



Final Technical Design

by Team Maverick

Backend (DB): Darius Clark, Nick Gangloff, Tyler Jackson

Frontend (GUI): Bre'Shard Busby, Jessica Yeh, Story Zanetti

Table of Contents

| | |
|---|-------|
| Introduction of Project and Team | 3 |
| Software Features | 3 |
| Use Case Diagram | 4 |
| Software Architecture | 5-6 |
| <i>Technologies and Libraries Used</i> | 6 |
| Database Model | 7-8 |
| Screenshots of Web Application | 9-29 |
| <i>Home Screen and Searching Features</i> | 9-11 |
| <i>Tutor Profile Page and Profile Editing</i> | 12-15 |
| <i>Navigation Bars</i> | 16-17 |
| <i>Register and Edit Account</i> | 18-19 |
| <i>History</i> | 20-26 |
| <i>Tutor Applications</i> | 27-28 |
| <i>About Us and Contact Us</i> | 29 |
| Screenshots of Android Application | 30-35 |
| Testing | 36 |
| Team Reflection | 37-38 |
| <i>Out-of-class Learning</i> | 37-38 |
| <i>Overall Reflection</i> | 38 |
| Wrap Up | 39 |
| Appendix: Data Dictionary | 40-44 |

Introduction of Project and Team

The A-LEC provides tutoring to SMU students, but little information is available online for which subjects are tutored and when tutors are available. Our website allows students to search for tutors. Our website also allows students to rate and comment on tutors and to view a history of all the meetings they've attended at the A-LEC. These features will allow the A-LEC to function more seamlessly and to foster a productive environment with the tutors and the students.

Mustang Tutors was developed by Team Maverick, a six person team. Bre'Shard Busby, Jessica Yeh, and Story Zanetti developed the web UI and the Android application, while Darius Clark, Nick Gangloff, and Tyler Jackson developed the RESTful API. Tyler Jackson also developed a prototype iOS application to accompany the website and Android application.

Software Features

Visitor

- Search for available tutors based on a variety of specifications
- View tutor profile pages
- Create an account on Mustang Tutors using the student's SMU ID
- View tutoring history of a student

Student

- All functionality of a visitor
- Log in onto created account
- Edit account information
- Rate and comment on tutor profile pages
- View history of tutoring sessions the student attended
- Grant access of history to a third party
- Apply to be registered as a tutor

Tutor

- All functionality of a student
- Complete "student session information form" for each tutoring session
- View history of sessions with students the he/she tutored
- Toggle availability status

Admin

- All functionality of a student
- Update tutor's regular hours (on Tutor Profile)
- Update list of courses tutors are authorized to tutor (on Tutor Profile)
- Toggle tutor active status (whether still with the company or not)
- Approve/deny tutor applications

Use Case Diagram

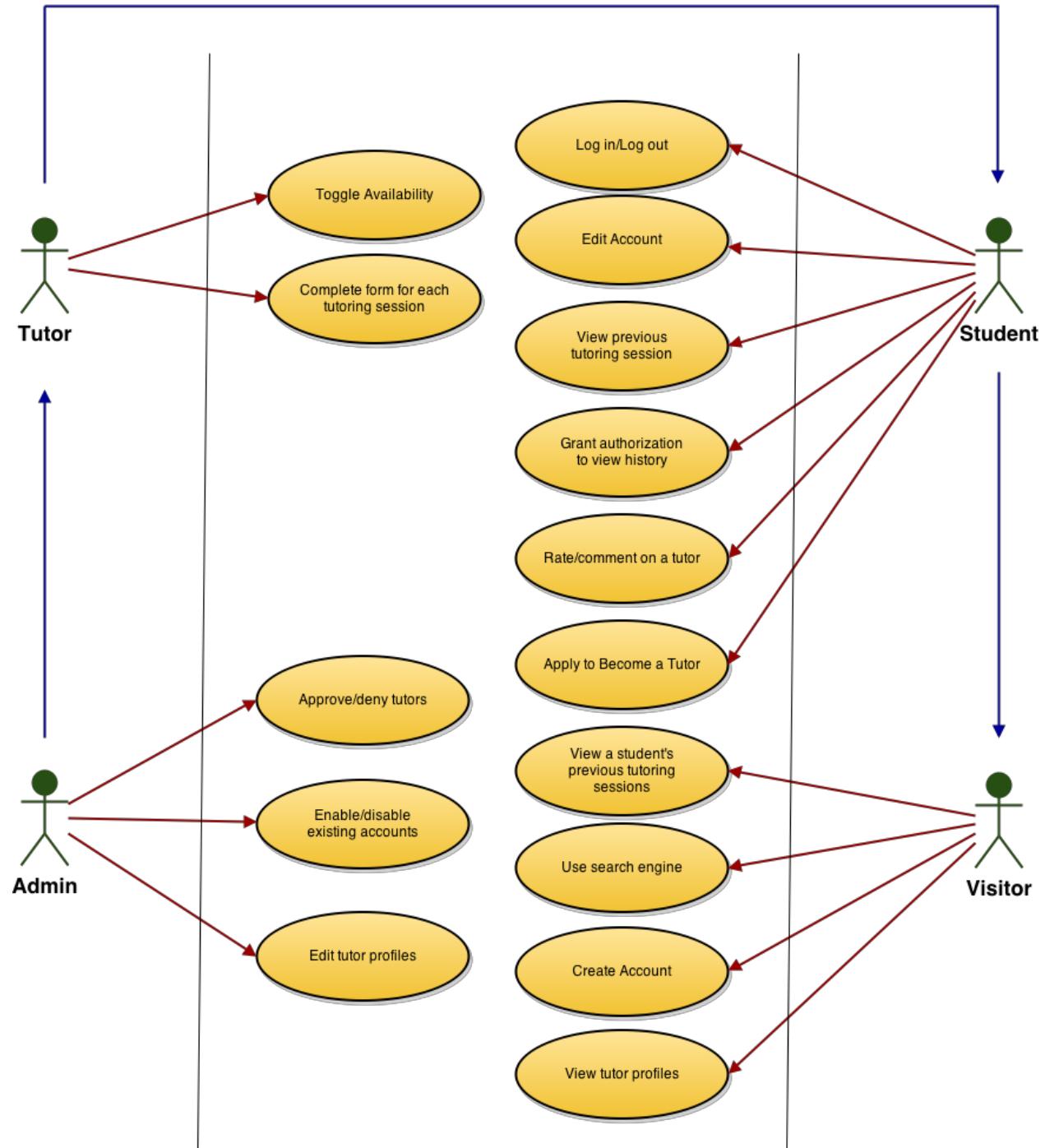


Figure 1: Use Case Diagram

Software Architecture



Legend:

- = Response
- = Request

Image Sources

http://www.unikstickers.com/image/cache/data/stickers/jquery/jquery_zummer_sh-900x600.png
http://www.w3.org/html/logo/downloads/HTML5_Logo_512.png
<http://marketplace.serviceosocket.com/static/images/logos/less3Logo.png>
http://html5live.com/ganesh_logos/s3Logo.png
<https://www1.cbsnews.com/pictures/10-best-new-smartphones-for-2013/motorola-droid-razr-m/>
http://www.appian.com/blog/wp-content/uploads/2011/04/800px_Amazon_Web_Services_logo.svg.png
<http://www.con-drug.net/wp-content/uploads/2013/01/Server-icon-256.png>
https://iconfinder.com/iconset/database/PNG/512/Database_4.png
<http://www.softicons.com/free-icons/computer-icons/mobile-phone-icon-by-lrj-jagineni/nfc-desire-icon.html>
<http://icons.iconarchive.com/icons/iconsoft/elite-n-desktop/256/My-Computer-No-Wallpaper-icon.png>
<http://standardista.com/velocity/images/android.png>

Figure 2: Software Architecture Diagram

Technologies and Libraries Used

jQuery

<http://jquery.com>

This is a Javascript library that simplifies and minimizes Javascript code. Unlike pure Javascript, it works the same across browsers. All of our Javascript files are written using jQuery, including the XML HTTP requests.

hoverIntent jQuery Plugin

<http://cherne.net/brian/resources/jquery.hoverIntent.html>

This plugin provides an alternative to jQuery's hover method. When a user hovers over an element that has a normal jQuery hover event binded to it, the event fires immediately. HoverIntent is different in that it waits for the user's mouse to stop on the element before firing the event. We use hoverIntent on the tutor info boxes that slide left/right when a user hovers over tutors on the search page on the website.

moment.js Javascript Library

<http://momentjs.com>

Javascript does not provide a very developer-friendly built-in Date class. To simplify manipulation of time and date strings, we use moment.js, which is “a Javascript date library for parsing, validating, manipulating, and formatting dates.”

CSS Toggle Switch

<http://ghinda.net/css-toggle-switch>

Since HTML5 does not have a built-in input element that functions like the toggle switch found in the standard Android library, we use this CSS Toggle Switch that works like a checkbox with styling to appear like a toggle switch.

Laravel

<http://laravel.com/>

This is the PHP framework used mainly to handle the routing of our REST API and connecting to our database. Laravel is very versatile due to the plethora of features it contains, which allows it to be used for projects of any size. We didn't use many of Laravel's features, such as the Query Builder and Blade Templating, since they abstract away most of what we learned in class. We did, however, make use of Laravel's mail API and its HTTP Input abstraction to avoid some of the more tedious caveats of PHP.

Database Model

| Table | Attribute | Type | Constraints |
|-----------------|--------------------|----------------------|-------------|
| Users | User_ID | INT (Auto Increment) | Primary Key |
| | SMU_ID | INT | Unique Key |
| | First_Name | VARCHAR(30) | |
| | Last_Name | VARCHAR(30) | |
| | Email | VARCHAR(30) | |
| | Password | VARCHAR(200) | |
| | Codeword | VARCHAR(30) | |
| | Available | INT | |
| | Active | INT | |
| | Tutor | INT | |
| | Admin | INT | |
| Courses | Course_ID | INT (Auto Increment) | Primary Key |
| | Subject | VARCHAR(30) | |
| | Course_Number | INT | |
| | Course_Name | VARCHAR(30) | |
| Courses_Tutored | Courses_Tutored_ID | INT (Auto Increment) | Primary Key |
| | Course_ID | INT | Foreign Key |
| | User_ID | INT | Foreign Key |
| Schedule | Schedule_ID | INT (Auto Increment) | Primary Key |
| | User_ID | INT | Foreign Key |
| | Day | INT | |
| | Start_Time | TIME | |
| | End_Time | TIME | |

Figure 3: Table of Entities

Database Model Continued

| Table | Attribute | Type | Constraints |
|--------------|----------------|----------------------|-------------|
| Ratings | Rating_ID | INT (Auto Increment) | Primary Key |
| | User_ID | INT | Foreign Key |
| | Tutor_User_ID | INT | Foreign Key |
| | Rating | INT | |
| Comments | Comment_ID | INT (Auto Increment) | Primary Key |
| | User_ID | INT | Foreign Key |
| | Tutor_User_ID | INT | Foreign Key |
| | Comment | VARCHAR(500) | |
| Records | Timestamp | DATETIME | |
| | Record_ID | INT (Auto Increment) | Primary Key |
| | User_ID | INT | Foreign Key |
| | Tutor_User_ID | INT | Foreign Key |
| Applications | Course_ID | INT | Foreign Key |
| | Date | DATE | |
| | Start_Time | TIME | |
| | End_Time | TIME | |
| Applications | Summary | VARCHAR(255) | |
| | Application_ID | INT (Auto Increment) | Primary Key |
| | User_ID | INT | Foreign Key |
| | Pending | INT | |

Figure 3: Table of Entities

Screenshots of Web Application

Home Screen and Searching Features

The Home Screen

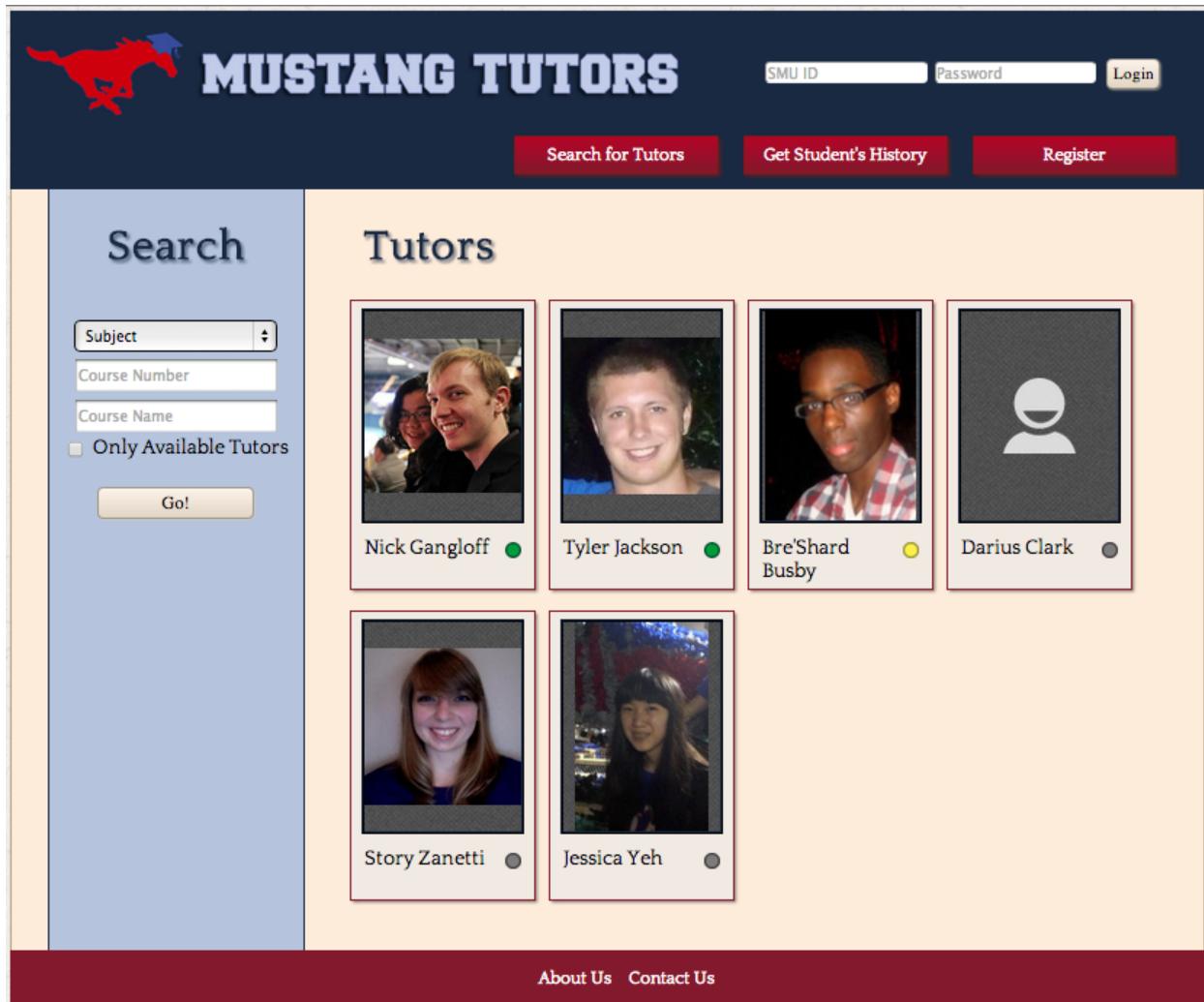


Figure 4: Home Screen

Figure 4 above is the home page of Mustang Tutors. Whether or not a user is logged in, he can search for tutors using the course subject, number, or name. Additionally, the user can choose to see only tutors who are currently available at the time of his search. By default, all tutors in the system are listed in the tutors section, sorted first by availability and then in alphabetical order.

The Hover View

The screenshot shows the Mustang Tutors website. At the top, there is a navigation bar with a red horse logo, the text "MUSTANG TUTORS", and login fields for "SMU ID" and "Password" with a "Login" button. Below the navigation bar are three buttons: "Search for Tutors", "Get Student's History", and "Register". The main content area has a light blue sidebar on the left labeled "Search" containing input fields for "Subject", "Course Number", "Course Name", and a checked checkbox for "Only Available Tutors", followed by a "Go!" button. The main body is titled "Tutors" and displays four tutor profiles in cards. The first card, for Nick Gangloff, is expanded, showing a photo of him smiling, his name, an "Available" status indicator, an average rating of 3 stars, and a green dot indicating availability. The other three cards show photos, names, and gray dots. At the bottom, there is a footer bar with links for "About Us" and "Contact Us", and a URL "occult.us/tutor.html?user_id=2".

Figure 5: Home Screen Hover

When the user hovers his mouse over a tutor, the tutor's information is expanded, as shown above in *Figure 5*.

Searching With Criteria

The screenshot shows the Mustang Tutors website interface. At the top, there is a logo of a red Mustang and the text "MUSTANG TUTORS". A navigation bar includes "Search for Tutors", "Get Student's History", and "Register". Below the navigation bar, there is a search form on the left labeled "Search" with fields for "Subject" (CSE), "Course Number" (3345), "Course Name", and a checkbox for "Only Available Tutors". A "Go!" button is present. To the right, under the heading "Tutors", three tutor profiles are displayed in cards: Bre'Shard Busby (available), Jessica Yeh (busy), and Story Zanetti (available). At the bottom, there are links for "About Us" and "Contact Us".

Figure 6: Normal Search

Figure 6 shows a search result where the list of tutors has been filtered. Only tutors who are approved to tutor the course designated in the search criteria are shown.

The screenshot shows the Mustang Tutors website interface, similar to Figure 6 but with additional features for administrators. At the top, there is a logo of a red Mustang and the text "MUSTANG TUTORS". A navigation bar includes "Applications" (with a count of 1), "Search for Tutors", "Student History", and two status buttons: "Available" and "Busy". Below the navigation bar, there is an "About Us" link. On the left, there is a "Search" section with fields for "First Name" (Jessica) and "Last Name" (Yeh). The "Subject" dropdown is set to "CSE". The "Only Available Tutors" checkbox is checked. A "Go!" button is present. To the right, under the heading "Tutors", one tutor profile is displayed in a card: Jessica Yeh (busy). At the bottom, there are links for "About Us" and "Contact Us".

Figure 7: Admin Search

As Figure 7 shows, Admin users have the additional option of searching by first name and last name.

Tutor Profile Page and Profile Editing

View Tutor Profile

The screenshot shows the Mustang Tutors website interface. At the top, there is a navigation bar with a logo of a running horse, the text "MUSTANG TUTORS", and a welcome message "Welcome, Nick! ⚡". Below the navigation bar, there are buttons for "Search for Tutors", "Student History", "Available" (which is highlighted in red), and "Busy".

The main content area features a profile picture of Story Zanetti, her name "Story Zanetti", and a status indicator "Unavailable". It also displays her average rating (4 stars based on 1 rating) and a field for users to enter their own rating (4 stars).

Two boxes below the profile provide detailed information: "Courses Taught" (listing PHIL 1318, CSE 2341, CSE 1342, and CSE 3345) and "Regular Hours" (listing hours for Sunday through Saturday).

A "Comments" section follows, containing a large text input field, an "Add Comment" button, and two comment entries. Each comment includes a timestamp (Posted: 2014-03-28 10:42:22 AM), a rating (4 stars), and a "Tutor rating" field.

At the bottom of the page is a dark red footer bar with links for "About Us" and "Contact Us".

Figure 8: Tutor profile page

By clicking on a tutor on the search page (*Figure 4*), a user is taken to the tutor's profile page, shown in *Figure 8*, where they can view detailed information about a tutor, including what courses they tutor, their regular hours, their rating, and their comments. A logged in user can make ratings and leave comments.

Rate Tutor

The screenshot shows a user profile for Story Zanetti. At the top, there's a red silhouette of a running horse logo, the text "MUSTANG TUTORS", and a welcome message "Welcome, John! ⚡". Below the header are three buttons: "Search for Tutors", "Student History", and "Become a Tutor". The main content area features a photo of Story Zanetti, her name, and a status box saying "Unavailable". Her average rating is shown as 4.5 stars based on 2 ratings. A "Your rating:" field shows 5 stars with a pencil icon. A modal dialog box titled "Rate This Tutor!" is open, displaying a grid of 10 stars where the first 5 are blue and the last 5 are grey. To the right of the grid is the text "Not so good". A "Cancel" button is at the bottom of the dialog. To the left, under "Courses Taught", are listed PHIL 1318: Contemporary Moral Problems and CSE 2341: Data Structures. To the right, under "Regular Hours", are listed times for Monday through Friday.

Figure 9: Rating a tutor

By clicking on the pencil icon next to “Your rating,” as shown in *Figure 9*, a user can either add a rating for the tutor or modify a rating he previously made. This new rating is incorporated into the average rating.

Admin Profile View

The screenshot shows the 'Mustang Tutors' website interface. At the top, there's a red header bar with a red horse logo on the left, the text 'MUSTANG TUTORS' in large white letters, and 'Welcome, Story!' with a gear icon on the right. Below the header, a navigation bar has four items: 'Applications' (with a value of 1), 'Search for Tutors', 'Student History', and two buttons for 'Tutor Status': 'Available' (highlighted in red) and 'Busy'. The main content area has a light blue background. It features a profile picture of a woman with long brown hair, identified as 'Story Zanetti'. To her right, the word 'Available' is displayed in a green button. Below the profile picture, the text 'Average rating:' is followed by five blue stars, with the small text 'Based on 2 ratings' underneath. In the bottom right corner of this section, there's a yellow 'Edit Profile' button. Below this, the page is divided into two orange-colored sections: 'Courses Taught' on the left and 'Regular Hours' on the right.

Figure 10: Admin view of Tutor Profile

Admin Edit View

The screenshot shows the 'Admin Edit View' of a tutor profile. At the top, there's a logo of a red mustang and the text 'MUSTANG TUTORS'. To the right, it says 'Welcome, Story!'. Below the header, there are tabs for 'Applications' (with 1 notification), 'Search for Tutors', 'Student History', 'Available' (which is selected), and 'Busy'. A large photo of a smiling woman with brown hair is on the left. Her name, 'Story Zanetti', is displayed in large blue text. To her right, there are two buttons: 'Available' (green) and 'Enabled' (red). Below the photo, it says 'Average rating: ★★★★☆' based on 2 ratings. At the bottom right of the profile section is a 'Tutor's Profile' button. The main content area is divided into two sections: 'Courses Taught' and 'Regular Hours'. The 'Courses Taught' section contains a list of checked courses: PHIL 1318, CSE 2341, CSE 1342, and CSE 3345. A dropdown menu is open, showing additional course options: Choose a course, CSE 1342 Programming Concepts, CSE 2341 Data Structures, CSE 3345 Graphical User Interfaces, EE 1301 Modern Electronic Technology, MATH 1338 Calculus II, PHIL 1318 Contemporary Moral Problems, and STAT 4340 Statistics for Engineers. There are 'Reset' and 'Save' buttons at the bottom. The 'Regular Hours' section lists days of the week with start and end times: Sunday (12:00 AM to 12:00 AM), Monday (12:00 PM to 02:00 AM), Tuesday (12:00 AM to 12:00 AM), Wednesday (12:00 AM to 12:00 AM), Thursday (04:00 PM to 07:30 PM), Friday (12:00 AM to 12:00 AM), and Saturday (12:00 AM to 12:00 AM). There are also 'Reset' and 'Save' buttons at the bottom.

Figure 11: Admin edit of Tutor Profile page

An Admin is able to edit a tutor's information. When an Admin visits a tutor's profile, he sees the view in *Figure 10*, which is the same view that other visitors see with the addition of the "Edit Profile" button. Clicking this button directs the Admin to the page shown in *Figure 11*. An Admin is then able to "Enable" or "Disable" the tutor. When a tutor is disabled, he does not appear anywhere on the website as a tutor, though his information is still saved. Admins are the only ones able to see disabled tutors (they still show up in the search for admins). Additionally, an Admin is able to edit the courses a tutor can teach and the regular hours a tutor works.

Navigation Bars

Visitor



Figure 12: Visitor Navigation Bar

Student



Figure 13: Student Navigation Bar

Tutor



Figure 14: Tutor Navigation Bar

Admin



Figure 15: Admin Navigation Bar

Toggle Availability

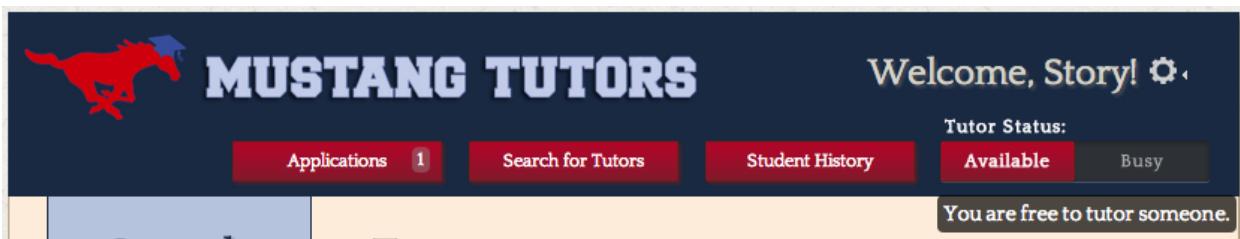


Figure 16: Tutor availability toggled to “Available”

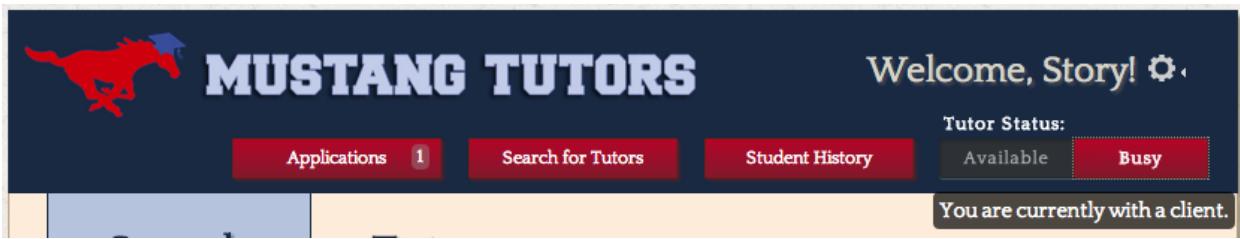
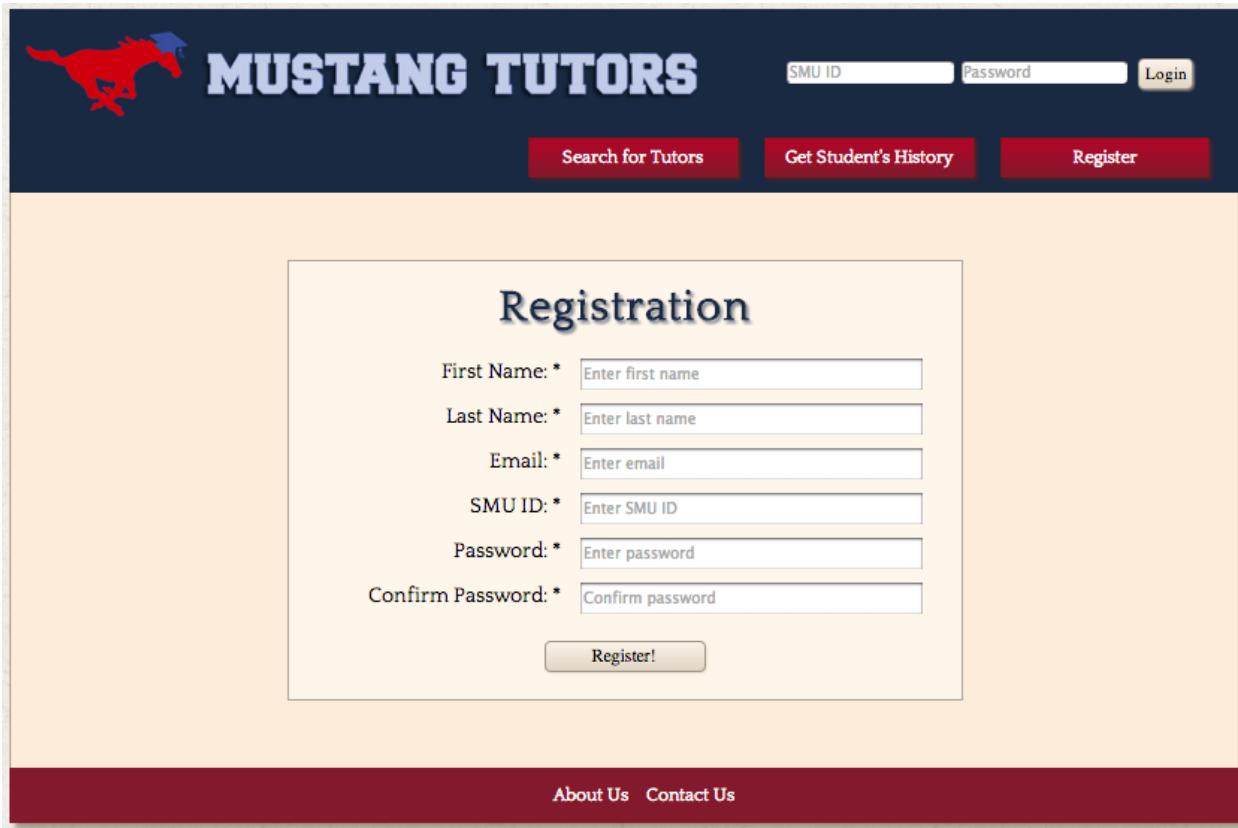


Figure 17: Tutor availability toggled to “Busy”

The navigation bar changes depending on who the user is. A user who is not logged in can only search for tutors and student histories, as shown in *Figure 12*. A logged in user with no special permissions has the navigation shown in *Figure 13*. A tutor’s navigation bar, shown in *Figure 14*, has a toggle switch to change his availability. An admin, shown in *Figure 15*, has a link to a page with pending tutor applications. Included in the link is a notification for how many pending applications are in the system. Users can be either a tutor or admin or both. As shown in *Figure 16* and *Figure 17*, a tutor can indicate using the toggle switch whether they are free to tutor someone or if they are currently with a client.

Register and Edit Account

Registration



The screenshot shows the Mustang Tutors website's registration page. At the top, there is a dark blue header bar with the "MUSTANG TUTORS" logo featuring a red silhouette of a running horse. To the right of the logo are fields for "SMU ID" and "Password" with a "Login" button. Below the header are three buttons: "Search for Tutors", "Get Student's History", and "Register". The main content area has a light orange background and contains a white registration form. The form is titled "Registration" in a large, bold, dark blue font. It includes six input fields with validation messages: "First Name: * [Enter first name]", "Last Name: * [Enter last name]", "Email: * [Enter email]", "SMU ID: * [Enter SMU ID]", "Password: * [Enter password]", and "Confirm Password: * [Confirm password]". Below these fields is a "Register!" button. At the bottom of the page, there is a dark red footer bar with links for "About Us" and "Contact Us".

Figure 18: Registration page

A guest can create an account by clicking on “Register” in the navigation bar. The registration screen is shown in *Figure 18*. The user is automatically logged in once he presses the “Register!” button, so long as all the information is valid, and then in the future he will be able to log in to the website using the SMU ID and password he used to register.

Edit Account

The screenshot shows the Mustang Tutors website interface. At the top, there is a dark blue header with a red horse logo on the left, the text "MUSTANG TUTORS" in large white letters, and "Welcome, Story!" with a gear icon on the right. Below the header, there is a navigation bar with four buttons: "Applications 1", "Search for Tutors", "Student History", and "Tutor Status: Available". The main content area has a light orange background and contains a form titled "Edit Your Account". The form includes fields for First Name (Story), Last Name (Zanetti), New Password (placeholder: Enter new password), Confirm New Password (placeholder: Confirm new password), and Current Password (placeholder: Enter your current password). Below the form is a note: "In order to update any of your account information, please enter your current password to confirm your identity." At the bottom of the form are two buttons: "Reset" (pink) and "Update" (green). At the very bottom of the page, there is a dark red footer bar with links for "About Us" and "Contact Us".

Edit Your Account

First Name: * Story

Last Name: * Zanetti

New Password: Enter new password

Confirm New Password: Confirm new password

In order to update any of your account information, please enter your current password to confirm your identity.

Current Password: * Enter your current password

Reset **Update**

About Us Contact Us

Figure 19: Edit Account

Figure 19 shows the “Edit Your Account” page, which allows users to change their first and last name, as well as their password.

History

Your Student History: Student View

The screenshot shows the Mustang Tutors website interface. At the top, there is a dark blue header with a red horse logo on the left, the text "MUSTANG TUTORS" in large white letters, and a "Welcome, John!" message with a gear icon on the right. Below the header are three buttons: "Search for Tutors", "Student History", and "Become a Tutor". The main content area has a light beige background and features a title "Your Student History" with a share icon. A single meeting entry is displayed in a box: "MATH 1338: Calculus II" tutored by "Story Zanetti" on "2014-04-28" from "4:00 PM to 5:00 PM". A note from the tutor states, "The student was very prepared." At the bottom of the page is a dark red footer bar with "About Us" and "Contact Us" links.

Figure 20: Your Student History (Student)

A user who is a student can view a history of all of his past tutoring sessions. Whenever a tutor documents a tutoring session with this user using the website or Android app, a meeting is added to his student history. Each meeting in the history includes the tutor's name, the course tutored, the date and time of the meeting, and a short description submitted by the tutor to recount the meeting.

Your Student History: Tutor View

The screenshot shows the Mustang Tutors website interface. At the top, there is a logo of a red mustang horse and the text "MUSTANG TUTORS". To the right, it says "Welcome, Nick! 🌟" and "Tutor Status: Available". Below this, there are two buttons: "Search for Tutors" and "Student History". The "Student History" button is highlighted in red. There is also a "Busy" button. A "Tutor View" button is also visible. On the left, the title "Your Student History" is displayed. The main content area lists two tutoring sessions:

- CSE 1342: Programming Concepts**
Tutored by: Story Zanetti
horrible
- CSE 2341: Data Structures**
Tutored by: Story Zanetti
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris diam ligula, varius et lobortis in, imperdiet at metus. Nunc malesuada mi sed nisi pretium porta. Ut sed augue ac tellus consequat tristique in eu odio. Nunc porta odio sapien, non porttitor elit pretium a. Integer eu ultricies metus.

At the bottom, there is a dark red footer bar with links to "About Us" and "Contact Us".

Figure 21: Your Student History (Tutor)

Tutors also have access to their own student history, since it is possible for tutors to attend tutoring sessions themselves. A tutor's student history additionally has the "Tutor View/Student View" toggle button to switch between the tutor's Student View and the tutor's Tutor View.

Your Tutor History

The screenshot shows the Mustang Tutors website interface. At the top, there is a logo of a red and blue mustang, followed by the text "MUSTANG TUTORS". To the right, it says "Welcome, Nick!" with a gear icon. Below this, there is a "Tutor Status" section with two buttons: "Available" (highlighted in red) and "Busy". A navigation bar below the header includes "Search for Tutors", "Student History", "Tutor View" (highlighted in blue), and "Student View". The main content area is titled "Your Tutor History". It features a button "Document a meeting with a student." with a plus sign icon. Below this, there are two entries for meetings:

- CSE 2341: Data Structures**
Student tutored: Story Zanetti
Meeting notes: worst student ever
Date: 2014-03-26
Time: 3:00 PM to 4:00 PM
- EE 1301: Modern Electronic Technology**
Student tutored: Darius Clark
Meeting notes: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris diam ligula, varius et lobortis in, imperdiet at metus. Nunc malesuada mi sed nisi pretium porta. Ut sed augue ac tellus consequat tristique in eu odio. Nunc porta odio sapien, non porttitor elit pretium a. Integer eu ultricies metus.
Date: 2014-03-21
Time: 3:00 PM to 5:00 PM

Figure 22: Your Tutor History

By flipping the toggle switch over to Tutor View, a tutor can see a list of all the meetings he has documented with various students, with the most recent meetings listed first.

Documenting a Meeting

The image shows a screenshot of the Mustang Tutors website. At the top, there is a dark blue header with a red and blue horse logo on the left, the text "MUSTANG TUTORS" in large white letters, and "Welcome, Nick!" with a gear icon on the right. Below the header, there are three buttons: "Search for Tutors", "Student History", and "Tutor Status: Available". Under "Tutor Status", there are two sub-options: "Available" (highlighted in red) and "Busy". The main content area has a light orange background and features the heading "Your Tutor History" in a dark blue box. Below this, there is a form titled "Document a meeting with a student." It includes fields for "Student ID" (with placeholder "Enter student ID"), "Course Tutored" (with placeholder "Choose a course"), "Date" (set to "05/02/2014"), and "Time" (set to "01:00 PM to 02:00 PM"). There is also a text area for "Comments (1-3 sentences)" with a large input box. At the bottom of the form are two buttons: "Reset" and "Add". A small minus sign icon is located in the top right corner of the main content area.

Figure 23: Document a Meeting with a Student

By clicking the plus sign next to “Document a meeting with a student,” a tutor can fill out a form to record information about a tutoring session that he had with a student. Information that the tutor must write down are the student’s SMU ID, which course was tutored, the date and time of the meeting, and a short description of the meeting.

Sharing Your History

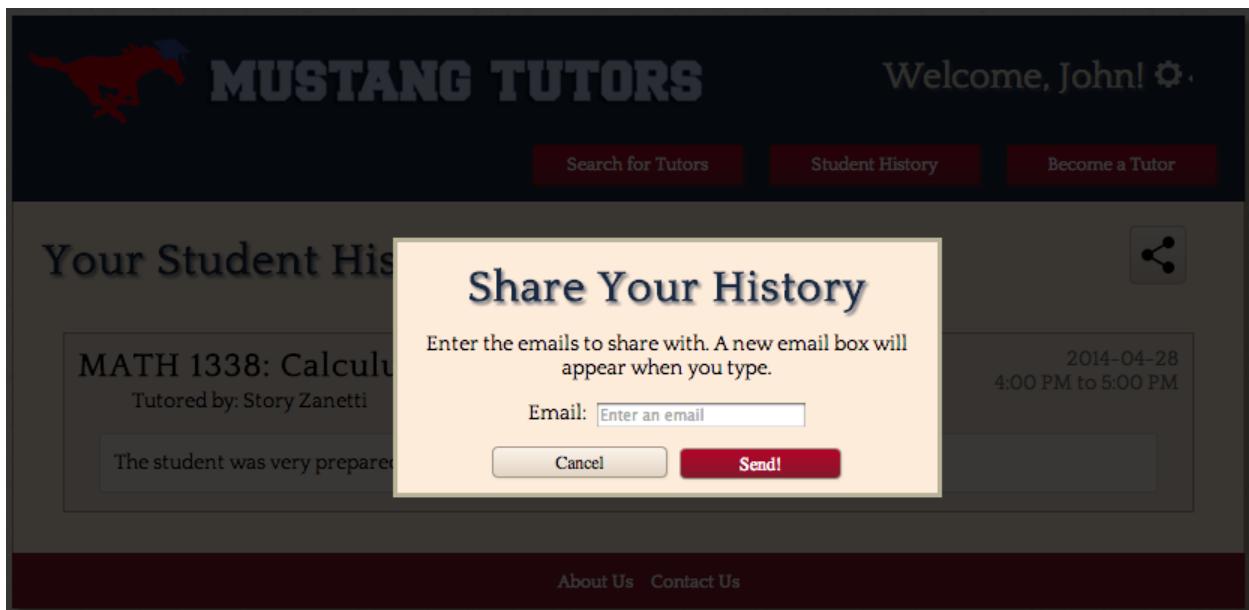
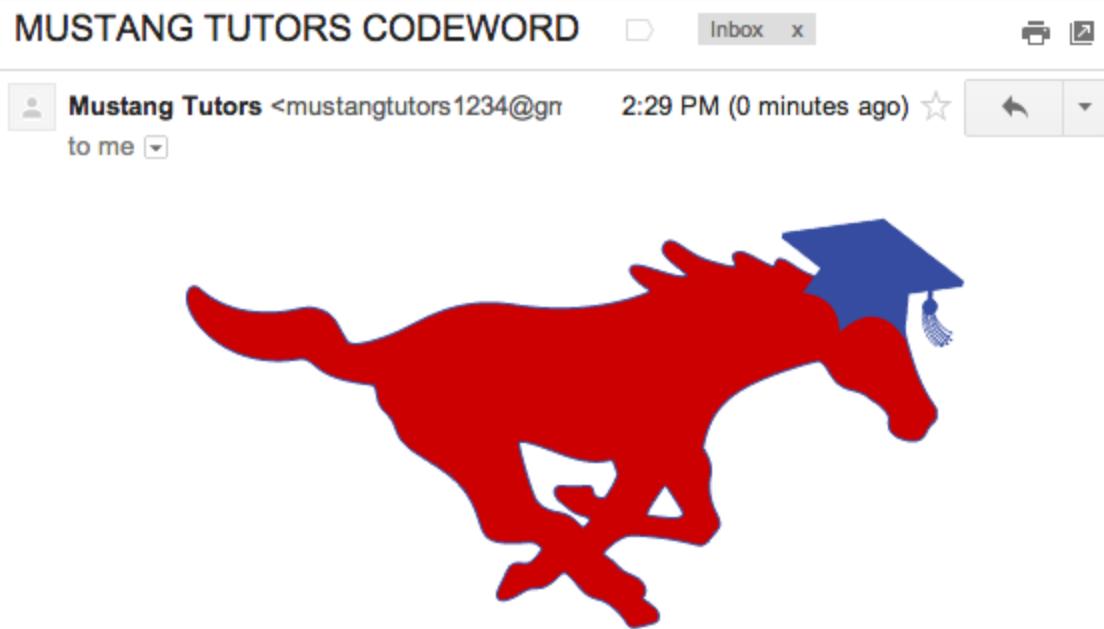


Figure 24: Share Your History

By clicking on the “share” icon in the top right of the student history page, a pop-up appears for the user to share his history with anybody. Multiple email addresses can be entered.

Get a Student's History: Email



Student's Records

Hello, You have been given access to a student's records through Mustang Tutors. In order to access this student's records please go to mustangtutors.floccul.us and click on the "Get Student's History" tab. Or you can follow this link directly to the page [Get Student's Records](#). Once on the page use this information to access their history.

Smu Id: 99999999

Codeword: 2024bb211c3

Thanks, Mustang Tutor's Staff

Figure 25: Email received getting a student's history/sharing a student's history

After a student shares his history with someone, as shown in *Figure 24*, that person will receive an email that has instructions on how to view the student's history. The email is shown in *Figure 25*.

Get a Student's History

The screenshot shows the Mustang Tutors website interface. At the top, there is a dark blue header with a red silhouette of a running horse on the left, followed by the text "MUSTANG TUTORS" in large white letters. To the right of the horse is a welcome message "Welcome, Nick! ⓘ". Below the header, there are two buttons: "Search for Tutors" and "Student History". To the right of these buttons is a "Tutor Status:" section with two tabs: "Available" (which is selected) and "Busy". A light orange banner below the header contains the text "Get Student's History". Underneath this, there are two input fields: "Enter SMU ID of Student: 99999999" and "Enter code word: 2024bb211c3", followed by a "Go!" button. The main content area has a light orange background and displays the results for a student named John Smith (ID 99999999). It shows a session for "MATH 1338: Calculus II" tutored by Story Zanetti on April 28, 2014, from 4:00 PM to 5:00 PM. A note in a box states: "The student was very prepared." At the bottom of the page is a dark red footer bar with links for "About Us" and "Contact Us".

Figure 26: Get Student's History

Anyone, regardless of whether he is logged in, can access the page to pull up a student's history. This page is accessed by hovering over "Student History" and clicking on "Get Student's History." After entering the SMU ID and the code word found in the email in *Figure 25*, the user will see a list of tutoring sessions that the requested student has attended.

Tutor Applications

Apply to be a Tutor

The screenshot shows a web-based application for becoming a tutor. At the top, there is a placeholder for a profile picture with a "Choose File" button and a preview area showing a photo of a person with blonde hair. Below this, the user's name "John Smith" and email "john@gmail.com" are displayed. On the left, under "Courses", there are two dropdown menus both set to "CSE 2341: Data Structures" and a pair of green and red circular buttons. On the right, under "Regular Hours", a table lists daily availability from Sunday to Saturday, with specific times for each day. A "Submit Application" button is located at the bottom right of the form area.

| Day | Start Time | End Time |
|-----------|------------|----------|
| Sunday | 12:00 AM | 12:00 AM |
| Monday | 05:00 PM | 10:00 PM |
| Tuesday | 12:00 AM | 12:00 AM |
| Wednesday | 12:00 PM | 08:00 PM |
| Thursday | 12:00 AM | 12:00 AM |
| Friday | 12:00 PM | 10:00 PM |
| Saturday | 05:00 PM | 08:00 PM |

Figure 27: Application to become a tutor

Application Sent

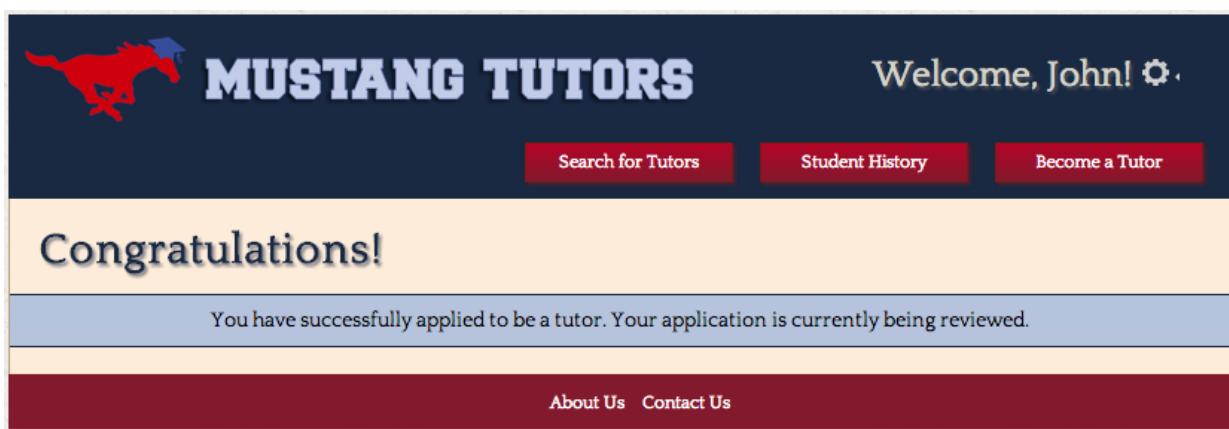


Figure 28: Confirmation of application's successful transmission

Users can apply to become tutors. In their application, they must provide a list of courses they can and want to tutor, as well as a list of hours that they can work. They can upload a profile picture, but it is optional. The application is shown in *Figure 27*. When the application is successfully submitted, the user is taken to the page in *Figure 28*.

Review Tutor Applications

The screenshot shows a web application interface for managing tutor applications. At the top, there is a header with a red horse logo, the text "MUSTANG TUTORS", and a welcome message "Welcome, Story! ⚙️". Below the header, there are several navigation buttons: "Applications 1", "Search for Tutors", "Student History", "Tutor Status: Available", and "Busy". The main content area is titled "Review Tutor Applications" and features a section for a "Test Account". This section includes a placeholder profile picture, a table of "Courses" (CSE 2341: Data Structures, CSE 1342: Programming Concepts), and a table of "Regular Hours" showing availability from Sunday to Saturday. At the bottom of the page are "Deny" and "Approve" buttons, and a footer with links to "About Us" and "Contact Us".

Figure 29: Admin page to review tutor applications

An Admin can review a list of all the tutor applications. He can either approve or deny each application. If denied, the user's application information is deleted, and the user can re-apply. If approved, the user's account is turned into a tutor account.

About Us and Contact Us

About Us

The screenshot shows the 'About Us' page of the Mustang Tutors website. At the top, there is a dark header bar with the 'MUSTANG TUTORS' logo on the left and a 'Welcome, John!' message on the right. Below the header, there are three red buttons labeled 'Search for Tutors', 'Student History', and 'Become a Tutor'. The main content area has a light beige background. It features a title 'About Us' in bold blue font, followed by a sub-headline 'Searching for a tutor? We can lend a hand.' A paragraph explains the challenges of the traditional A-LEC and how the service addresses them. To the right of the text is a photograph of two students, a boy and a girl, sitting at a desk and looking at an open book together. At the bottom of the page, there is a dark footer bar with links to 'About Us' and 'Contact Us'.

Figure 30: About Us page

Contact Us

The screenshot shows the 'Contact Us' page of the Mustang Tutors website. The layout is similar to the 'About Us' page, with a dark header bar, red navigation buttons, and a light beige main content area. The title 'Contact Us' is centered at the top of the content area. Below it are two boxes: 'Our Information' containing address and phone number details, and 'Tutoring Hours' listing days and times. A third box, 'Office Hours', shows a schedule from Monday to Sunday. The footer contains links to 'About Us' and 'Contact Us'.

Figure 31: Contact Us page

In the footer of the website are links to About Us and Contact Us pages, as shown in *Figure 30* and *Figure 31*. These are static pages that provide information about what the A-LEC is, where it is located, and what the general hours are.

Screenshots of Android Application

Homepage, Not Logged In and Logged In

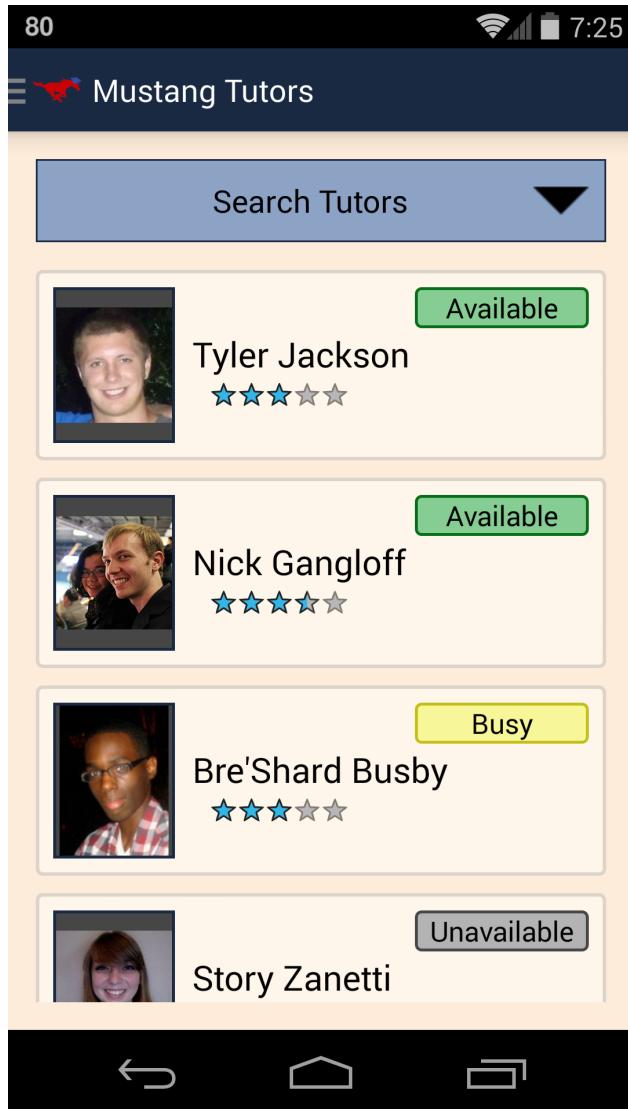


Figure 32: Homepage, not logged in

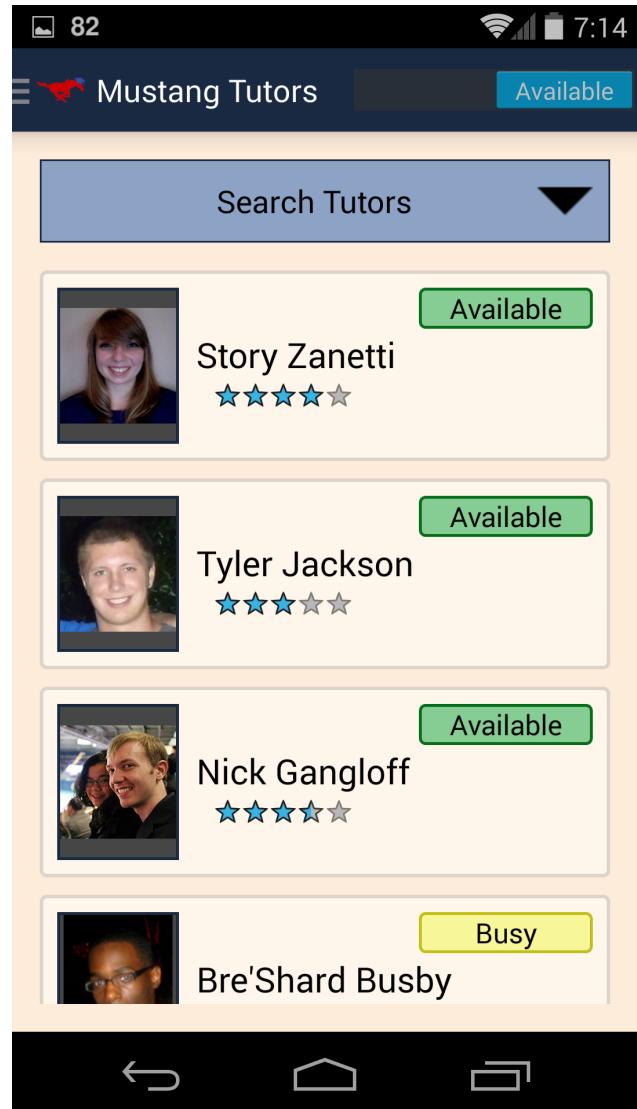


Figure 33: Homepage, logged in

Figure 32 shows the first page that a user sees when he opens the app. It displays a list of all the tutors at the ALEC and is sorted by their current availability. When a tutor is logged in, the action bar on the main page has a toggle switch for the tutor to toggle his availability between Available and Busy, as shown in Figure 33.

Tutor Search

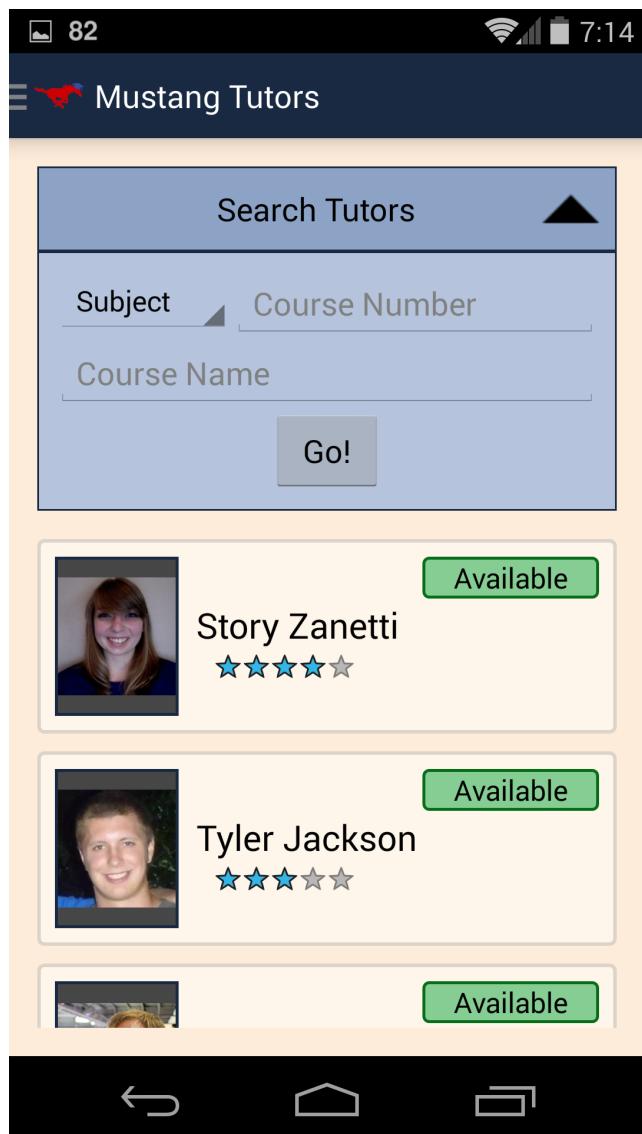


Figure 34: Search options

Clicking on “Search Tutors” will bring up a form that filters the tutors, as shown in *Figure 34*. A user can filter the tutors by subject (ex: CSE), course number (ex: 3345), or course name (ex: Graphical User Interfaces).

Navigation Drawer

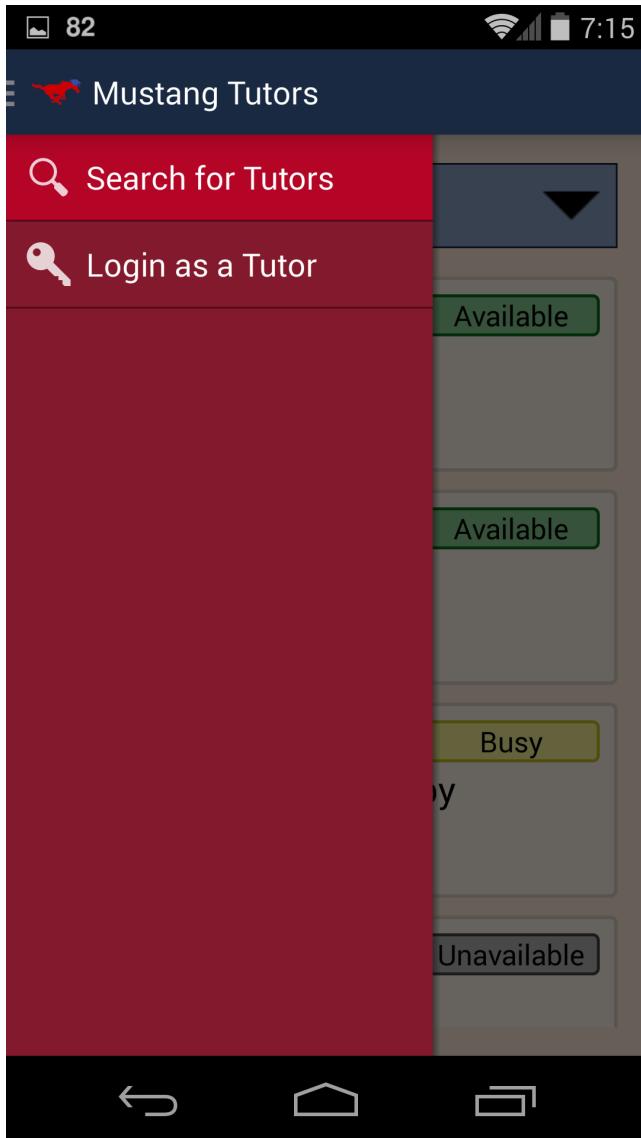


Figure 35: Navigation drawer, not logged in

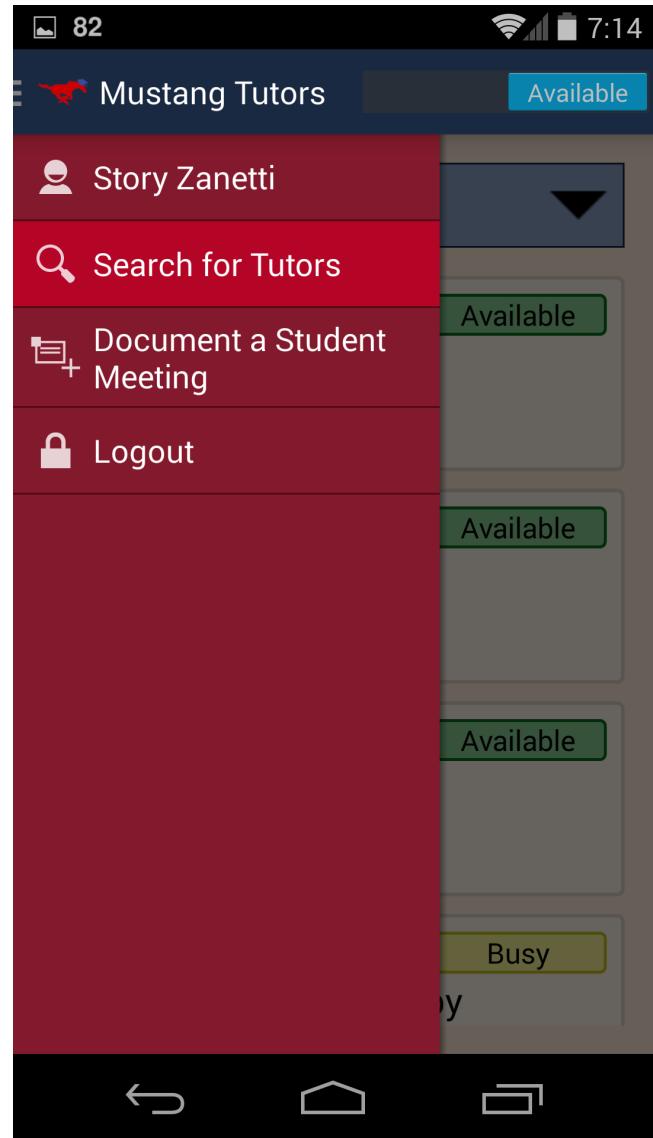


Figure 36: Navigation drawer, logged in

The navigation drawer has links to navigate to all the different pages of the application. The links in the navigation drawer change depending on whether the user is logged in, as shown in *Figure 36*, or not, as shown in *Figure 35*.

Tutor Profile Page

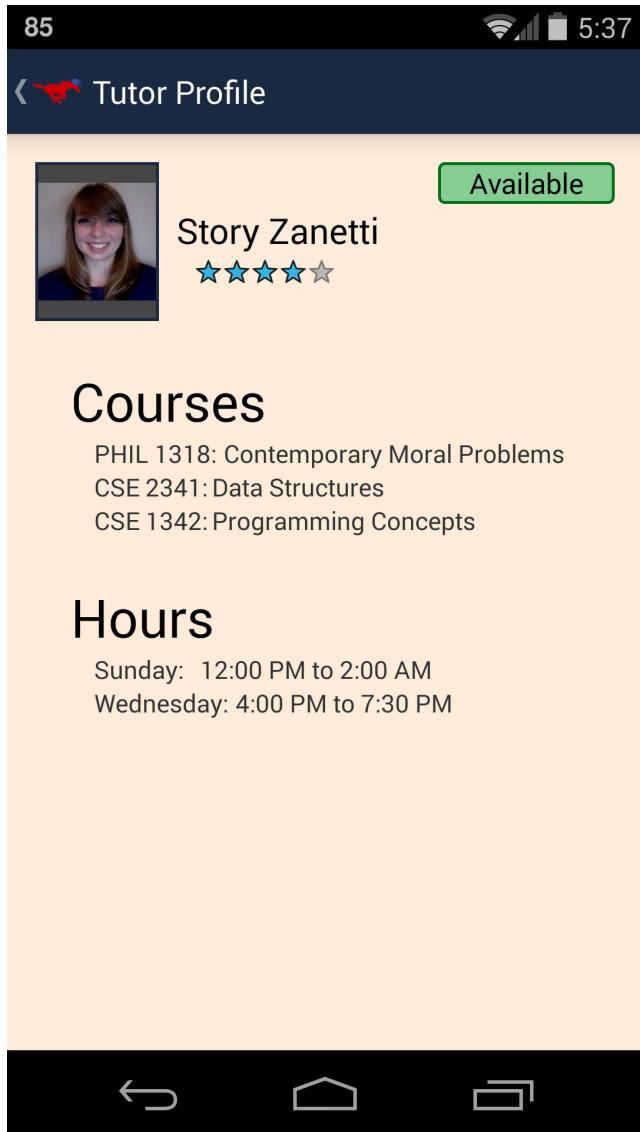


Figure 37: Tutor profile page

When a user clicks on a tutor in the search results, or if a logged in tutor clicks on their own name in the navigation drawer, they are taken to the page in *Figure 37*. The page is populated with the selected tutor's name, picture, average rating, availability, courses, and regular hours.

Login as a Tutor

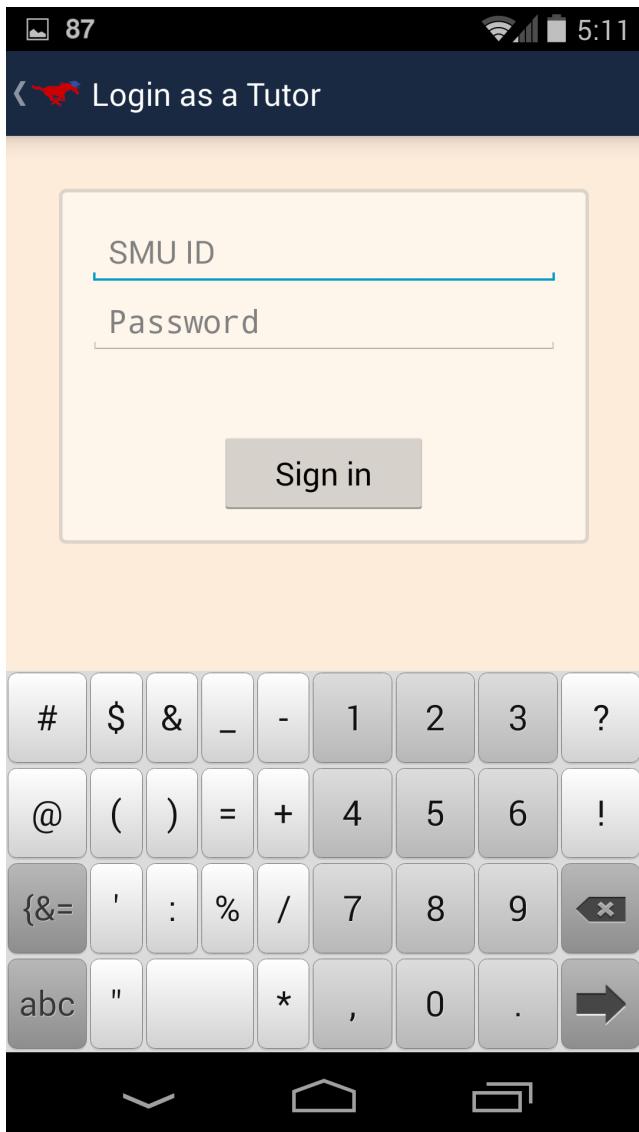


Figure 38: Log In

A tutor can use the page shown in *Figure 38* to log in to the app. Non-tutors who try to log in will be told that only tutors can log in.

Document a Student Meeting

The screenshot shows a mobile application interface for documenting a student meeting. At the top, there is a navigation bar with a back arrow icon, the text "Document a Student Meeting", signal strength, battery level, and the time "2:49". Below the navigation bar is a form area with the following fields:

- Student ID:** A text input field containing the value "11111111".
- Course Tutored:** A text input field containing the value "PHIL 1318: Conte..". To the right of this field is a "More >>" button.
- Date:** A blue button displaying the date "5/3/2014".
- Time:** Two blue buttons showing the time range "3:00 PM - 5:00 PM".
- Comments (1-3 sentences):** A text input field containing the value "bad student".

At the bottom of the form are two buttons: "Reset" (gray) and "Submit" (red). The bottom of the screen features a black navigation bar with three icons: a left arrow, a home icon, and a square icon.

Figure 39: Documenting a meeting

A logged in tutor has access to the page shown in *Figure 39* from the navigation drawer. After a tutor meets with a student, he/she can document the details of the tutoring session by filling out this form.

Testing

The testing phase of the software development cycle is an important part of creating an application. Our team took testing very seriously; our team members constantly tested each other's work during development, and whenever an issue was discovered, it was noted in a shared document. Throughout the entire development phase, team members regularly showed the product prototype to friends, family, or other peers to collect constructive feedback from them. The criticism we received was very valuable. One testee complained that some of the terminology was overly confusing, such as the difference between Available and Busy on the toggle switch. Another testee disliked the hovering mechanism on the tutor search page on the website, and she suggested that a delay be added so that the additional information only pops up after a period of hovering. Whenever we received suggestions such as these, we evaluated whether many people would likely have the same concern, and if so, we made the change in our product.

In addition to receiving suggestions from peers not in CSE 3330 or CSE 3345, our team's product was tested by the members of another team. This team was very thorough in testing our product, and found many issues that we overlooked during development and in our own tests. Most of the issues that they reported were legitimate problems that our team agreed needed to be fixed, and we did so quickly. We maintained a good line of communication with this team. Whenever they needed something about our application clarified, they would send us a message with their question, and we would promptly respond.

We reciprocated the testing relationship by testing a different team's product for bugs and issues. During the first testing phase our team had difficulty finding issues with their web application since it was lacking in features. However, good progress had been made with their Android application, so we were able to give good feedback on it, and we reported many issues. In the second testing phase, the other team was able to provide more features on both their web and Android applications, so our team was able to give feedback for both. They responded quickly to any issues we posted. Occasionally they would need clarification on a bug report, in which case the reporter of the issue would respond with more information. They appreciated our reports and made efforts to address them all.

Team Reflection

One of the primary challenges starting this project was that half of the team had already taken databases, while hardly anyone had experience with GUI. This created a slight unbalance in the team as some team members already knew things others were learning. This did not inhibit work from getting done, but rather stirred up controversy as each side (DB and GUI) wanted some independence from each other. Either way, the work still got done in a very efficient manner, and there were always opportunities to ask team members for help when needed.

Out-of-class Learning

One of the biggest contributors to the success of our project was the extensive amount of out-of-class learning done by all of our team members. In addition, the majority of this research occurred very early in the semester, so we never felt pressured by time. By establishing the majority of the infrastructure of our project early on, we were able to sail smoothly through the rest of the project without hitting major roadblocks.

The biggest out-of-class challenge for the database team members was learning how to implement a RESTful API using the Slim Framework (for Taco Truck) and Laravel Framework (for Mustang Tutors). Perhaps the most challenging part of using these frameworks was setting them up on the server; many minor configuration issues can prevent the framework from functioning. After the setup process, writing the HTTP routes and functions was relatively simple, since there is a plethora of documentation and tutorials. The RESTful API also posed a challenge to the GUI team members, since in order to use the routes and to assist in debugging, they needed to also have some level of understanding of how the frameworks functioned.

Our team agreed on using JSON for passing data back and forth between the backend and frontend, but since JSON is primarily a topic that is discussed in CSE 3345 and not CSE 3330, the database members needed to learn how JSON worked if they didn't already know. Fortunately, JSON is relatively simple to understand.

One topic that was not discussed in CSE 3345, but the GUI members found very helpful for the project, was jQuery AJAX requests. XML HTTP Requests written in pure Javascript are unwieldy, and given the large quantity of API calls that need to be made, any amount of code simplification is helpful. Instead of using Javascript XML HTTP Requests, the GUI members used jQuery AJAX, which greatly condensed the code.

The main tenet of our group was starting early, and the Android application was no exception. In our iteration plans, we divided up the development of the Android app evenly between the first and second iterations, which meant we would have to begin learning Android development many weeks before we officially began learning it in CSE 3345. Learning Android development without the guidance of a professor was extremely challenging, but the outcome was rewarding, since a significant portion of the app was finished early in the semester and it didn't need to be rushed. Part of the reason that learning it on our own was so difficult was because we didn't have any notion of the level of difficulty of the things we were trying to do, like implementing a navigation drawer or using fragments or using a toggle switch. Many of the pieces we worked with turned out to be topics never covered in CSE 3345.

In addition to developing an Android application, we created a prototype of an iOS application. Since the database members finished the majority of their during the first iteration and the beginning of the second iteration, they had free time at the end of the project. One database member decided to use this time to create an additional mobile application for a different platform, in order to support as many users as possible, in the event that the A-LEC were to actually use our application. iOS application development is a topic that is taught in neither CSE 3330 or CSE 3345, so this team member brought in a significant amount of outside knowledge and additional research.

Overall Reflection

Aside from the initial misunderstanding one where to stand one's ground, our group performed exceptionally well. There is no perfect group, but our group did perform at a very efficient pace without causing too much inner disturbances. Many of the initial problems that the group had faced could have easily been solved with better communication at the beginning. While our team did divide up all parts of the project and stress the importance of deadlines; our group did not specify how we wanted to approach our problems. This was where the "crossing into others ground" problem came about. With more thorough communication at the beginning, this problem probably would have never risen.

Wrap Up

Keeping track of a large number of tutors is a daunting task, especially considering the fact that some tutors, for one reason or the other, may discontinue working as a tutor. So, as it stands, our product keeps record of all tutors that have worked, or are currently working at the A-LEC, whether they are currently employed or not. Therefore, in the future, we plan to be able to delete tutors from the system that are no longer working at the A-LEC, so as to keep the amount of data to track minimal.

While our application grants users the ability to apply for the position of a tutor, it does not notify applicants of the status of their application. Currently, the only ways that a user can learn whether or not their application has been accepted is by being told personally by the A-LEC or by logging into the application and discovering that their profile page has changed. To facilitate better communication about a user's application, we plan on sending email notifications to users in the event that their application has been approved or denied.

We have demonstrated the capabilities of our web application and our Android application. Now, we plan on making our production available across more platforms, particular on iOS. We strive to create an iPhone application that provides much of the same functionality as our Android application. This will grant other mobile users access to our product, without requiring them to completely change their phone of choice.

Finally, we feel that our product will be a large asset to the A-LEC. It's ease-of-use can greatly increase the productivity of the A-LEC and help it to become organized. Therefore, we are hoping to sell our Mustang Tutors application to Southern Methodist University for use by the A-LEC. Of course, we also wish to continue being part of the project in order to maintain and update it when needed. This also means that we want to remain the primary owners of the product.

Appendix: Data Dictionary

Users:

The users table holds all of our user's information. This includes their first and last name, as well as their SMU Id and a password that they can use to login to their account. We also store an email address, and a code word that we use to give someone access to the student's history. The user's table also holds a field for available and active. Their availability will change based on whether they are currently able to tutor. Their active field will be changed by an admin if a tutor stops tutoring for a period of time. The tutor field is how we store if a user is a tutor or if they are just a normal user. The Admin field is similar, and will store if the user is an Admin. Their user identification is a unique identifier for each user.

| Attribute | Type | Valid Range | Description |
|------------|--------------|--|--|
| User_ID | INT | 1-∞ | The unique ID that identifies each individual user. |
| SMU_ID | INT | Any 8 digit integer | SMU identification number |
| First_Name | VARCHAR(30) | Any string | User's first name |
| Last_Name | VARCHAR(30) | Any string | User's last name |
| Email | VARCHAR(255) | Any string following the regex "[a-z0-9]+@[a-z0-9]+\.[a-z0-9]+" | The email associated with their account. |
| Password | VARCHAR(255) | Any string | The password used to log in. |
| Codeword | VARCHAR(255) | Any string | The codeword used to gain access to their student history. |
| Available | INT | 0-2 | Their availability status(2 for available, 1 for busy). |
| Active | INT | 0-1 | Their active status (1 for active, 0 for inactive). |
| Tutor | INT | 0-1 | Identifier for a tutor (1 for tutor, 0 for normal user). |
| Admin | INT | 0-1 | Identifier for an admin (1 for admin, 0 for not). |

Courses:

The courses table holds the course information for the courses offered by our tutors. The subject field is the course subject abbreviation (MATH, PHYS, CSE). The course number is the course number and the course name is the course name. The course id is the field we use to uniquely identify each course offered. One example would be “CSE 3381 Digital Logic Design”. In this case CSE is the subject, 3381 is the course number and Digital Logic Design is the course name. We would then assign it a unique identifier for the course id.

| Attribute | Type | Valid Range | Description |
|---------------|-------------|-------------|--|
| Course_ID | INT | 1-∞ | The unique ID that identifies each individual course |
| Subject | VARCHAR(30) | Any string | The subject abbreviation for a course (PHY,CSE,MTH, etc.). |
| Course_Number | INT | 1-∞ | The course number for a course. |
| Course_Name | VARCHAR(30) | Any string | The course name for a course. |

Courses_Tutored:

The courses tutored table holds the relationship between our tutors and the courses each of them offer. The course id field references the course id from our courses table that corresponds to the course the tutor is offering. The user id references the user id from our users table that corresponds to the tutor that offers that course. The courses tutored id is the unique identifier we use for the table.

| Attribute | Type | Valid Range | Description |
|--------------------|------|-------------|---|
| Courses_Tutored_ID | INT | 1-∞ | The unique ID that identifies each relation between Course and User |
| Course_ID | INT | 1-∞ | The foreign key referencing the corresponding course id from the courses table. |
| User_ID | INT | 1-∞ | The foreign key referencing the corresponding tutor's user id from the users table. |

Schedule:

The schedule table holds each tutor's available hours. When they set up their account, they can specify what days and times they are available to tutor, or they can modify their schedule at any time after. The user id field references the corresponding tutor's user id from the users table. The day field holds a numeric value ranging from 1-7 that represents the day they can tutor (1 representing Sunday, 2 Monday, etc.). The start time is the field for the beginning of their shift, and the end time holds when their shift ends.

| Attribute | Type | Valid Range | Description |
|-------------|------|-------------|--|
| Schedule_ID | INT | 1-∞ | The unique ID that identifies each individual piece of a schedule |
| User_ID | INT | 1-∞ | The foreign key referencing the corresponding tutor's user id from the users table |
| Day | INT | 1-7 | The numeric value corresponding to a day of the week (1 = Sunday, 2 = Monday, etc.). |
| Start_Time | TIME | Any Time | The time of the beginning of their shift. |
| End_Time | TIME | Any Time | The time when their shift ends. |

Ratings:

The rating table holds each of the tutor's user submitted ratings. A user (student) can go to a Tutor profile and offer a rating. This rating is stored as an int from 1-5 and will be added to the Tutor_ID's bank of ratings. These values then can be pulled with a query to obtain the average rating of a tutor. Individual ratings from specific students can be seen when the tutor history page or tutor profile page is requested.

| Attribute | Type | Valid Range | Description |
|-----------|------|-------------|---|
| Rating_ID | INT | 1-∞ | The unique ID that identifies each individual Rating. |
| User_ID | INT | 1-∞ | The ID of the user who gave the rating |
| Tutor_ID | INT | 1-∞ | The ID of the tutor being rated |
| Rating | INT | 1-5 | The rating of the tutor, on a scale of 1 to 5 |

Comments:

Each user, primarily a student, can go to a particular tutor (identified by the Tutor_id) to submit a VARCHAR comment. A user can submit multiple comments to a particular Tutor. Every comment that is submitted will also be marked with a default configured time stamp of a date and time.

| Attribute | Type | Valid Range | Description |
|------------|--------------|--------------|--|
| Comment_ID | INT | 1-∞ | The unique ID that identifies each individual Comment. |
| User_ID | INT | 1-∞ | The ID of the user commenting |
| Tutor_ID | INT | 1-∞ | The ID of the tutor being commented on |
| Comment | VARCHAR(500) | Any String | The comment written by the user who was tutored |
| Timestamp | DATETIME | Any DATETIME | The date and time the comment was posted |

Records:

After every teaching session that a tutor has with a student, the tutor will submit a student record. This record will be associated with the User_ID. These records must be associated with the User_ID to ensure that when a parent requests the student records, they only retrieve one particular student. These records also contain the date and the time that the student was tutored. Within these records is a short summary composed of a VARCHAR. All records are associated with only the User_ID and the Tutor_ID.

| Attribute | Type | Valid Range | Description |
|---------------|--------------|-------------|--|
| Record_ID | INT | 1-∞ | The unique ID that identifies each individual Record |
| User_ID | INT | 1-∞ | The foreign key referencing the ID of the user who was tutored |
| Tutor_User_ID | VARCHAR(30) | 1-∞ | The foreign key referencing the ID of the tutor |
| Date | DATE | Any Date | The date that the tutoring session took place |
| Summary | VARCHAR(255) | Any string | A summary of the results of the session, written by the tutor |
| Start_Time | TIME | Any Time | The time the session began |
| End_Time | TIME | Any Time | The time the session ended |

Applications:

The application table holds the information used to allow a student to apply for a tutor role. This table holds the Application_ID which identifies specific applications. Then the User_ID will associate with the student that wants to become a tutor. The attribute of pending, marked by a int, will act as a flag for whether the user will be granted privileges as a tutor.

| Attribute | Type | Valid Range | Description |
|-----------------------|------|-------------|---|
| Application_ID | INT | 1-∞ | The unique ID that identifies each individual Application. |
| User_ID | INT | 1-∞ | The ID of the user applying |
| Pending | INT | 0-2 | The status of the application (0=pending, 1=approved, 2=denied) |