL: http://35.203.131.184/m3/Index_Page.html

All Major Committed Functions:

- Users shall be able to make an account
- Users shall be able to login
- Users shall be able to access the office hour scheduling page
- Users shall be able to reserve a time slot with a professor
- Users shall be able to cancel their time slot with a professor
- Users shall be able to search for professors
- Professors shall be able to cancel their appointment with a student
- Professors shall be able to block office hour days and time
- Professors shall be able to specify their free time slot
- Users shall be able to update their name
- Users shall be able to update their password
- Users shall be able to declare whether they are a student or professor
- Users shall be able to contact the developers for any issue
- Professors shall be able to view their appointments with students on calendar
- Professors shall be able to update their name
- Professors shall be able to update their password
- Professors shall be able to contact the developer for any issues

What is unique about GatorDater:

Currently, there is nothing like GatorDater that has been implemented for SFSU. For scheduling office hours, students need to either email the professor and try to come up with a time to meet, or sign up via a paper list posted outside the teachers room. This is very inconvenient as its difficult to sync up with professors and stay organized.

GatorDater allows users and professors to schedule appointments via an easy to use web interface and displays upcoming appointments, along with the ability to cancel, and reschedule them seamlessly.

Usability Test Plan

Test Case: Search for a professor

- **Test Objective:**
 - The professor search feature is a major component of our application. This is the first step to scheduling office hours. Without this feature, the rest of the application would not be usable.
 - Our goal is for the user to be able to search for the professor via the search bar, select the professor that matches the input, and then get redirected to the professor's scheduling page where the user can then schedule office hours with them.
 -
- **Test Description:**
 - **System setup:**
 - The system will be set up with the latest version of Chrome, Safari, or FireFox. The operating system will be either OSX, Windows, or Linux.
 - **Starting point:**
 - The starting point of the test case will be at the user's home page which is located at /homepage.html. This is the page that includes the professor search bar, which a user can use to find a professor to schedule an appointment with.
 - **Intended users:**
 - The intended user for this operation would be a student, as they would be the ones who are searching for a professor and attempting to schedule office hours.
 - **URL of the system:**
 - The url of the system where the test will begin is /homepage.html.
 - **What is to be measured:**
 - We are measuring the ability to fetch results from the database that match the inputted search term.

- Usability task description:
 - The user will begin at their homepage and locate the search bar in the center of the page.
 - The user will then enter a professor's name, and will press the search button or hit enter.
 - Based on whether the professor is in our system, the user will be presented with a list of professors that match the search term, or be presented with text informing them that no professor was found for the entered name. When clicking on a professor's name, the user is then redirected to the professor's scheduling page.
- Questionnaire:
 - Action: The user enters a professor's name and then clicks search or presses enter to submit the query.
 - Question: I was satisfied with the speed of the search bar.
 - Strongly Disagree
 - Disagree
 - Agree
 - Strongly Agree
 - Action: After submitting a search query, the user receives a list of professors to choose from, or no list if no matching professor exists.
 - Question: I was satisfied with the results of the search bar.
 - Strongly Disagree
 - Disagree
 - Agree
 - Strongly Agree
 - Action: The user finds the search bar on their homepage and utilizes it to find a professor.
 - Question: I was able to easily use the search bar without any confusion.
 - Strongly Disagree
 - Disagree
 - Agree
 - Strongly Agree

QA Test Plan

Test objectives:

Ensure the professor search feature is functional and returns a list of professor names that match or partially match the search query.

HW and SW setup (including URL):

The user will be using the latest build of GatorDater and load their homepage at /homepage. The user will be using the latest version of Chrome, Firefox, or Safari to conduct the test.

Feature to be tested:

The user will test the professor search functionality of our application.

QA Test plan:

Number	Description	Test Input	Expected Output	Result
1	Test % in LIKE professor_name field	“Jose”	Get the professors that have “Jose” in their name.	PASS
2	Test ability to load professor page from search result	Clicking on “Jose Ortiz-Costa” after searching “Jose”	Redirect to the professor’s page.	PASS
3	Test ability to display no professors found.	“Emily”	“No matches found”	PASS

Code Review

From: Aaron Li
Date: Sun, Dec 1, 2019 at 4:01 PM
Subject: Prof Search Code Review
To: tvajjhala@gmail.com

Hi Tejas,

The following is a snippet of the professor search code for review. I've hooked it from the homepage search Sanil made with the backend function you made for the VP.

HTML:

```
$(document).ready(function(){
  $('.search-box input[type="text"]').on("keyup input", function(){
    /* Get input value on change */
    var inputVal = $(this).val();
    var resultDropdown = $(this).siblings(".result");
    if(inputVal.length){
      $.get("vp/backend-search.php", {term: inputVal}).done(function(data){
        // Display the returned data in browser
        resultDropdown.html(data);
      });
    } else{
      resultDropdown.empty();
    }
  });

  // Set search input value on click of result item
  $(document).on("click", ".result p", function(){
    $(this).parents(".search-box").find('input[type="text"]').val($(this).text());
    //sets search field val
    $(this).parent(".result").empty();
    //clears dropdown
    username = $(this).text().replace(/\s/g,"");
    window.open("vp/user-page.php?user="+username);
  });
});

</script>
</head>
<body>
```

```
<p>
</p>
<div class="search-box">
  <input type="text" autocomplete="off" placeholder="Start Typing Professor Name" />
  <div class="result"></div>
</div>
</body>
```

--Aaron

Response:

From: Tejas Vajjhala
Date: Sun, Dec 1, 2019 at 4:45 PM
Subject: Re: Prof Search Code Review
To: liaaron727@gmail.com

Hey Aaron,

Thanks for sending me the code. Just a couple of changes I would make.
For the following:

```
$.get("vp/backend-search.php", {term: inputVal}).done(function(data)
```

I would change the path of the backend search php file to something that is NOT in the vp folder, as this will be deleted in production. We should keep the vp code separate from the production code. But not a huge change.

Second:

```
resultDropdown.html(data);
});
} else{
  resultDropdown.empty();
}
});
```

I would have the search be redirected to a new page rather than do a drop down of results. It kind of breaks the continuity/theme of the site as the drop down looks out of place. I think a good thing to do rather than a drop down is have the actual search results be displayed ON the page below the search bar (rather than a drop down field that originates from the search bar if that makes sense).

But the code looks solid, this is basically all we need for the search functionality. We'll just clean up how the output is displayed/formatted.

Thanks,
Tejas

External Code Review

From: Tejas Vajjhala
Date: Tues, Dec 3, 2019 at 3:57 PM
Subject: Prof Search Code Review
To: petarelectro@gmail.com

Hi Peter,

This following snippet is the code for updating user's name and password information from a user_list table. It'll contain php code and html as well where you can incorporate it into one file under .php. Let me know if there's any issues.

PHP:

```
<?php

session_start();

if(!isset($_SESSION["loggedin"]) || $_SESSION["loggedin"] !== true)
{
    header("location: Index_Page.php");
    exit;
}

require_once "config.php";

$new_name = $new_password = "";
$new_name_err = $new_password_err = "";

if($_SERVER["REQUEST_METHOD"] == "POST")
{
    if(empty($_POST["new_name"]))
    {
        $new_name_err = "Please enter a name.";
    }
    else{
        $new_name = trim($_POST["new_name"]);
    }
}
```

```
if(empty($_POST["new_password"]))
{
    $new_password_err = "Please enter a name.";
}
else{
    $new_password = trim($_POST["new_password"]);
}

if(empty($new_name_err) || empty($new_password_err)){
    $sql = "UPDATE users_list SET name =?,password=? WHERE uid = ?";
    if($stmt = mysqli_prepare($link, $sql)){
        mysqli_stmt_bind_param($stmt, "ssi", $param_name, $param_password,
        $param_id);
        $param_name = $new_name;
        $param_password =password_hash($new_password,
        PASSWORD_DEFAULT);
        $param_id = $_SESSION["id"];
        if(mysqli_stmt_execute($stmt)){
            session_destroy();
            header("location: homepage.php");
            exit();
        } else{
            echo "Oops! Something went wrong. Please try again later.";
        }
    }
    mysqli_stmt_close($stmt);
}
```

```
    mysqli_close($link);
}
?>
```

HTML:

```
<html>
<head>
    <meta charset="utf-8">
    <link rel="stylesheet" type="text/css" href="CSS/styles.css">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
        href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
    <script
        src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script
        src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>
</head>
<body>
    <div class="container-fluid">
        <nav class="navbar navbar-expand-sm bg-dark navbar-dark">
            <div class="col-md-12">
                <a class="navbar-brand" href="homepage.php"></a>
                <h2>Welcome <b><?php echo htmlspecialchars($_SESSION["name"]); ?>
                </b>,<a href="logout.php">Logout</a></h2>
            </div>
        </nav>
    </div>
    <div class="container">
        <div class="row">
            <!--Start Information ---->
            <div class="col-md-3">
                <div class="col-sm-12">
                    
                    <h4><a href="account_info_professor.php">Account</a></h4>
                </div>
                <div class="col-sm-12">
                    
                    <h4><a href="Update_Availability.php">Update</a></h4>
```

```

</div>
<div class="col-sm-12 Calendar">
    
    <h4><a href="Professor_Calander.php">Calendar</a></h4>
</div>
<div class="col-sm-12 Inbox">
    
    <h4><a href="inbox.php">Inbox</a></h4>
</div>
<div class="col-sm-12 Help">
    
    <h4><a href="help.php">Help</a></h4>
</div>
</div>
<!--End Information ---->
<!--Start Account Information-->
<div class="col-md-8">
    <h1>Account Information</h1>
    <hr>
    <hr>
    <!--Name satrt --->
    <form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>">
method="post">
    <div class="col-sm-2">
        Name:
    </div>
    <div class="col-sm-8">
        <input type="text" name="new_name" class="form-control" placeholder="Name">
    </div>
    <div class="col-sm-2">
        <div class="btn-group">
            <button><a href="#">Add</a>/<a href="#">Edit</a></button>
        </div>
    </div>
    <!--Name end --->
    <p>&nbsp;</p>
    <div class="col-sm-2">
        Status:
    </div>

```

```
</div>
<div class="col-sm-8">
    <input type="text" Name="Status" class="form-control" placeholder="Status">
</div>
<div class="col-sm-2">
    <div class="btn-group">
        <button><a href="#">Add</a>/<a href="#">Edit</a></button>
    </div>
</div>
<!--Status end --->
<p>&nbsp;</p>
<div class="col-sm-2">
    Courses:
</div>
<div class="col-sm-8">
    <input type="text" Name="Courses" class="form-control"
placeholder="Courses">
</div>
<div class="col-sm-2">
    <div class="btn-group">
        <button><a href="#">Add</a>/<a href="#">Edit</a></button>
    </div>
</div>
<!--Courses end --->
<p>&nbsp;</p>
<div class="col-sm-2">
    Password:
</div>
<div class="col-sm-8">
    <input type="password" name="new_password" class="form-control"
placeholder="*****">
</div>
<div class="col-sm-2">
    <div class="btn-group">
        <button><a href="#">Add</a>/<a href="#">Edit</a></button>
    </div>
</div>
<!--Password end --->
<!--Password End --->
```

```
<div class="col-sm-12">
    <input type="submit" Value="Update" name="Submit">
</div>
</form>
</div>
</body>
</html>
```

--Tejas

Response:

From: Peter Juric
Date: Tues, Dec 3, 2019 at 6:45 PM
Subject: Re: Prof Search Code Review
To: tvajjhala@gmail.com

Hey Tejas,

Thanks for sending me the code. Everything seems to be perfect and aligned. However, there was an error into updating both the username and password. Where if you'd update the password, the name would be empty while updating the name but not the password would make it empty as well.

```
if(empty($new_name_err) || empty($new_password_err)){
    $sql = "UPDATE users_list SET name =?,password=? WHERE uid = ?";
```

I would change the sql query into two different queries where one will “update users_list and set the name” when new_name_err is not empty and a seperate one when new_password_err to have another query of “update users_list set password. That should change the error within updating information.

Thanks,

Peter

Self-check On Best Practices For Security

- List of major assets that are being protected

- User passwords
 - UserIDs
 - User IP addresses
 - User email addresses
- We are encrypting user passwords using bcrypt.
 - Search input data is validated via character check to make sure all characters are letters, hyphens, or spaces. Therefore only valid names are searchable for the professor search. We are also preventing MySQL injection by using prepared statements in conjunction with MySQLi. For example
 - ```
$stmt = $dbConnection->prepare('SELECT * FROM professors_list
WHERE name = ?'); $stmt->bind_param('s', $name); // 's' specifies the
variable type => 'string' $stmt->execute(); $result = $stmt->get_result();
while ($row = $result->fetch_assoc()) { // Do something with $row }
```

## **Self-check: Adherence to original Non-functional specs**

### **Security:**

1. Login shall be mandatory for student and professor -DONE
2. Username shall be the user's registered email - DONE
3. Password shall be encrypted before saving it in the database - DONE
4. User's session shall be ended after 30 minutes of inactivity - ON TRACK
5. User's session shall only be ended by code design - DONE
6. This site shall not accept third-party cookies - DONE

### **Audit:**

1. The site admin shall be the only person authorized to delete users - DONE
2. Users shall not be allowed to modify any web configuration files - DONE
3. Users shall not be allowed to login into the admin page - DONE
4. Unregistered users shall not be able to access anything other than the login/signup page - DONE

### **Performance:**

1. The site loading time shall be less than 1 second for all the screens - DONE
2. "Search" shall be executed in a background thread for improving performance - DONE
3. Query shall be executed in a background thread for improving performance - DONE

### **Capacity:**

1. The total data storage allowed by the web site shall not exceed 80 % of the server capacity for this site - DONE
2. The web site shall be prepared to support scalability for adding future new features - DONE
3. The web site shall be capable to handle at least 50 users - DONE

### **Reliability:**

1. Downtime for maintenance shall be less than 1 hour per month - DONE
2. Downtime for maintenance shall not affect the main functionality of the site - DONE

3. In all cases, downtime for maintenance shall be informed to the users either by email - DONE
4. In all cases, downtime for maintenance shall be informed to the users by publishing an announcement in a visible place of the main page. - DONE

Recovery:

1. In a total failure case, the whole site should be put down to revision - DONE
2. If broken, the mean time to recovery shall not exceed one day - DONE

Data Integrity:

1. Database tables shall be backed up every week - DONE
2. Admin shall be able to execute a recovery when needed - DONE
3. Images size shall be limited up to 1 megabyte - ON TRACK
4. Images shall be uploaded in the correct format (jpg or jpeg) - ON TRACK

Compatibility:

1. The site shall be compatible with the last version of Safari browser - DONE
2. The site shall be compatible with the last version of Firefox browser - DONE
3. The site shall be compatible with the last version of Chrome browser - DONE
4. The site shall be compatible with at least an old version of all the browsers listed above - DONE
5. Third party applications shall not be able to modify any content that may affect the site compatibility - DONE
6. The site shall be ready to support with any or minimal changes any other compatibility that may be added in future versions - DONE
7. The site should be compatible to escalate to new databases - DONE

Conformance with Coding Standards:

1. Architecture and design standards shall meet all the requirements listed under the High-Level Architecture section of this document - DONE
2. Only working code that meets all the code standards shall be submitted to the project repository - DONE
3. Any working code shall be tested and debugged before being considered to be working code - DONE

4. Any internal errors or exceptions returned by the code shall be stored in a log - DONE
5. Any error that may affect the functionality of the site shall be reported to the user - DONE
6. Any error shall be handled in a way that does not affect the functionality of the site - DONE
7. This site shall not be launched without all the priority one features finished and tested - DONE
8. This site shall be tested and debugged as a whole 2 weeks before the delivery due date - DONE

Look and Feel Standards:

1. The application and its layouts shall look professional - DONE
2. The site shall be enough simple to handle by all the parties involved - DONE
3. Elements on the screen shall have the correct density to meet the compatibility standard of the browsers - DONE
4. Elements on the screen shall have the correct size to meet the compatibility standard with all mobile devices - DONE
5. Elements on the screen shall have rich and beautiful colors for user delight - DONE
6. The site shall be able to work correctly without mouse interaction - DONE
7. The site shall be able to work correctly without keyboard interaction - DONE
8. Elements in screen shall be resized automatically without user interaction when being loaded in all the different platforms supported by the site - DONE

Internationalization / Localization Requirements:

1. The default language of this site shall be English - DONE
2. The site shall support scalability to add more supported languages in future versions -DONE

## Project Screenshots

The screenshot shows the Gator Dater homepage. At the top left is the Gator Dater logo and name. At the top right are links for "Henry Jones" and "Logout". On the left side, there is a sidebar with three items: "Account" (represented by a user icon), "Calendar" (represented by a calendar icon), and "Help" (represented by a question mark icon). The main content area is titled "Professor Search" and contains a search bar with the placeholder text "Enter Professor Name".

This screenshot is identical to the one above, but it shows the results of a search. The search bar now contains the text "jacob", and below it, a card displays the result "Jacob Smith".

Gator Dater

Henry Jones ,Logout

today < >

December 2019

week month

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
| 1   | 2   | 3   | 4   | 5   | 6   | 7   |
| 8   | 9   | 10  | 11  | 12  | 13  | 14  |
| 15  | 16  | 17  | 18  | 19  | 20  | 21  |
| 22  | 23  | 24  | 25  | 26  | 27  | 28  |

Gator Dater

Henry Jones ,Logout

today < >

Dec 1 – 7, 2019

week month

|      | Sun 12/1 | Mon 12/2 | Tue 12/3 | Wed 12/4 | Thu 12/5 | Fri 12/6 | Sat 12/7 |
|------|----------|----------|----------|----------|----------|----------|----------|
| 6am  |          |          |          |          |          |          |          |
| 7am  |          |          |          |          |          |          |          |
| 8am  |          |          |          |          |          |          |          |
| 9am  |          |          |          |          |          |          |          |
| 10am |          |          |          |          |          |          |          |
| 11am |          |          |          |          |          |          |          |
| 12pm |          |          |          |          |          |          |          |
| 1pm  |          |          |          |          |          |          |          |



Gator Dater

Henry Jones ,Logout



Account



Calendar



Help

today &lt; &gt;

Dec 1 – 7, 2019

week month

|      | Sun 12/1 | Mon 12/2 | Tue 12/3 | Wed 12/4 | Thu 12/5 | Fri 12/6 | Sat 12/7 |
|------|----------|----------|----------|----------|----------|----------|----------|
| 6am  |          |          |          |          |          |          |          |
| 7am  |          |          |          |          |          |          |          |
| 8am  |          |          |          |          |          |          |          |
| 9am  |          |          |          |          |          |          |          |
| 10am |          |          |          |          |          |          |          |
| 11am |          |          |          |          |          |          |          |
| 12pm |          |          |          |          |          |          |          |
| 1pm  |          |          |          |          |          |          |          |



Gator Dater

Henry Jones ,Logout



Account



Calendar



Help

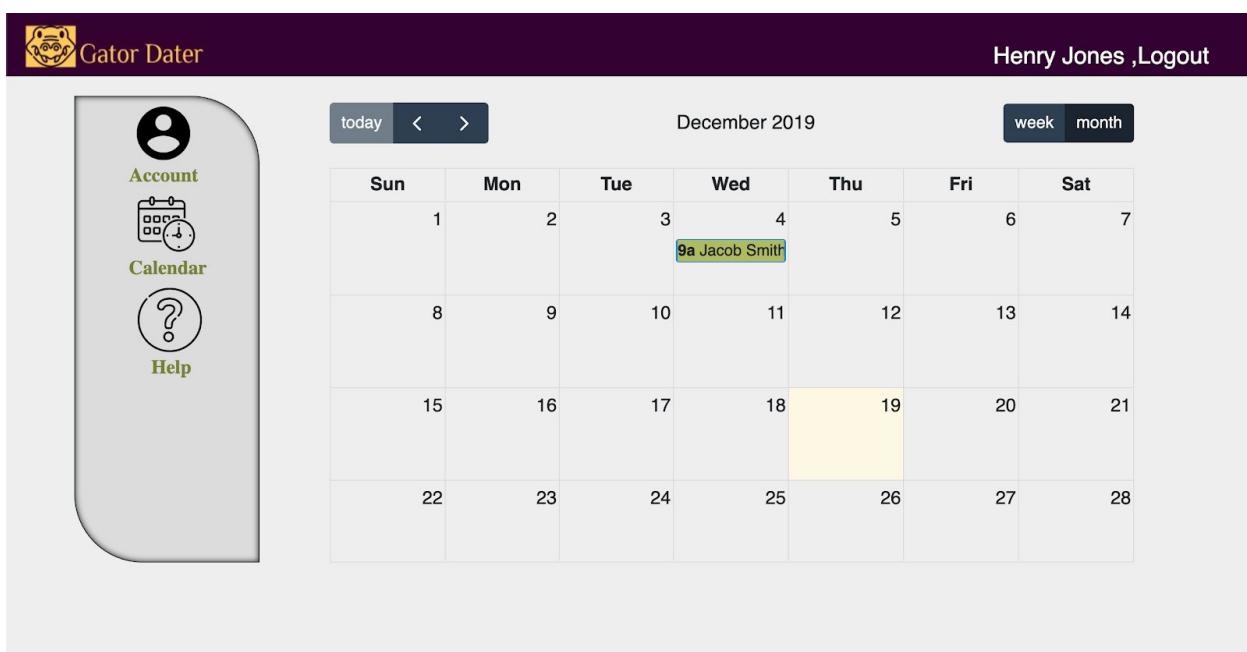
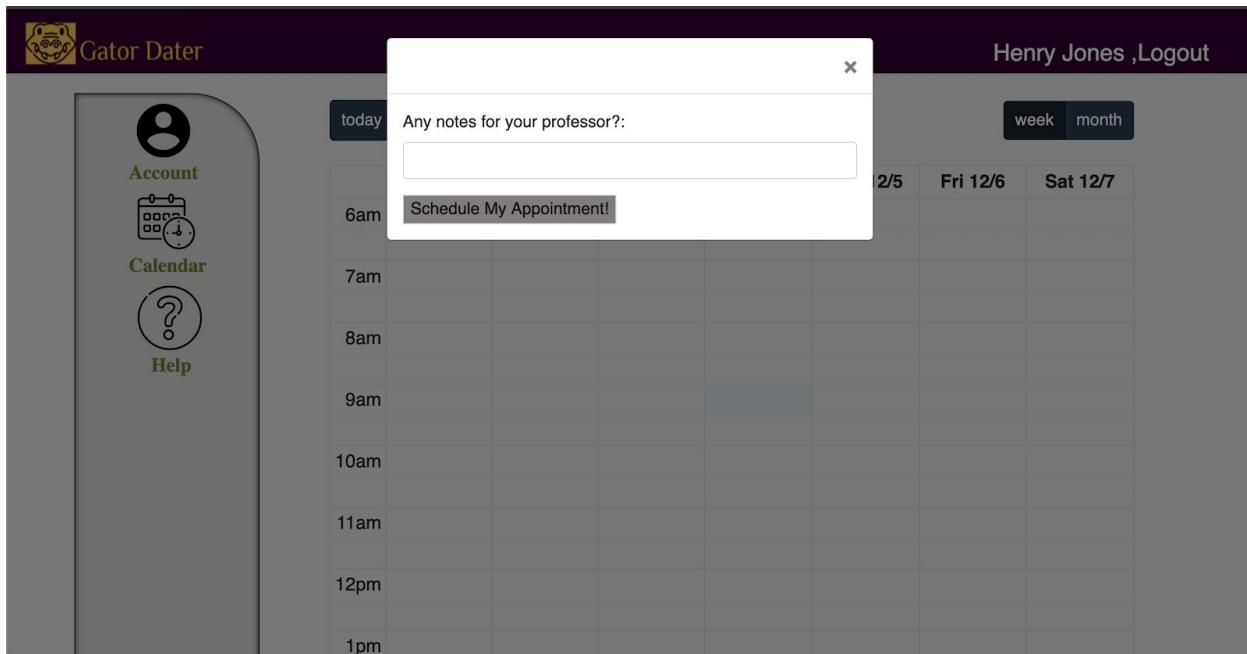
today

Any notes for your professor?:

Schedule My Appointment!

week month

|      |  |  |  |  |  |  |  |
|------|--|--|--|--|--|--|--|
| 6am  |  |  |  |  |  |  |  |
| 7am  |  |  |  |  |  |  |  |
| 8am  |  |  |  |  |  |  |  |
| 9am  |  |  |  |  |  |  |  |
| 10am |  |  |  |  |  |  |  |
| 11am |  |  |  |  |  |  |  |
| 12pm |  |  |  |  |  |  |  |
| 1pm  |  |  |  |  |  |  |  |



 Gator Dater

Jacob Smith ,Logout

today < >

December 2019

week month

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
| 1   | 2   | 3   | 4   | 5   | 6   | 7   |
| 8   | 9   | 10  | 11  | 12  | 13  | 14  |
| 15  | 16  | 17  | 18  | 19  | 20  | 21  |
| 22  | 23  | 24  | 25  | 26  | 27  | 28  |

 Gator Dater

Henry Jones ,Logout

today < >

December 2019

week month

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
| 1   | 2   | 3   | 4   | 5   | 6   | 7   |
| 8   | 9   | 10  | 11  | 12  | 13  | 14  |
| 15  | 16  | 17  | 18  | 19  | 20  | 21  |
| 22  | 23  | 24  | 25  | 26  | 27  | 28  |

Gator Dater

Henry Jones ,Logout

today < >

December 2019

week month

| Sun | Mon | Tue | Wed          | Thu | Fri            | Sat |
|-----|-----|-----|--------------|-----|----------------|-----|
| 1   | 2   | 3   | 4<br>9a BUSY | 5   | 6              | 7   |
| 8   | 9   | 10  | 11           | 12  | 13             | 14  |
| 15  | 16  | 17  | 18           | 19  | 20<br>12p BUSY | 21  |
| 22  | 23  | 24  | 25           | 26  | 27             | 28  |

Gator Dater

Jacob Smith ,Logout

today < >

December 2019

week month

| Sun | Mon | Tue | Wed                 | Thu | Fri                 | Sat |
|-----|-----|-----|---------------------|-----|---------------------|-----|
| 1   | 2   | 3   | 4<br>9a Henry Jones | 5   | 6                   | 7   |
| 8   | 9   | 10  | 11                  | 12  | 13                  | 14  |
| 15  | 16  | 17  | 18                  | 19  | 20<br>12p Henry Jon | 21  |
| 22  | 23  | 24  | 25                  | 26  | 27                  | 28  |

## **Screenshots Key DB Tables**

**phpMyAdmin**

Server: localhost:3306 » Database: db01 » Table: appointments\_list

Browse Structure SQL Search Insert Export Import Operations Triggers

Showing rows 0 - 1 (2 total, Query took 0.0002 seconds.)

ELECT \* FROM `appointments\_list`

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Options professor\_id student\_id appointment\_id start\_event end\_event notes student\_name allDay professor\_name busy

| professor_id | student_id | appointment_id | start_event         | end_event           | notes | student_name | allDay | professor_name | busy |
|--------------|------------|----------------|---------------------|---------------------|-------|--------------|--------|----------------|------|
| 29           | 28         | 28             | 2019-12-04T09:00:00 | 2019-12-04T09:30:00 | None  | Henry Jones  | 0      | Jacob Smith    | BUSY |
| 29           | 28         | 29             | 2019-12-20T12:00:00 | 2019-12-20T12:30:00 | none  | Henry Jones  | 0      | Jacob Smith    | BUSY |

Check all With selected: Edit Copy Delete Export

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

**phpMyAdmin**

Server: localhost:3306 » Database: db01 » Table: users\_list

Browse Structure SQL Search Insert Export Import Operations Triggers

Showing rows 0 - 1 (2 total, Query took 0.0002 seconds.)

SELECT \* FROM `users\_list`

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options uid email name type major password username

| uid | email            | name        | type      | major            | password                                                            | username |
|-----|------------------|-------------|-----------|------------------|---------------------------------------------------------------------|----------|
| 28  | hjones@gmail.com | Henry Jones | Student   | Computer Science | \$2y\$10\$uacv0J9Uk3jdAjA1EghSeS2mnjCNw9MegvnBZICVl... HenryJones   |          |
| 29  | jsmith@gmail.com | Jacob Smith | Professor | Computer Science | \$2y\$10\$w3Si7pLN8nePkGnHL5GdjuMnSpSzYOSKsRBnw13fDi0... JacobSmith |          |

Check all With selected: Edit Copy Delete Export

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

35.203.131.184/phpmyadmin/sql.php?server=1&db=db01&table=users\_list&pos=0&token=98990f091d922db586e96490f756104e

# Trello

The image shows a Trello board titled "GatorDater". The board has four columns: "Task", "In Progress", "Finished Milestone", and "Finished Projected/Component".

- Task:**
  - Pushing everything to Github (1 comment)
  - + Add another card
- In Progress:**
  - Final Project
  - Presentation
  - Milestone 5 (1 comment, 2 tasks)
  - + Add another card
- Finished Milestone:**
  - Milestone 4 V2 (3 comments, 1 task, 6/6 done)
  - Milestone 4 (1 comment, 1 task, 6/6 done)
  - Milestone 3 Feedback (1 comment, 2 tasks)
  - Milestone 2 V2 (1 comment, 1 task, 9/9 done)
  - Milestone 2 (1 comment, 1 task, 9/9 done)
  - Milestone 1 V2 (2 comments, 2 tasks)
  - Milestone 1 (1 comment, 1 task, 8/8 done)
  - + Add another card
- Finished Projected/Component:**
  - Prototype (3 comments, 2/2 done)
  - UI mock-ups
  - + Add another card

## Team Member Contributions

TV

Tejas Vajjhala <tvajjhala@gmail.com>



Thu 12/19/2019 4:22 AM

To: Sanil <srijal0509@gmail.com>

Cc: Aaron Li; t.zhang70@gmail.com

Hi Team,

Here are some of my contributions:

- Fixed any errors with website code (HTML/PHP/JS)
- Full stack development
- Created Server
- Transferred files from github to server
- Created DB
- Milestone documentation
- Re-Coded calendar for backend integration
- pushed files that were sent to me directly by team members to the repo.

I have made 30 commits to github and that number will continue to go up until the project is completed.

Thanks for the great semester,

Tejas Vajjhala

...

## CSC 648 GatorDater Contributions

TZ

Thomas Zhang <t.zhang70@gmail.co  
m>



Thu 12/19/2019 12:22 AM  
tvajjhala@gmail.com; srijal0509@gmail.com; Aaron Li ▾

### My Contributions:

- A) I contributed ideas to the milestone docs such as priorities, main use cases, functional requirements, non functional requirements, UI mock-ups and storyboard, QA Test Plan. I coded some of the front-end aspect of the student side of things, while maintaining the CSS to be relatively the same and unbroken across student and professor.
- B) I made about 8 commits to the development server, but a lot of them were sent by email to my team leader.



Sanil <srijal0509@gmail.com>

Thu 12/19/2019 2:47 AM

tvajjhala@gmail.com; Aaron Li; t.zhang70@gmail.com



Hi Team,

My part of Contributions:

- Created Team page
- Assisted team in parts of Milestone documentation
- Created some part of the mockups on the word to get started
- Designed and created whole front end on the Figma
- Designed and developed the front end

I have not made any submission onto the GitHub till now but like everyone I been sending all of my files to the Team Lead directly for approval and checkmark.

Good to be the part of this Team. Thanks,

Regards

Sanil Rijal

Team 204



Aaron Li

Thu 12/19/2019 12:17 AM



To: tvajjhala@gmail.com

Cc: srijal0509@gmail.com; t.zhang70@gmail.com ✉

Hi Team,

My contributions:

- Documentations of Milestones
- Creating ERD and DB Model
- Backend for Registration/Login/Update
- Wrote SQL queries and relationships

I made 3 submissions onto the GitHub dev branch but majority of files were sent to team lead directly.

It was great working with you guys

Best Regards,

Aaron Li

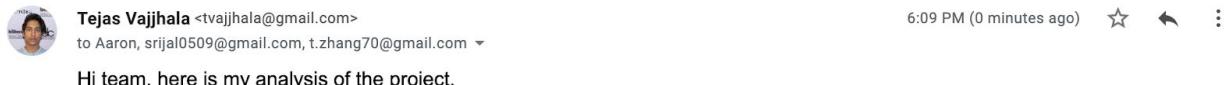
## Post Analysis

Throughout this project, I have learned quite a lot. The main challenges we faced was lack of experience in a group setting. This lead to have some conflicts with pushing updated code and merging them together. Sometimes team members would work on updates and features, but not use the latest version of the file. This leads to merge conflicts as one of us had to manually transpose the updated code into the updated file. Furthermore, I had to learn to work with an external calendar (fullcalendar) and implement it into our product. I personally am not a frontend developer, but I was able to learn it and implement most functions and have a working scheduling feature. But due to my inexperience with front end web development, I had struggled with some implementations.

What I would do next time, is spend more time explaining how we can work better in parallel. By doing this, we would have reduced the time spent on fixing merge conflicts, and would not have had to rush at the end of the semester to get a working project submitted.

But overall, I think everyone had a great experience this semester. We learned a lot regarding how software is developed in the real world, and found out how difficult it is to meet priorities and timeline requirements. It was a lot of fun creating something that solves a problem, and I'm sure we will continue improving our product in the future as it will be very helpful if it is implemented throughout the campus.

### Screenshot:



Throughout this project, I have learned quite a lot. The main challenges we faced was lack of experience in a group setting. This lead to have some conflicts with pushing updated code and merging them together. Sometimes team members would work on updates and features, but not use the latest version of the file. This leads to merge conflicts as one of us had to manually transpose the updated code into the updated file. Furthermore, I had to learn to work with an external calendar (fullcalendar) and implement it into our product. I personally am not a frontend developer, but I was able to learn it and implement most functions and have a working scheduling feature. But due to my inexperience with front end web development, I had struggled with some implementations.

What I would do next time, is spend more time explaining how we can work better in parallel. By doing this, we would have reduced the time spent on fixing merge conflicts, and would not have had to rush at the end of the semester to get a working project submitted.

But overall, I think everyone had a great experience this semester. We learned a lot regarding how software is developed in the real world, and found out how difficult it is to meet priorities and timeline requirements. It was a lot of fun creating something that solves a problem, and I'm sure we will continue improving our product in the future as it will be very helpful if it is implemented throughout the campus.

Thanks for the great experience,  
Tejas Vaijhala