Greek Life Member Management System

10.05.2018

Software Engineering 1 - Group 9 - Report 1

Jason Pulis, Christopher Whetsel, Mike Winkelmann

File Repository - https://github.com/mawinkelmann/databaseUpgradeSEGroup9

Development Server - http://glmms.online

Client Server - http://spdmizzou.com

Development Blog - https://glmms.home.blog

# All team members contributed equally.

Table of Contents

1 Project Description

1.1 Problem Diagnosis

One of the problems that the engineering fraternity Sigma Phi Delta is facing is the inability to effectively communicate events, knowledge and other pertinent information to the members of the group. This is brought about by the extensive list of applications that are designed to bring groups to a common line of communication. Sigma Phi Delta is currently using a hybrid of communication networks such as Slack, GroupMe, email and OrgSync. As a result, the members and officers in our organization are frustrated by the constant application hopping, inconsistent notifications for events and the lack of carry-over from each application. The Executive Board of the fraternity has also found out that many of the members have turned off the notifications from the chat because of the spam that comes from the communication overload. Thus, the Executive Board of Sigma Phi Delta has decided to seek a centralized location for member information and communication.

Another issue that the fraternity is facing is a jumbled mess of documents, by-laws and bills that stream through emails and get put into a cluttered Google Drive. Unless a member of the fraternity plays a large role in the upkeep and development of the documents that are used every week, there is no reason for them to look at them. That being said, when a member does need to reference a by-law or lookup formatting for a bill, the Google Drive becomes a maze of folders - leading to dead-ends, out of date files and duplicates. Because of this, some variety of search function and organization needs to be implemented to help the less tech-savvy people in the fraternity.

For the executive board and other decision makers in the fraternity to be successful they need to be able to access information about the members they are leading. Currently, when we need information, the member seeking it needs to search through the fraternity Google Drive for it, if it even exists. This process is time consuming and often fruitless. If that information cannot be found, they resort to polling the group message and hoping all necessary members will respond with their information. This is inefficient and often inaccurate, costing time and delaying meaningful work that requires these answers.

The Secretary of Sigma Phi Delta has also raised the issue that weekly officer reports are time consuming to compile and collect from each position holder. It is a menial task he must perform every week. They are submitted by each officer via email to the secretary who is incharge of compiling them. The officers often forget to send one in or do not indicate to the Secretary that they will not be submitting one. The Fraternity also is seeking a way to speed up its officer elections. Currently voting is done by hand and counted manually by the Secretary. This takes time and is possible to introduce error.

Last year, an attempt was made by members of the Sigma Phi Delta fraternity to implement a member information system. While the design of the database itself was good, the implementation of the web interface and the small set of features offered made sure that it quickly failed. It has since fallen out of use and never fulfilled its goals. Its interface was confusing and difficult to navigate, providing little instruction on how to use the system. The database also never held information for all members, so using it did not provide valuable information. We need a way to provide the functionality this system tried to achieve, but in a way that will be used by the fraternity for years to come.

Clearly, Sigma Phi Delta is facing many issues regarding management of information, documents, and communication that frustrates our members, costs fraternity leaders time, and could lead to inaccurate action to be taken. These issues need to be addressed.

1.2 Solution

To solve the problems described in the above section, we propose a Greek Life Member Management System, which will build upon the existing member information database that Sigma Phi Delta previously attempted to implement. Lack of interest because of unfinished features and an unpolished user experience were major drawbacks of the previous solution. So much so, that it has been abandoned as a solution to our problems. The new solution must be more user friendly and provide new features which will make it more appealing for general members and decision makers in our organization alike. The main benefit of the system to users will be a centralized and searchable location accessible to all members for information regarding organization personnel, announcements, events, and documents. We also want this solution to be useable and maintainable for many years, so that we have a continuity of member information.

Our desired solution has several goals to improve fraternity communication and information management. One goal in creating this system is to provide valuable and accurate information for the leaders of the fraternity as they make decisions impacting all members. The solution must also be easy to use and should save members time and hassle when they are trying to compile information. The information must be gatherable by the members without having to reach out to each member individually or through a group message. The system should also provide a way to coordinate members for events and will provide an organized archive for past chapter documents. Events should also be synchronized with the fraternity google calendar as most members already have it on their phones. Our solution will also make it simple to find important announcements for the chapter and will allow urgent messages to be sent out by text to all members. Another desirable feature would be to randomly pick members to clean up after events and to keep track of who has already been chosen in the past. A feature to allow officers to submit their reports online and compile them would be desirable. The system should also send

reminders to officers that have not submitted a report. There should be a way to create a voting poll for each position when we hold elections that allows each member to vote once. Finally, the solution must be secure and only accessible to members of the hosting organization.

This project hopes to satisfy the requirements for what a Greek Life chapter needs to function as one cohesive unit. If the implemented features are easy to learn and maintain, the better the organization will survive in the long run because we will be able to manage and learn from our information. Each member must be able to pick-up the programs that are created and intuitively solve whatever they seek to accomplish. It should also be expandable so that in the future, the fraternity can add features as new needs arise. As an end goal, this could be used for any Greek Life organization. One of the things that we realize this relies on is member participation in the system, because if we do not consistently use this as our form of communication and keep the information up to date, we will not be able to benefit from this system.

1.3 Glossary of Terms

Database - a usually large collection of data organized especially for rapid search and retrieval (as by a computer).

Fraternity - a student organization for scholastic, professional, or extracurricular activities.

Greek Life - Fraternities and sororities, or Greek letter organizations (GLOs) (collectively referred to as "Greek life") are social organizations at colleges and universities.

Fraternity Father/Son - In fraternities, during the new member education process, a new member to the organisation is given a “father” to help mentor them in the fraternity.

Chapter - a local branch of an organization (fraternity) or a weekly meeting held by a fraternity or sorority which all members are required to attend.

Greek Life Member - a member of a Greek Life organization.

Fraternity Member - an initiated member of a fraternity.

Member Information - information about a member of an organization that could be useful for the organisation to keep track off. E.g. major, phone number, graduation date, email etc.

Sigma Phi Delta - an international professional-social fraternity of engineers.

Executive Board - the group of elected individuals who oversee the activities of the organization. In Sigma Phi Delta, the Executive Board consists of the President, Vice President, Internal Operations Engineer, Secretary, Treasurer, Social Chair, Recruitment Chair, Philanthropy Chair, and New Member Educator.

Slack - a cloud-based set of proprietary team collaboration tools and services. The name is

an acronym for "Searchable Log of All Conversation and Knowledge".

GroupMe - a mobile group messaging app.

OrgSync - an online community management system to higher education institutions in the United States and Canada. OrgSync enables colleges and universities to communicate with students and staff, track student involvement, and manage campus organizations and programs.

Google Drive - a file storage and synchronization service developed by Google.

Decision Makers - members of the organisation who have to make decisions that will impact the entire organisation.

Officer Reports - something that an certain member of the fraternity sends in to have read during chapter, usually important announcements

Email Server - a computer running special software which enables the exchange of emails using Simple Mail Transfer Protocol (SMTP).

SMS Gateway - it allows a computer to send or receive Short Message Service (SMS) transmissions to or from a telecommunications network, mostly routing to mobile phone networks.

Adobe Photoshop - a computer program that allows users to create and edit images interactively on the computer screen developed and published by Adobe Systems for macOS and Windows.

draw.io - draw.io is an open source technology for creating diagrams

Github - is a web-based hosting service for version control using Git which is mostly used for computer code.

Fraternity Google Calendar - Google Calender is a calendar application used by the Fraternity to keep track of organization events.

2 System Requirements

Based upon the needs of Sigma Phi Delta, we developed a list of requirements for the system to possess. For each requirement, we assign an identifier in the form of REQ-(1,2,3...x), as well as a priority weight from 1 to 5. A lower priority weight indicates that the corresponding requirement is more essential to the success of the project, and more critical to fulfilling the customer's needs.

2.1 Enumerated Functional Requirements

ID	Priority	Description
REQ-1	**1**	System shall have searchable member information database with the ability to generate spreadsheets from a query
REQ-2	**1**	System shall have secure login over https
REQ-3	**1**	System shall allow administrators to change member information and add new members
REQ-4	**1**	System shall send urgent announcements via text message to all active members
REG-5	**2**	System shall have different levels of access for certain users (i.e. admin or executive member)
REQ-6	**2**	System shall have the ability for each member to update and view their information on the system
REQ-7	**2**	Users shall be able to display member and alumni employment information and position history
REQ-8	**2**	The system shall be able to send initial and periodic reminders for users to update their information
REQ-9	**2**	System shall have filterable announcements feed with topics
REQ-10	**3**	Members who have a position should be able to access transition material and the description of their position responsibilities
REQ-11	**3**	Users should have the ability to recover account after forgotten password

REQ-12	**3**	Users should have the ability to randomly choose members from the database for extra duties
REQ-13	**3**	System should have the ability to link events from database to fraternity Google Calendar
REQ-14	**4**	Users should have the ability to create an event that users can RSVP to and the event creator can send email reminders to the RSVP-ed users
REQ-15	**4**	System should have an organised file archive
REQ-16	**5**	System should have the ability to generate a graphical fraternity family tree
REQ-17	**5**	System should have the ability to host fraternity election voting polls
REQ-18	**3**	System should have the ability to create/submit officer report forms

2.2 Enumerated Nonfunctional Requirements

ID	Priority	Description
REQ-19	**1**	UI shall be intuitive.
REQ-20	**1**	UI shall be mobile friendly.
REQ-21	**2**	System response times to User actions should be under 3 seconds.

2.3 On-Screen Appearance Requirements

Our website will serve as a starting point for the members of the fraternity. Important links should be easy to find, toolbars should be used in the header and/or footer. Our website must be intuitive and functional while remaining aesthetically pleasing. Many of the design elements will be taken from the central colors and themes of the fraternity. These images are mock-ups, and might change in the future based on needs/requirements.

.

fig. 1 (Crest of Sigma Phi Delta) Will be included in the design of the UI.

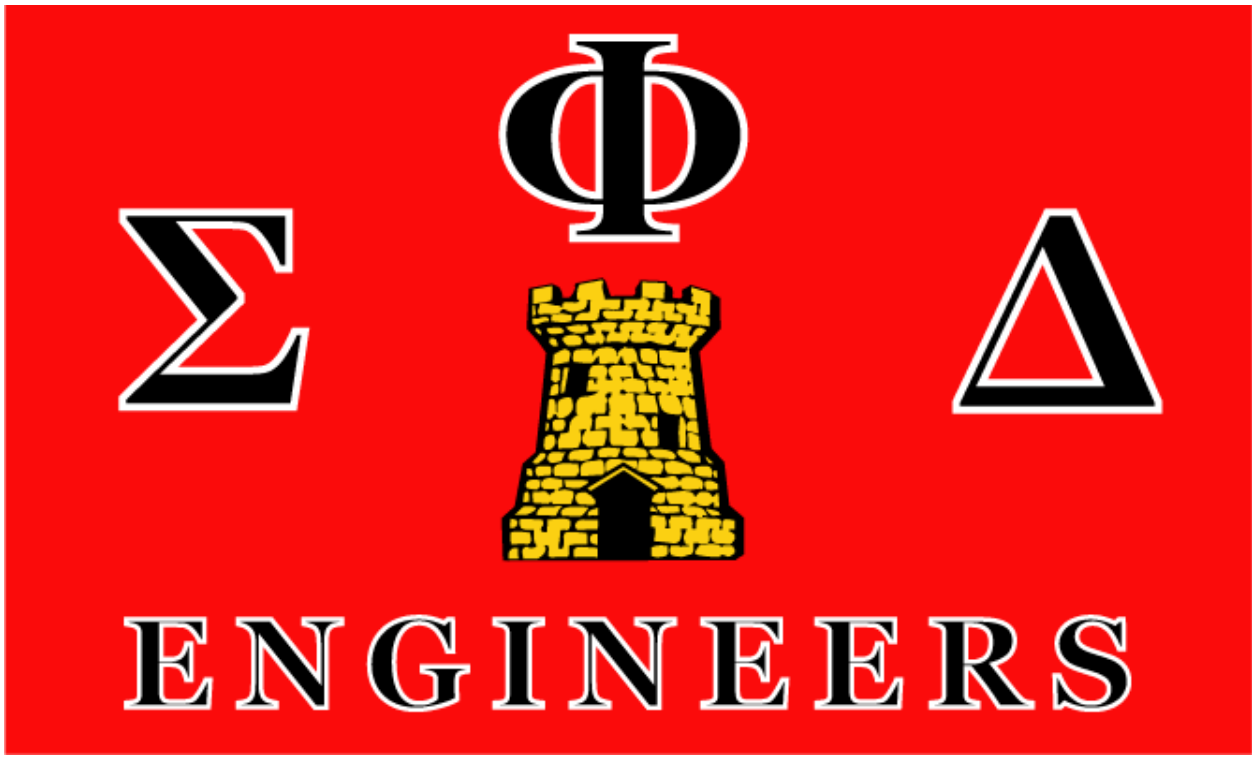


fig. 2 (Flag of Sigma Phi Delta) Will be included in the design of the UI.

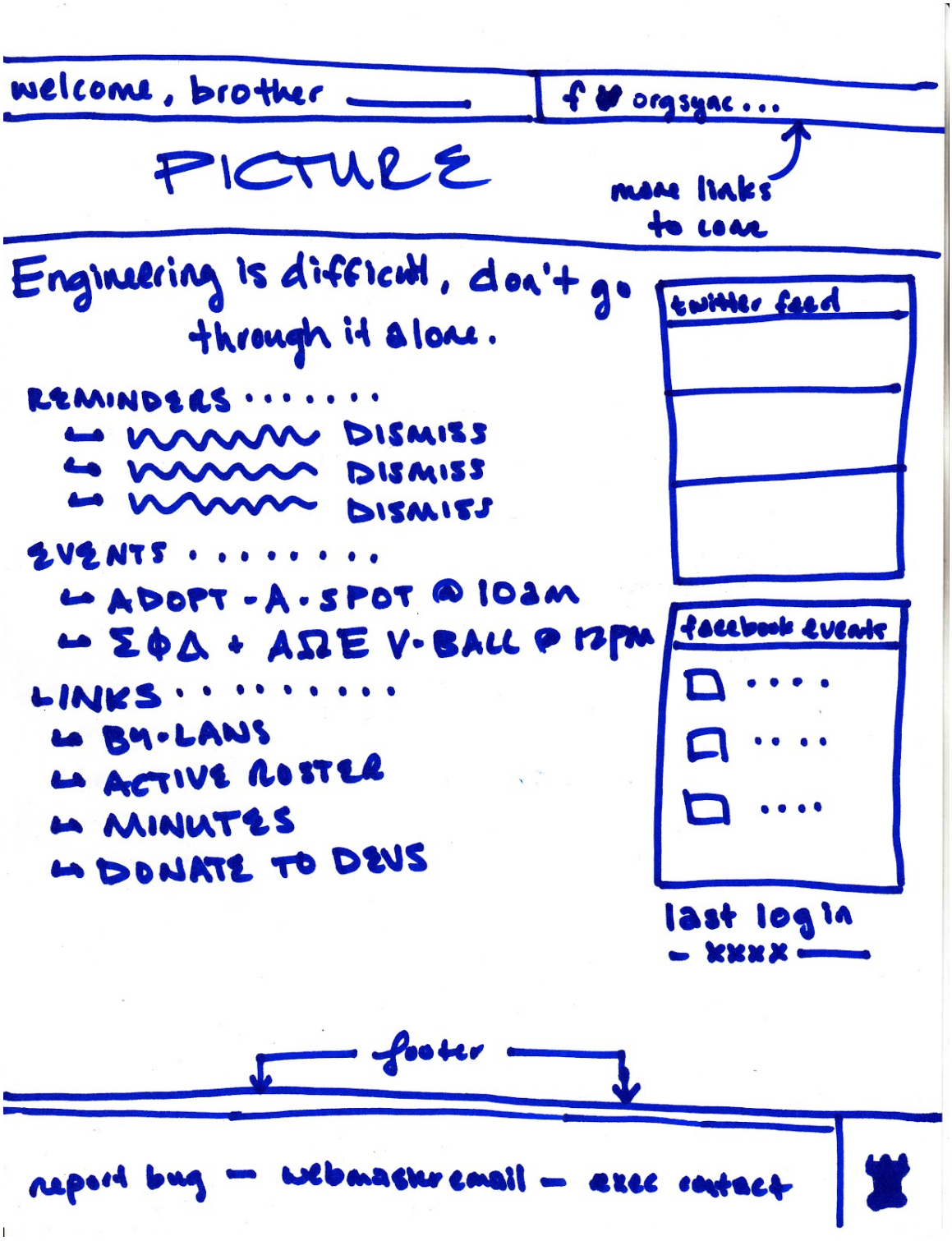


fig. 3 (Member desktop homepage)

fig. 4 (Member mobile homepage)

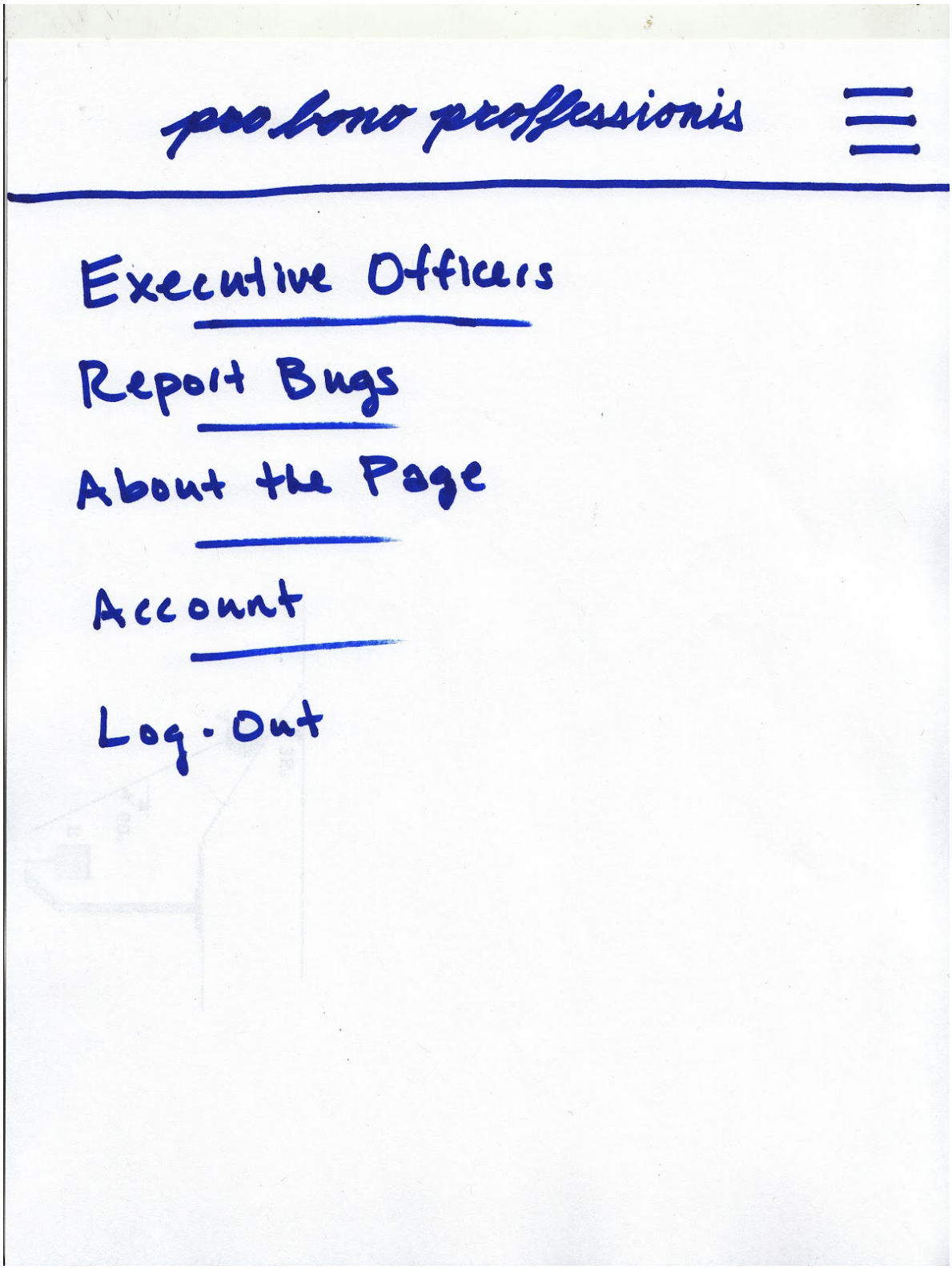


fig. 5 (Member hamburger menu)

3 Functional Requirements Specification

3.1 Stakeholders

Stakeholders include individuals and organizations which are interested in the completion and use of a given product. The amount of stakeholders and different types of stakeholders relies on the versatility and ease-of-use of the product in question. Stakeholders for this project include all current and future members and alumni of the Sigma Phi Delta fraternity. Other Greek Life organizations at Mizzou or around the country may also be interested in this project to help organize their information and communication inside their organization.

Specifically, from Sigma Phi Delta, the Executive Board has sanctioned the creation of this product and will be the primary product owners for the project. Members of the Executive Board include:

A. President: Jake Shulman
B. Vice President: Dameron Taylor
C. Internal Operations Engineer: Thomas Schuly
D. Secretary: Mike Winkelmann
E. Treasurer: Josh Westbrook
F. Social Chair: Aaron Henry
G. Recruitment Chair: Marcos Ehinger
H. Philanthropy Chair: Austin Kimes
I. New Member Educator: Zack Becker

3.2 Actors and Goals

Actors can be defined as are people or devices that will directly interact with the product, and can also be loosely labeled as either "initiators" or "participators". These actors will have a specific goal with the given product, which is what the actors are attempting to achieve by interacting with the system. Actors and their respective goals are: Human actors will be interacting with the system, through a created account. These actions could range from sending in Officer Reports to finding the list of actives and new members in the website database. Some actors will have specific functions at their disposal, exclusive to the executive board.

Actor	*Actor's Goal*	*Use Case*
User(initiator)	Create and edit all accounts (as administrator)	UC-2
User(initiator)	Manage users own account	UC-4
User(initiator)	Query database for member information and view results	UC-1
User(initiator)	Send announcements via text (SMS)	UC-3
User(initiator)	Recover lost password	UC-5
User(initiator)	Randomly choose active members for extra duties	UC-6
User(initiator)	Access fraternity files materials	UC-8, UC-12
User(initiator)	Create events with an RSVP that send reminders to those that plan to attend	UC-10
User(initiator)	Create and vote in fraternity elections	UC-14
User(Initiator)	View/create announcements	UC-7
User(initiator)	Type and submit officer reports every week using the system.	UC-9
User(Initiator)	User logins into the System using their Username and Password	UC-11
User(Initiator)	User wants to generate a graphical representation of member's Fraternity lineage.	UC-13
Database(participating)	The Database will store and retrieve information based on requests made by the User.	UC-1, UC-2, UC-3, UC-4, UC-5, UC-6, UC-7, UC-9, UC-10, UC-11, UC-13, UC-14

Email Server(participating)	Web email server will send an email to the a Mobile Carriers' SMS gateway or to a User.	UC-2, UC-3, UC-6, UC-10
SMS Gateway(participating)	Mobile Carrier email to SMS gateway will take in an email from the web server email server and send out a text message to the desired phone number.	UC-3
Google Calendar(participating)	The Fraternity Google Calendar will be updated by the System when a new Event is created.	UC-10

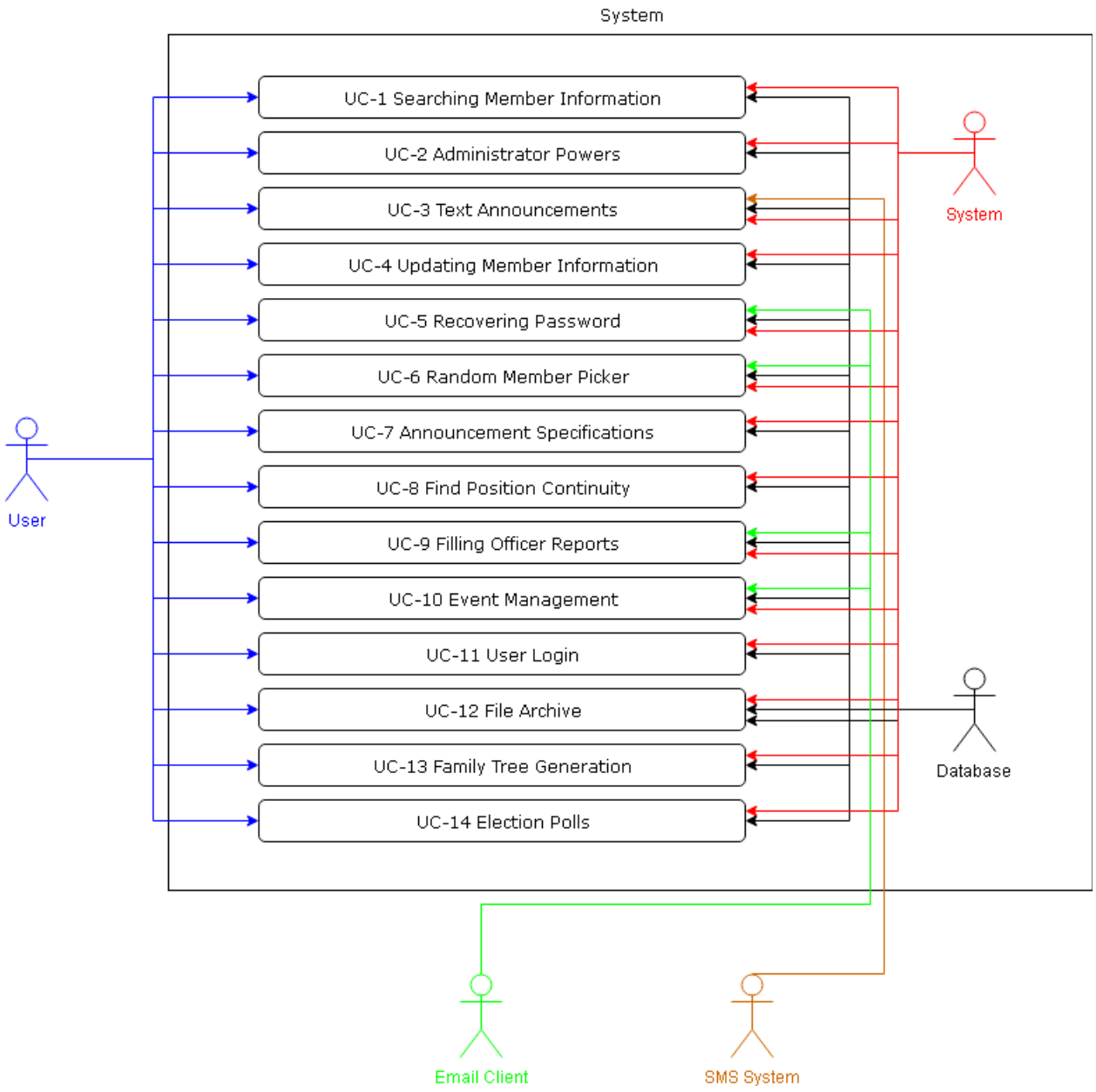
3.3 Use Cases

3.3.1 Casual Description

Use Case	Name	Description
UC-1	Searching Member Information	Users can query a database, based on the parameters set by the user. Any member will have read access to all information in the database. The user can choose to generate a spreadsheet from the results of this query. The searchable information will include: Employment history, fraternity position history, campus involvement, phone numbers, and other personal information about members.
UC-2	Using Administrator Powers	Administrative users can change member information and add new members.
UC-3	Creating a Text Announcement	Users can send urgent announcements via text message from the System to all active members
UC-4	Updating Member Information	Users can update their information in the Database on the system
UC-5	Recovering Password	Users can recover their account after forgotten password

UC-6	Picking Random Members	Users can randomly choose members from the Database for extra duties.
UC-7	Creating and Viewing Announcements	Users can access a filterable announcements feed with topics and create an announcement.
UC-8	Finding Position Information	Users who have a position will be able to access transition material and the description of their position responsibilities.
UC-9	Submitting Officer Reports	Users can send in weekly officer reports using the System.
UC-10	Managing Events	Users can create an event that users can RSVP to and the event creator can send email reminders to the RSVP-ed users. Created events will be linked to the fraternity Google Calendar
UC-11	Logging into the System	Users can login to the site with their unique Username and Password.
UC-12	Accessing File Archive	Users will have access to a organized fraternity file archive.
UC-13	Generation Family Tree	Users can generate a graphical fraternity family lineage by choosing a member.
UC-14	Voting in Elections	Users can vote in position elections on the System.

3.3.2 Use Case Diagram



3.3.3 Traceability Matrix

The Traceability Matrix allows the reader to cross the functional and non-functional requirements described earlier with the use cases. The importance of each Use Case is determined by the number of requirements associated with it and the average priority of its associated requirements. Lower priority weights are given to requirements that are more essential to the project. Non-functional requirements (REQ-19,20,21) are not taken into account when calculating these values. The tie-breaker for Use Cases will be if either is the only Use Case associated with a specific requirement e.g. UC-11 is the only Use Case associated with REQ-2 so it is given precedence over UC-9.

Note: Priority is represented as (!)

	!	UC-1	UC-2	UC-3	UC-4	UC-5	UC-6	UC-7	UC-8	UC-9	UC-10	UC-11	UC-12	UC-13	UC-14
REQ-1	1	X								X	X			X	
REQ-2	1											X			
REQ-3	1		X												
REQ-4	1			X											
REQ-5	2		X							X		X			
REQ-6	2				X										
REQ-7	2	X													
REQ-8	2			X							X				
REQ-9	2							X							
REQ-10	3	X							X			X			
REQ-11	3					X									
REQ-12	3						X								
REQ-13	3										X				

REQ-14	4										X			
REQ-15	4											X		
REQ-16	5												X	
REQ-17	5													X
REQ-18	3									X				
REQ-19	*1*	X			X		X		X		X	X		
REQ-20	*1*			X	X	X	X	X	X	X	X	X		X
REQ-21	*2*	X	X	X	X	X	X	X	X	X	X	X	X	X
Min PW		1	1	1	2	3	3	2	3	1	1	1	4	1	5
Total REQs		3	2	2	1	1	1	1	1	3	4	3	1	2	1
Avg. PW		2	1.5	1.5	2	3	3	2	3	2	2.5	2	4	3	5

3.3.4 Fully-Dressed Description

Use Case UC-1: Searching Member Information
Related Requirements: REQ-1, REQ-7, REQ-10, REQ-19, REQ-21, REQ-22 **Initiating Actor:** User **Participating Actors:** System, Database **Actor's Goal:** Display information from the Database **Preconditions:** • User must be logged into the Greek Life Member Management System. • The relevant information must be recorded in the Database.

Post-Conditions:
- The information searched for by the User will be displayed in tabular form.
- The User has the option to download the results as a Microsoft Excel spreadsheet.

Flow of Events for Main Success Scenario:
→ User selects the "Member Information" page from the System navigation bar.
← The system displays the "Member Information" page to the User.
→ On the "Member Information" page, the user selects the option to "Search Information."
← The system provides the user with a form containing options on what to search for and what filters to apply..
→ User makes their selections and presses the "Search" button.
← The System generates an SQL query from the User inputs and sends it to the Database. The results are displayed to the User in tabular form.

Flow of Events for Alternate Success Scenario (Spreadsheet Generation):
→ User selects the "Member Information" page from the System navigation bar.
← The system displays the "Member Information" page to the User.
→ On the "Member Information" page, the user selects the option to "Search Information."
← The system provides the user with a form containing options on what to search for and what filters to apply..
→ User makes their selections and presses the "Search" button.
← The System generates an SQL query from the User inputs and sends it to the Database. The results are displayed to the User in tabular form.
→ User then presses the "Generate Spreadsheet" button.
← The System creates a Microsoft Excel spreadsheet and offers it for download for the user.

Use Case UC-2: Using Administrator Powers

Related Requirements: REQ-3, REQ-5, REQ-21, REQ-22

Initiating Actor: User (Administrator)

Participating Actors: System, Database, Email Server

Actor's Goal: To add new Users to the System or update Users' information.

Preconditions:
- The User must be logged into the Greek Life Member Management System.
- The User must have Administrative access.

Post-Conditions:

- The initiating User will be displayed a message indicating successful completion of their action and a description of what was changed.
- A new User will be recorded in the Database.
- The new User will receive an email containing their Username and Password and instructions on how to access the System.
  OR
- The the chose User's information will be updated in the Database.

Flow of Events for Main Success Scenario:

→ User selects the "Admin" page from the System navigation bar.
← The system displays the "Admin" page to the User.
→ On the "Admin" page, the user selects the option to "Create a New User."
← The system provides the user with a form to enter the new user's pawprint and email.
→ User enters the pawprint and email of the new user.
← The System displays the information to the user for review.
→ User approves the creation or edits it.
← Once approved, the System creates an email going to the new user containing their system-generated password and instructions on how to access the System.
← The System records the new member in the Database.
← The email server sends the email to the new user.
← The system displays a success message to the User.

Flow of Events for Alternate Success Scenario (Update Member Information:

→ User selects the "Admin" page from the System navigation bar.
← The system displays the "Admin" page to the User.
→ On the "Admin" page, the user selects the option to "Update Member Info."
← The system provides the user with a form to edit the member's information.
→ User edits the relevant information.
← The System displays the information to the user for review.
→ User approves the change or edits it.
← The System records the new information in the Database.
← The system displays a success message to the User.

Flow of Events for Alternate Success Scenario (Cancel Member Creation):

→ User selects the "Admin" page from the System navigation bar.
← The system displays the "Admin" page to the User.
→ On the "Admin" page, the user selects the option to "Create a New User."
← The system provides the user with a form to enter the new user's pawprint and email.
→ User enters the pawprint and email of the new user.
← The System displays the information to the user for review.
→ User presses the "Cancel" button.
← The system exits the form and displays a cancel message to the User.

Use Case UC-3: Creating a Text Announcement

Related Requirements: REQ-4, REQ-8, REQ-20. REQ-21, REQ-22

Initiating Actor: User

Participating Actors: System, Email Server, SMS Gateway

Actor's Goal: To send a text message announcement to all active Fraternity members.

Preconditions:
- User must be logged into the System.
- Active members must have provided their phone numbers and mobile carriers to the Database.

Post-Conditions:
- The initiating User will be displayed a message indicating successful completion of their action.
- Each current member of the fraternity will receive a SMS message containing the announcement.

Flow of Events for Main Success Scenario:
→ User selects the "Announcements" page from the System navigation bar.
← The system displays the "Announcements" page to the User.
→ On the "Announcements" page, the user selects the option to "Create a New Announcement."
← The system provides the user with a form to enter their message.
→ User enters their message, chooses its topic, and selects the "Also Send as Text" option.
← The System displays the message to the user for review.
→ User approves the message or edits it.
← Once approved, the System creates an email addressed to the SMS Gateway of each active member.
← The System records the announcement in the Database.
← The email server sends the email to the SMS Gateways.
← The SMS Gateways send the announcement as SMS messages to the active members.
← The System displays a success message to the User.

Flow of Events for Alternate Success Scenario (User Cancels message):
→ User selects the "Announcements" page from the System navigation bar
← The system displays the "Announcements" page to the User.
→ On the "Announcements" page, the user selects the option to "Create a New Announcement."
← The system provides the user with a a form to enter their message.
→ User enters their message, chooses its topic, and selects the "Also Send as Text" option.

← The System displays the message to the user for review.
→ User decides not to send the message and selects the “Cancel” button.
← The System deletes the message and displays a cancel message to the User.

Use Case UC-11: Logging into the System

Related Requirements: REQ-2, REQ-5, REQ-10, REQ-19, REQ-20, REQ-21, REQ-22

Initiating Actor: All Users

Participating Actors: System, Database

Actor’s Goal: Gain access to the Greek Life Member Management System.

Preconditions:
- The user must have had an account created for them by an administrator who provided the user with a Username and Password.
- The user must know their Username and Password
- The user must be able to access the Login Page over the Internet.

Post-Conditions:
- The user has access to the Greek Life Member Management System until they log out.
- The user’s position and name are retrieved from the database .
- If the user is a privileged user (position holder, administrator), they have access to additional features.

Flow of Events for Main Success Scenario:
→ User connects to the Login page of the Greek Life Member Management System over https.
→ User enters their Username and Password into the fields and presses the login button.
← The system uses a hash function on the password and queries the Database for the user.
← The Username and hashed Password are found in the Database; the user is granted access and redirected to the homepage.
← The System records the Login in the Database.

Flow of Events for Alternate Success Scenario (Unknown User Denied Access):
→ User connects to the Login page of the Greek Life Member Management System over https.
→ User enters their Username and Password into the fields and presses the login button.
← The system uses a hash function on the password and queries the Database for the

user.
← The Username and hashed Password are not found in the Database, the user is displayed an error message saying their Username or Password is incorrect. They are not granted access to the System.

3.4 System Sequence Diagrams

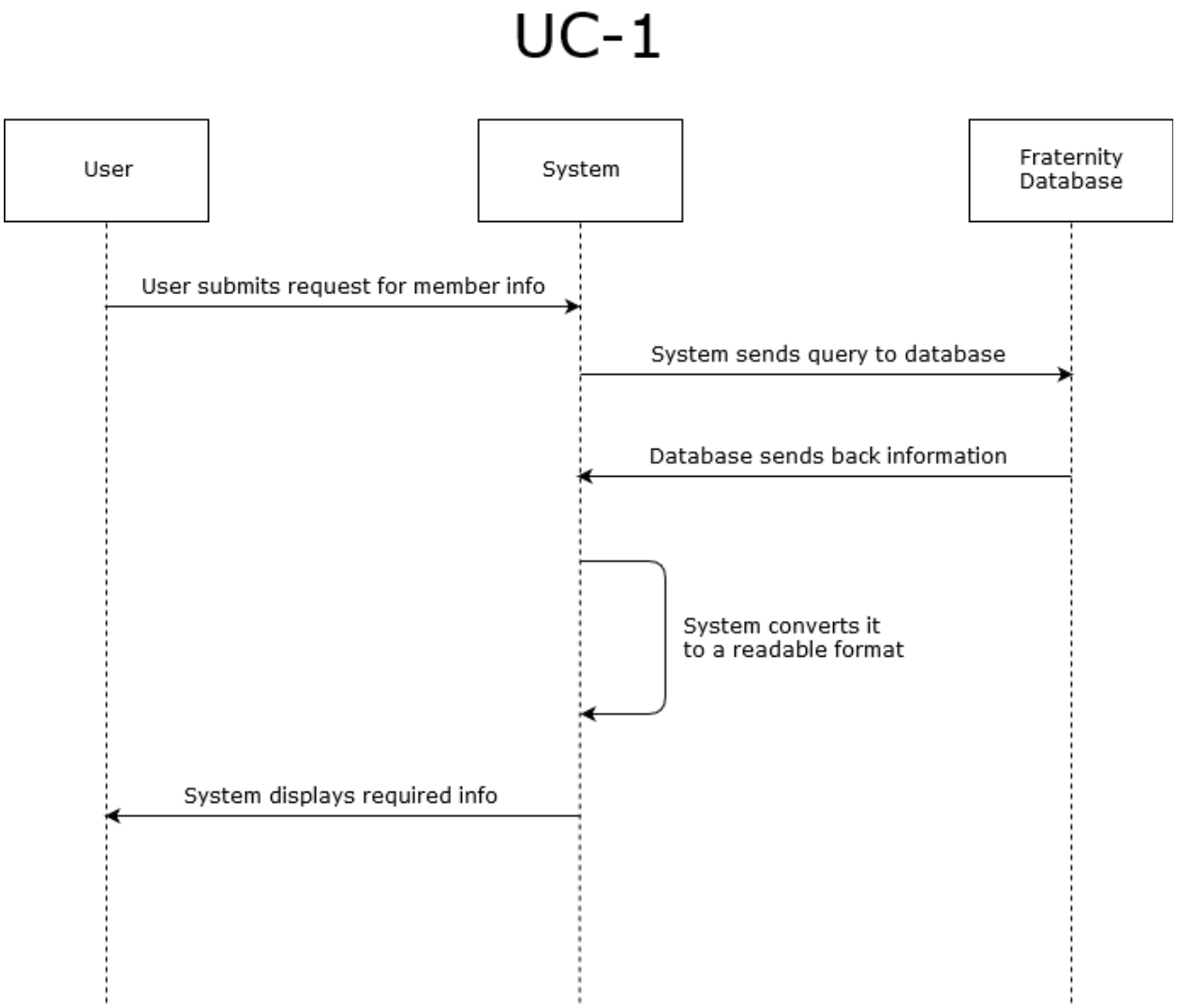


Figure 3.4.1 - Sequence Diagram of Use Case 1

Figure 3.4.1 shows the sequence diagram of Use Case 1 which is “Searching Member Information”. The objective in this diagram is to easily view data that is dynamically retrieved by the system, working hand in hand with the fraternity database. It is important because of the integral role that certain groups of members play within internal operations. The diagram corresponds to the Main Success Scenario.

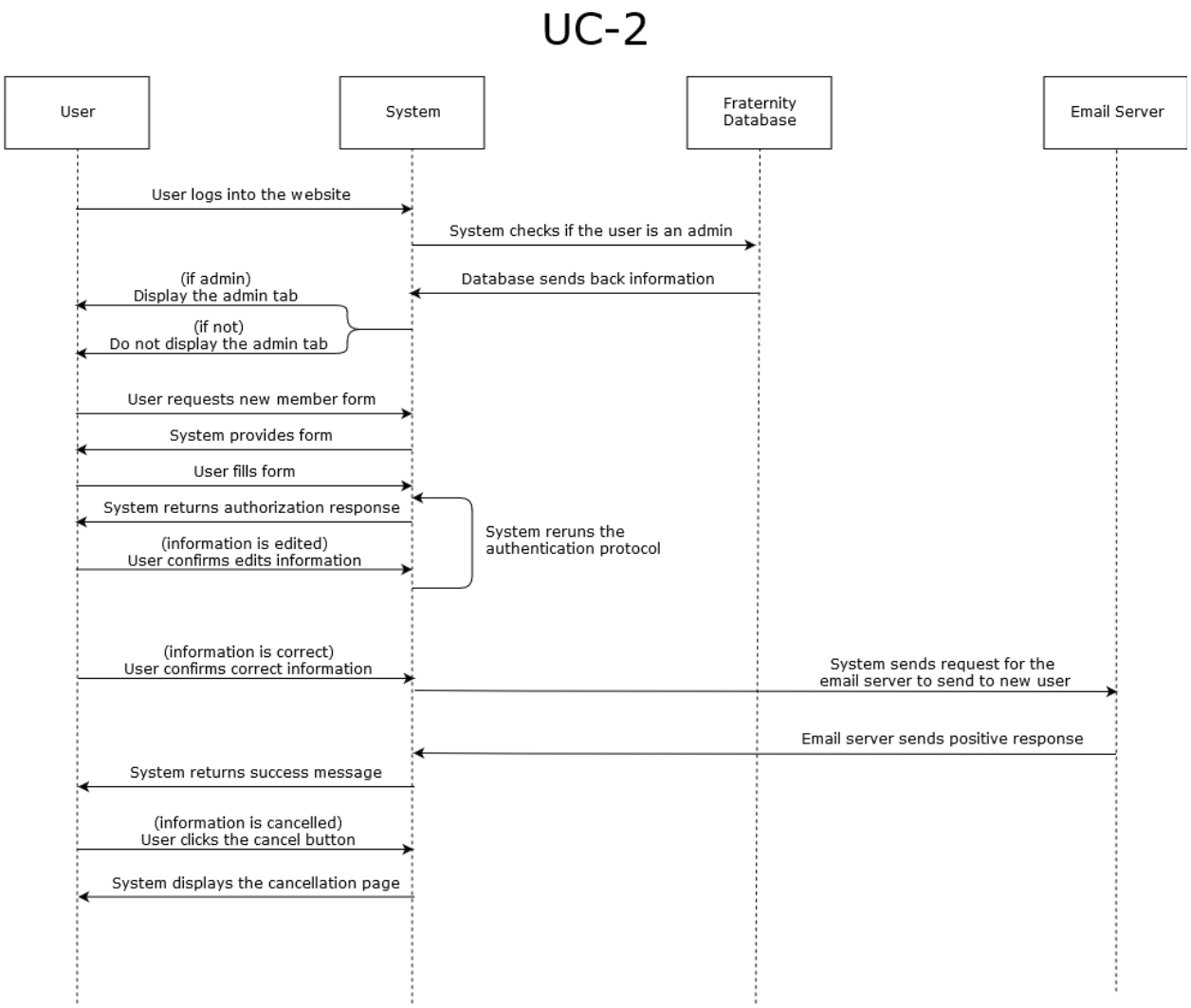


Figure 3.4.2 - Sequence Diagram of Use Case 2

Figure 3.4.2 shows the sequence diagram of Use Case 2 which is “Using Administrator Powers”. The objective of this diagram is to layout many possible outcomes of an admin only form. This use case relies on not only the fraternity database but also the email server to send the final confirmation message that the user has been sent the email. The diagram corresponds to both Main and Alternate Success Scenarios.

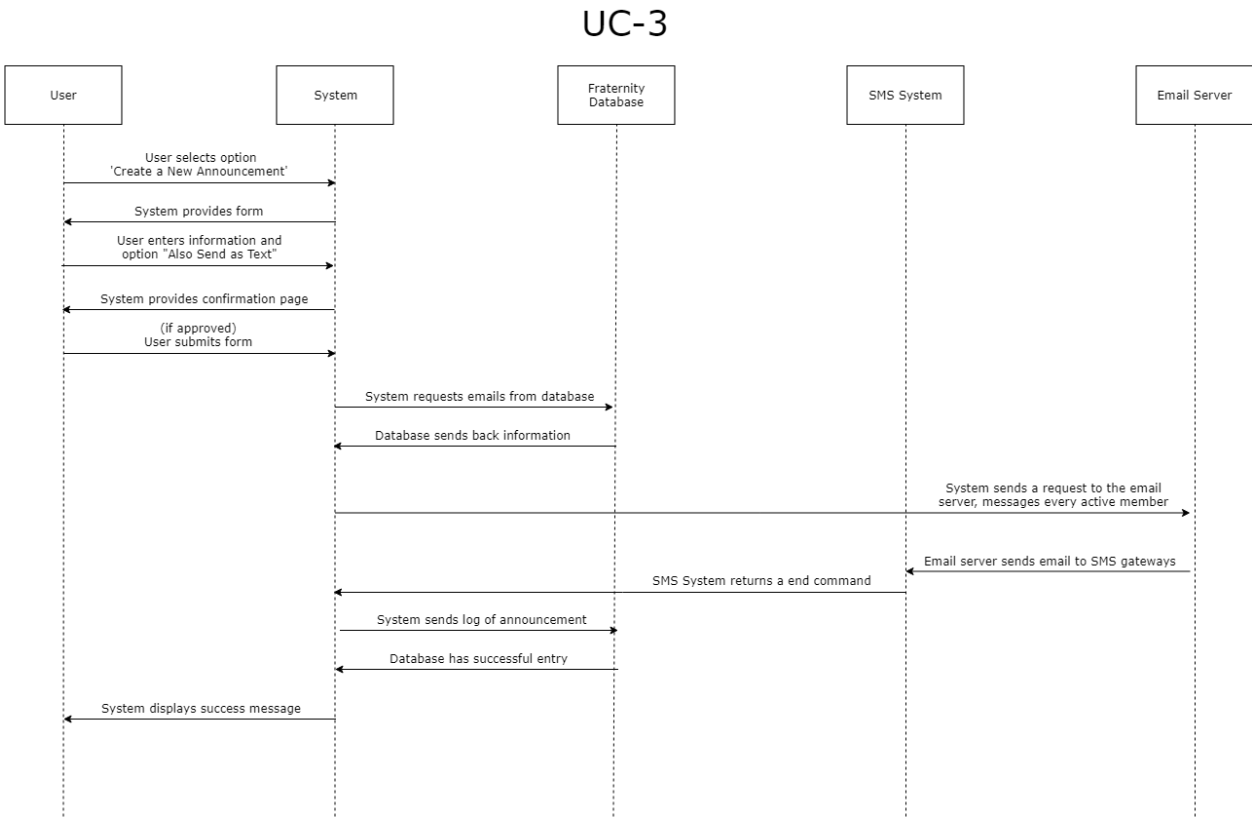


Figure 3.4.3 - Sequence Diagram of Use Case 3

Figure 3.4.3 shows the sequence diagram of Use Case 3 which is “Creating a Text Announcements”. The objective of this diagram is to create a trail of how the group plans on accomplishing the task of system input to mass text feature. This feature would be used mainly to send very important announcement and would be an executive board exclusive feature. The diagram corresponds to both Main and Alternate Success Scenarios.

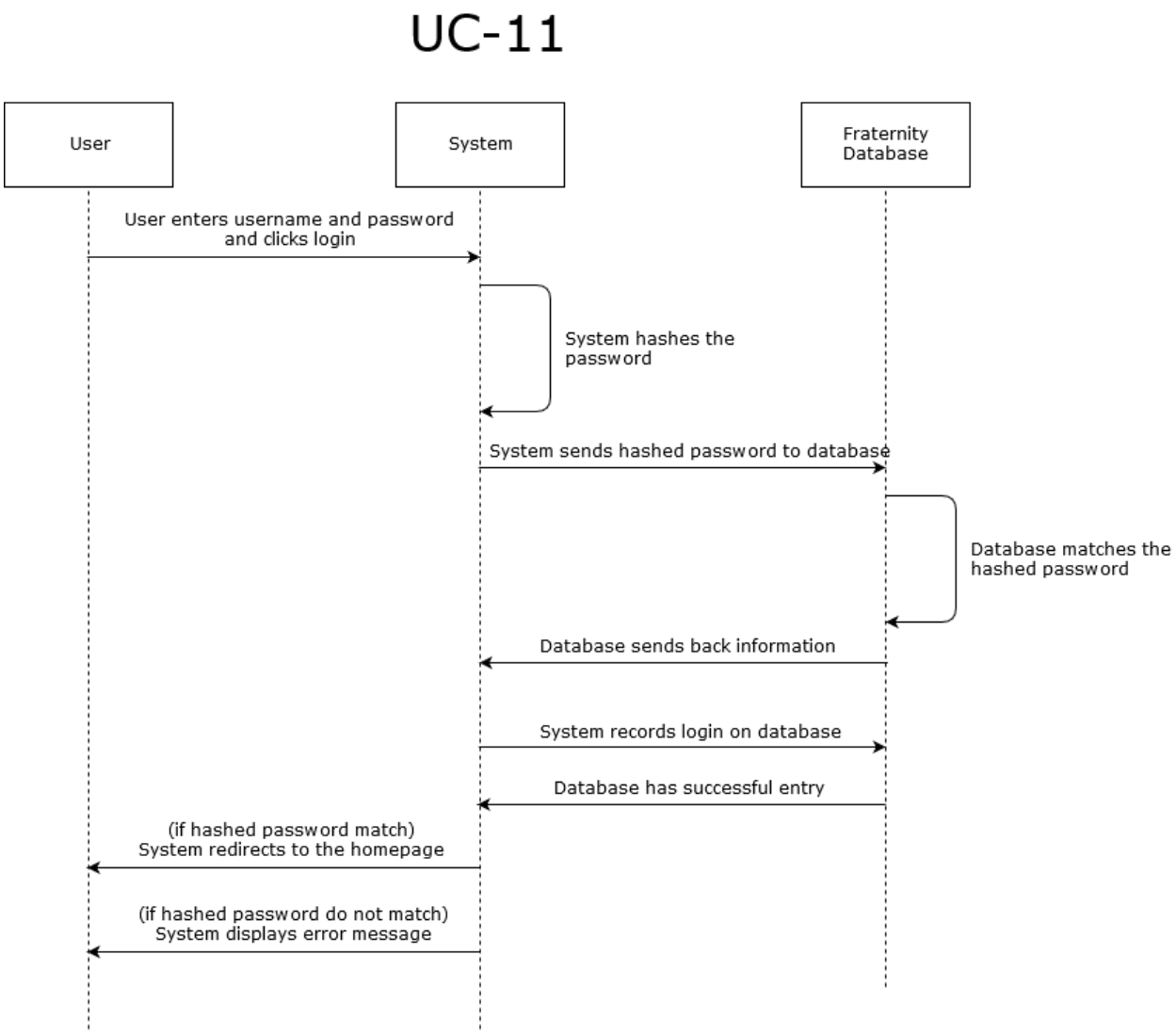


Figure 3.4.4 - Sequence Diagram of Use Case 11

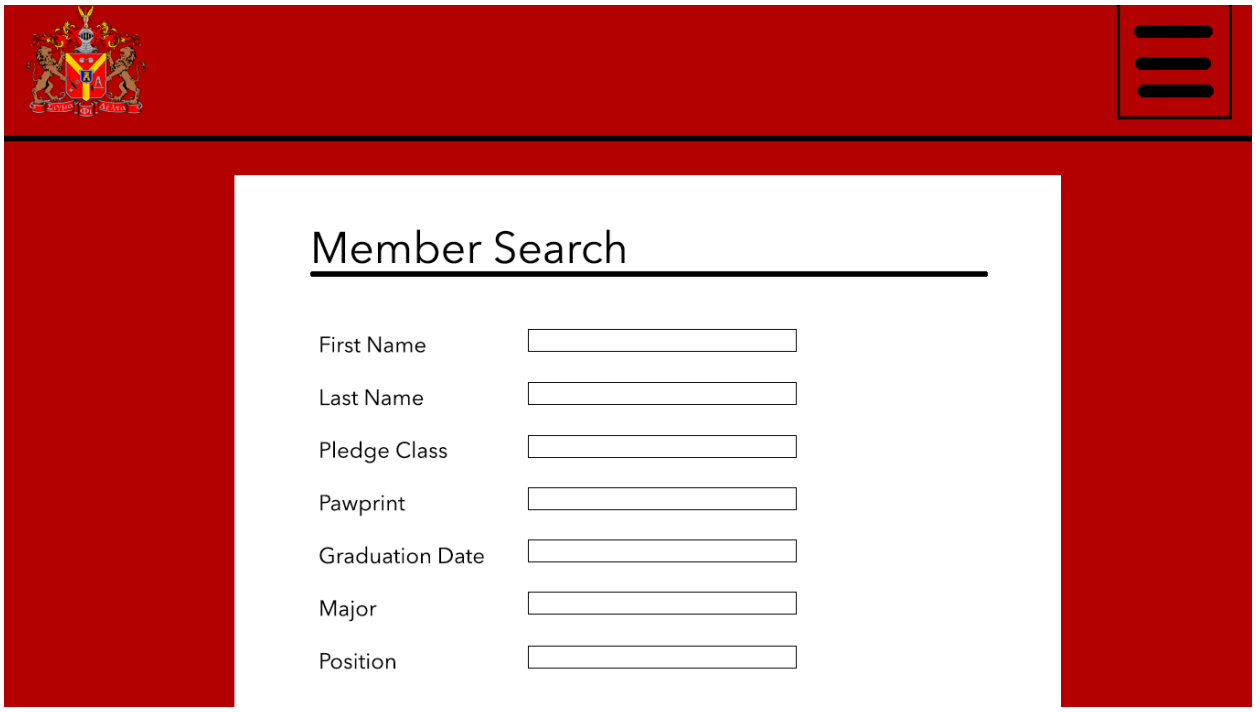
Figure 3.4.4 shows the sequence diagram of Use Case 11 which is “Logging into the System”. The objective of this diagram is to explain the importance of a secure login for the project. In order to access many of the features that are explained in the system requirements, the database needs to be password protected. This ensures secure entries and the safeguarding of data. The diagram corresponds to both Main and Alternate Success Scenarios.

4 User Interface Specification

4.1 Preliminary Design

4.1.1 Use Case UC-1: Searching Member Information

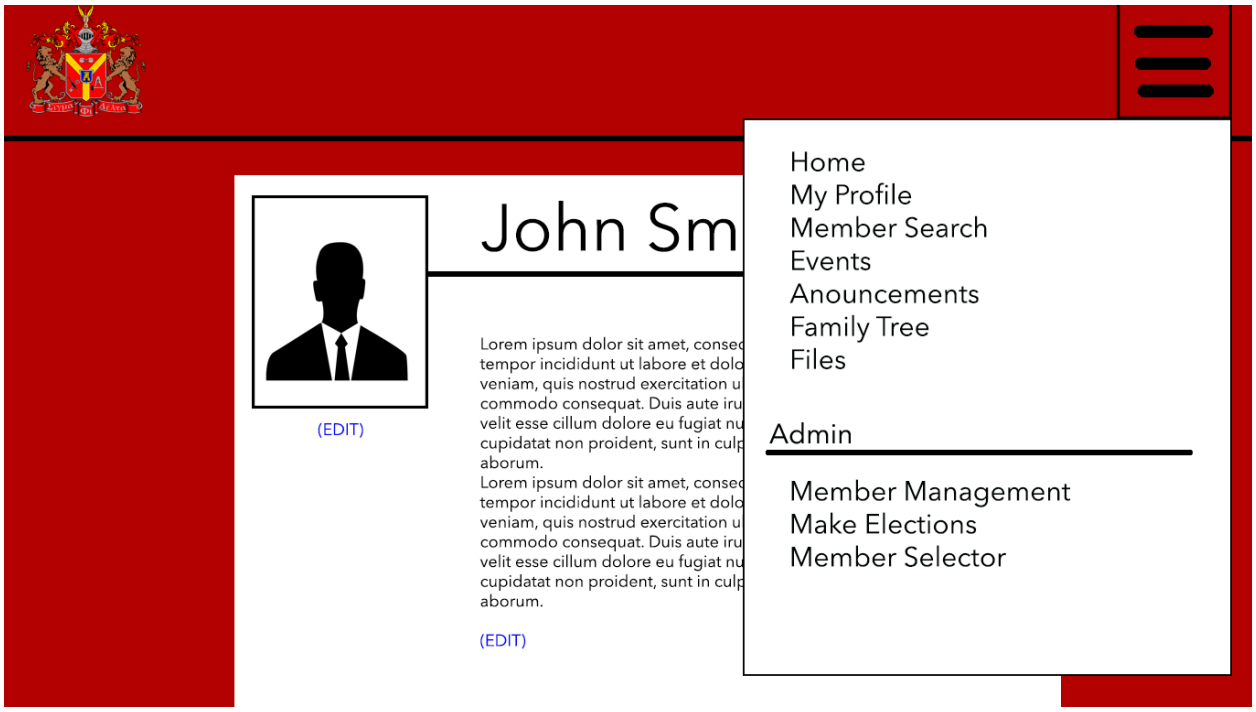
This is the member search page. There will be multiple fields for narrowing search results from the database. The search will be initiated by a submit button at the bottom of the page. The returned list will display on the screen and the name of the member can be selected and you will be taken to their profile (UC-4 but without the option to edit).



4.1.2 Use Case UC-2: Using Administrator Powers

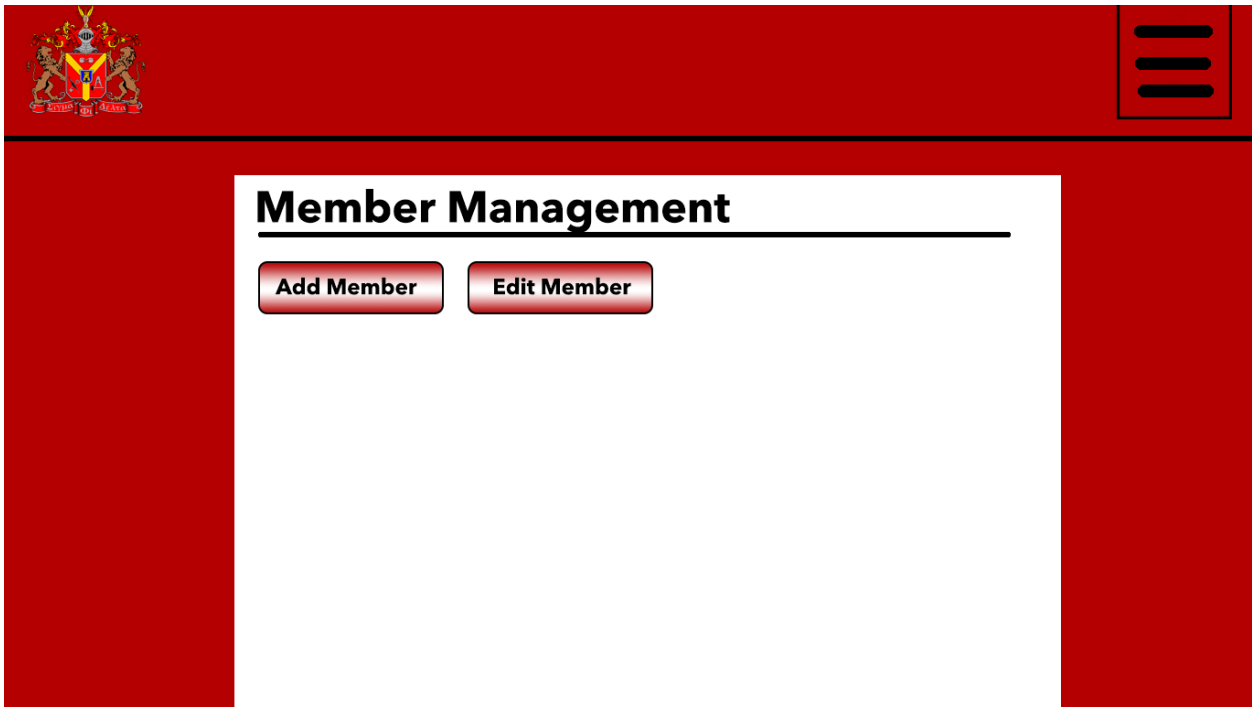
4.1.2.1 Dropdown Menu

This page demonstrates the extra abilities that members with administrative privileges will have. The extra abilities will take the administrator to the desired page.

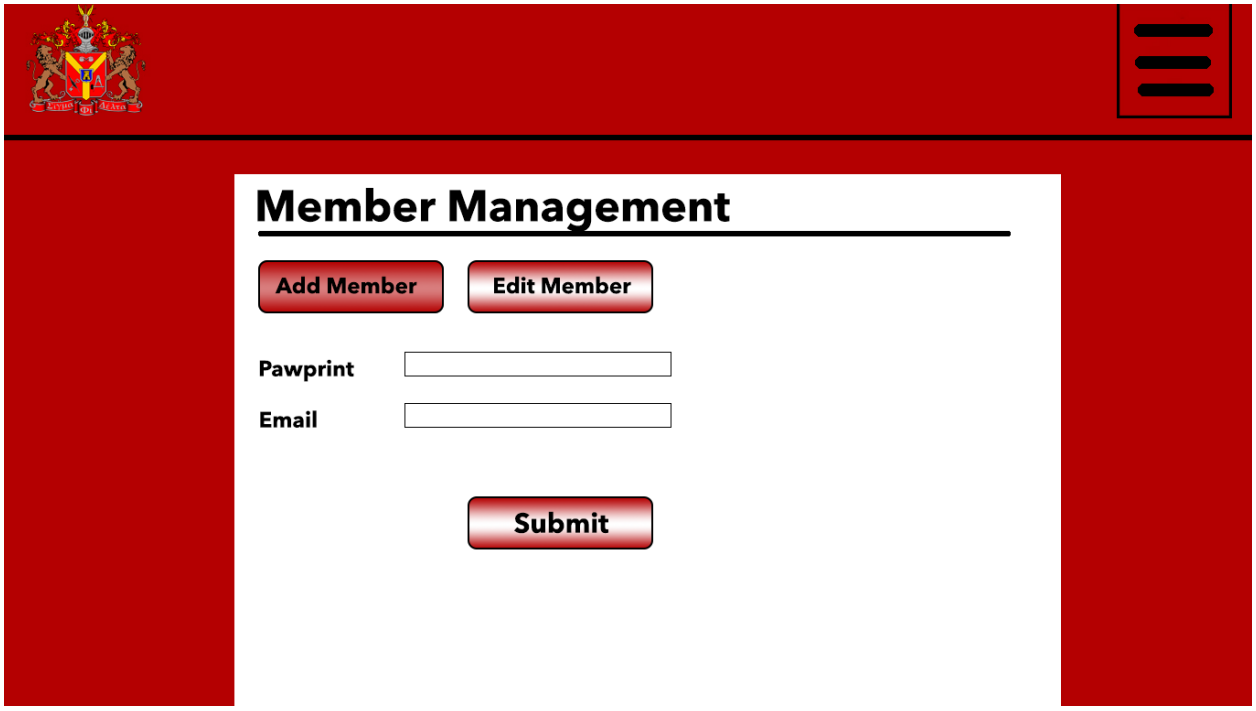


4.1.2.2 Add/Edit Members

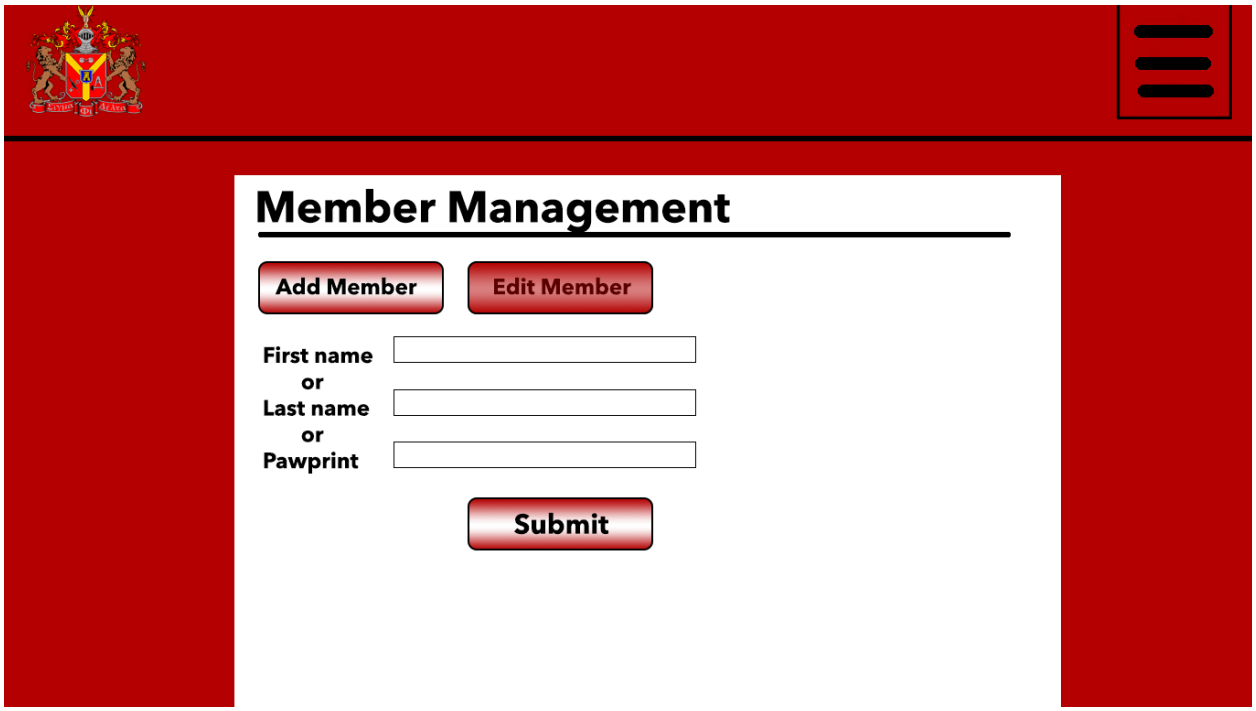
The member management page will have two options. Add member is used to create a new member and edit member is used to edit an existing member.



Pressing the add member button gives the user two text fields to enter. The first is used for the University of Missouri pawprint of the new member. The second is for their email. There is a submit button at the bottom to enter the information into the database.



Pressing the edit member button gives the user the ability to search for a member with three text fields: first name, last name, or pawprint. Only one field is necessary. There is a submit button at the bottom of the page to query the database. A list of members is then displayed and when selected the administrator user is taken to the members editable profile (UC-4).

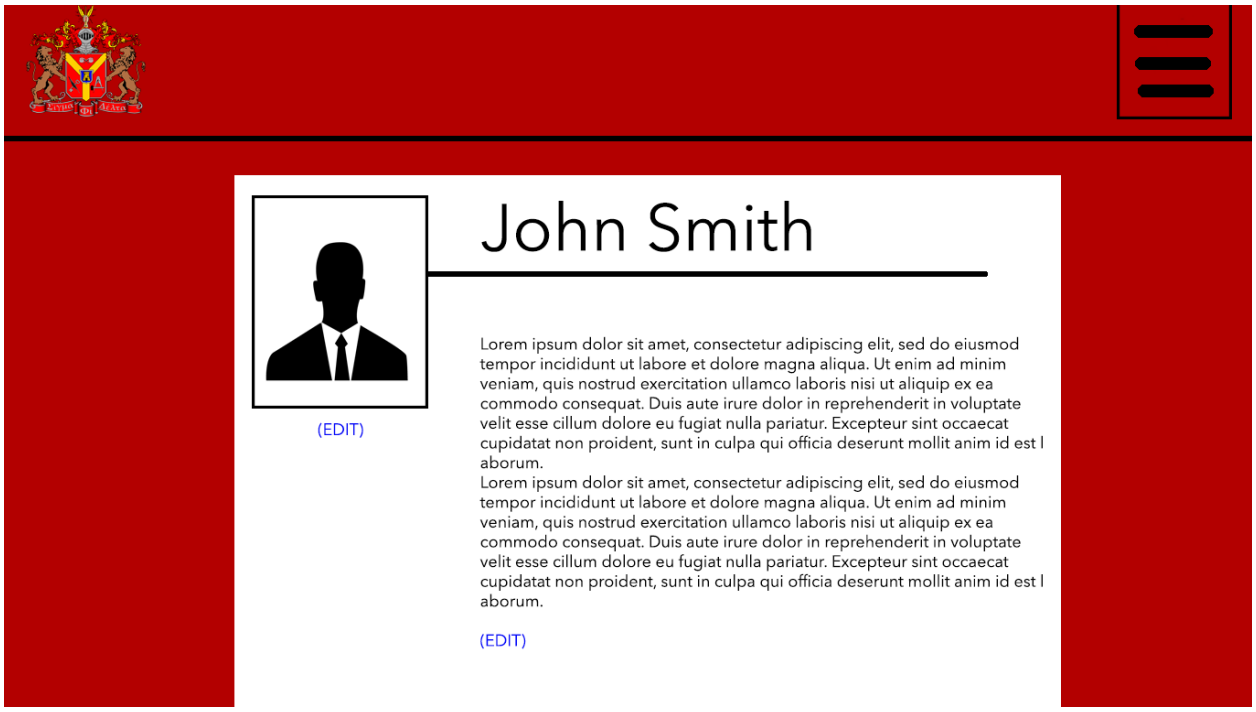


4.1.3 Use Case UC-3: Creating a Text Announcement

This case uses the same announcement system as that used in UC-7.

4.1.4 Use Case UC-4: Updating Member Information

This page is the standard page that a user will see after logging in to the system. Their ability to edit their information will be accessed by clicking the edit link next to the field they wish to edit. The page will stay relatively the same except the text will be displayed in a text entry field with a submit button under it. When editing is complete the user will select the submit button to update the database and return to this updated screen.

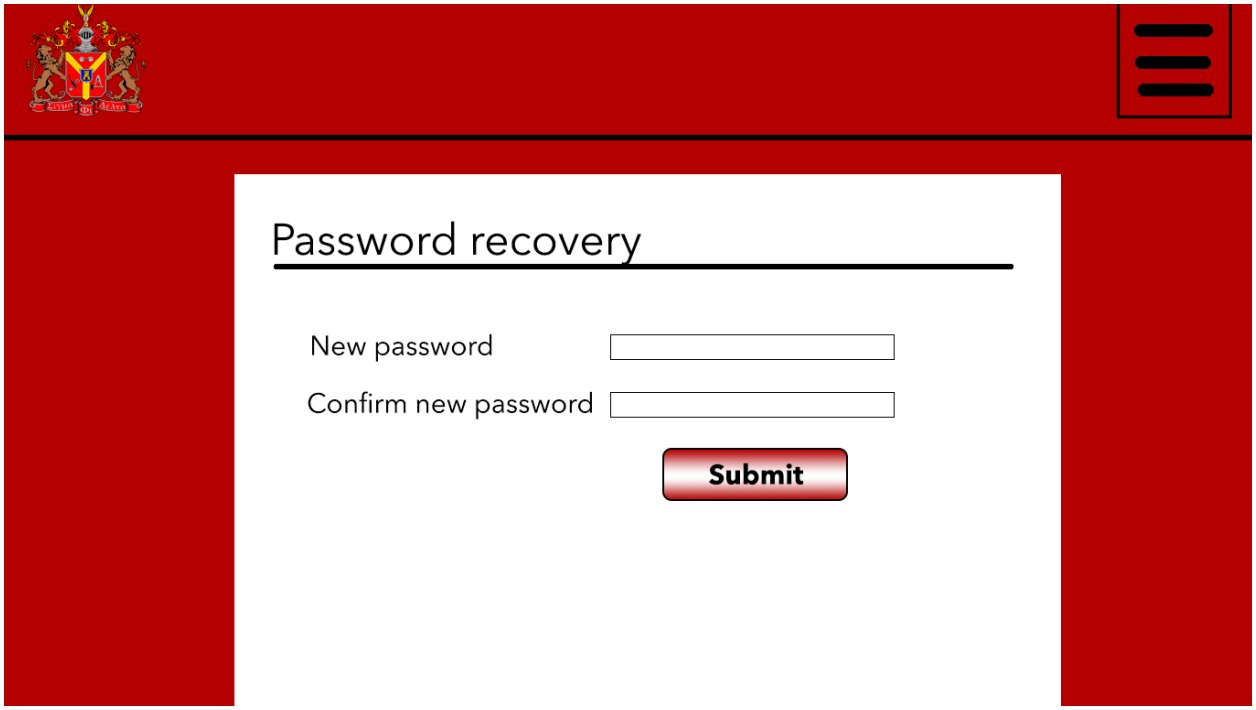


4.1.5 Use Case UC-5: Recovering Password

The following images are the pages that members will see when they wish to recover their account after a password is lost. Users will first enter the username for the account that they wish to recover. They will then be given a set of security questions. After answering the first question and submitting via the submit button they will be taken to the next question. After answering and submitting the second question correctly they will be given access to the password reset page and after submitting their account will be able to be accessed again with the new password.

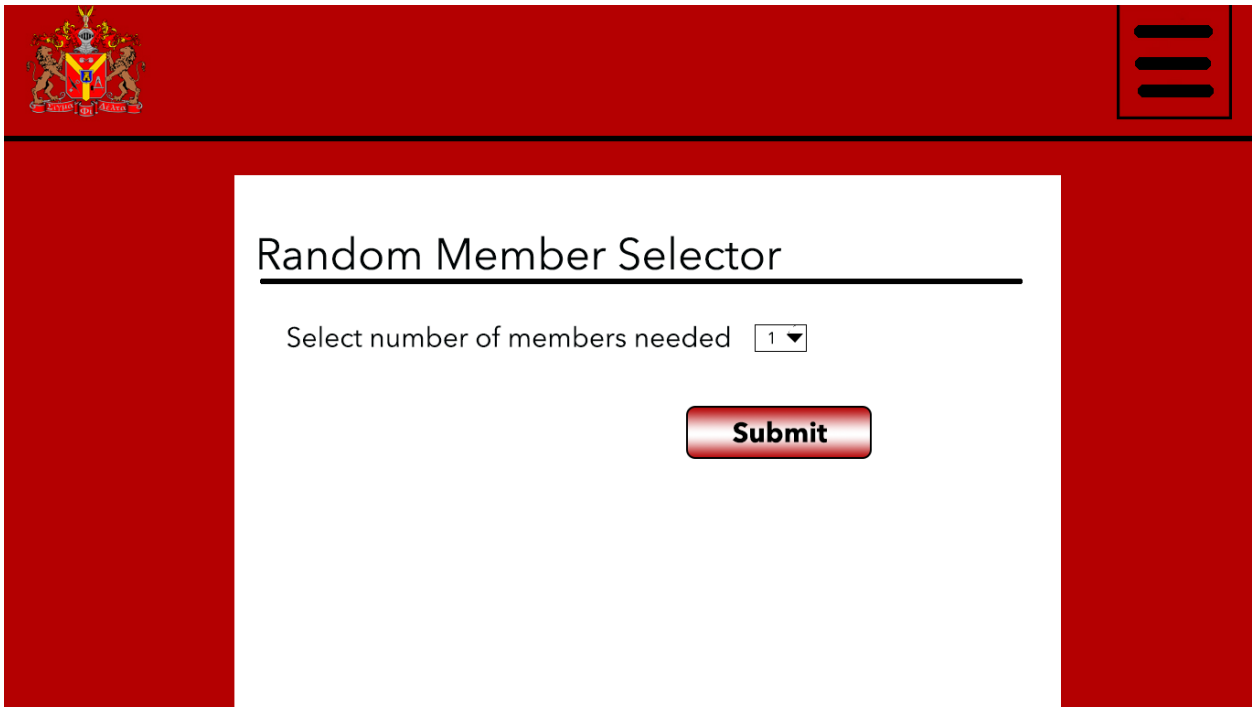

Password recovery
Username
Submit


Password recovery
Mothers maiden name
Submit



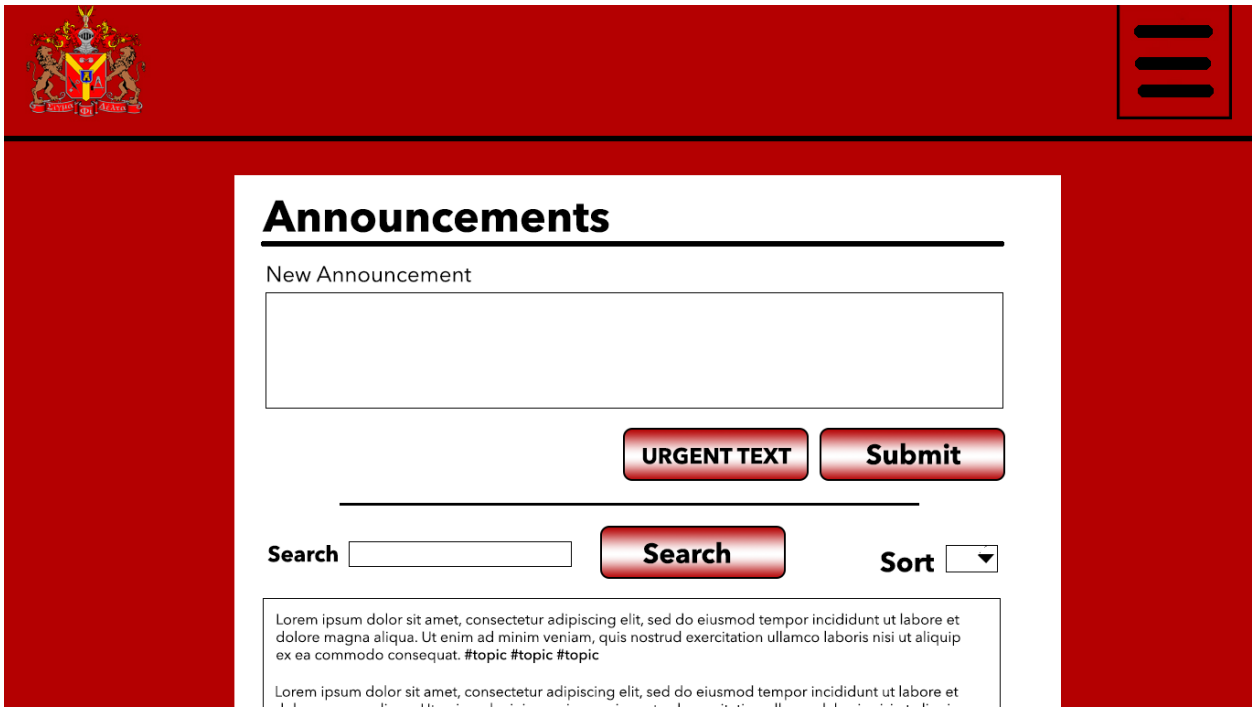
4.1.6 Use Case UC-6: Picking Random Members

The random member picker will be a simple dropdown menu with numbers from 1 to the amount of active members in the database. The user will press the submit button and a list of randomly selected members will be generated.



4.1.7 Use Case UC-7: Creating and Viewing Announcements

The announcements page will have a fillable text area for creating a new announcement. The announcement will be posted to the lower portion of the page upon selection of the submit button. In order to post the announcement and send a message to other members phones the urgent text button will be selected. The post history is keyword searchable by entering the keywords into the search box and selecting the search button. The list will be sortable by different options such as date, username, etc. The use of a hashtag(#) system will allow for topics to be created and quickly filtered by simply clicking on the desired tag name.



4.1.8 Use Case UC-8: Finding Position Information

This case uses the same file archive as that used in UC-12.

4.1.9 Use Case UC-9: Submitting Officer Reports

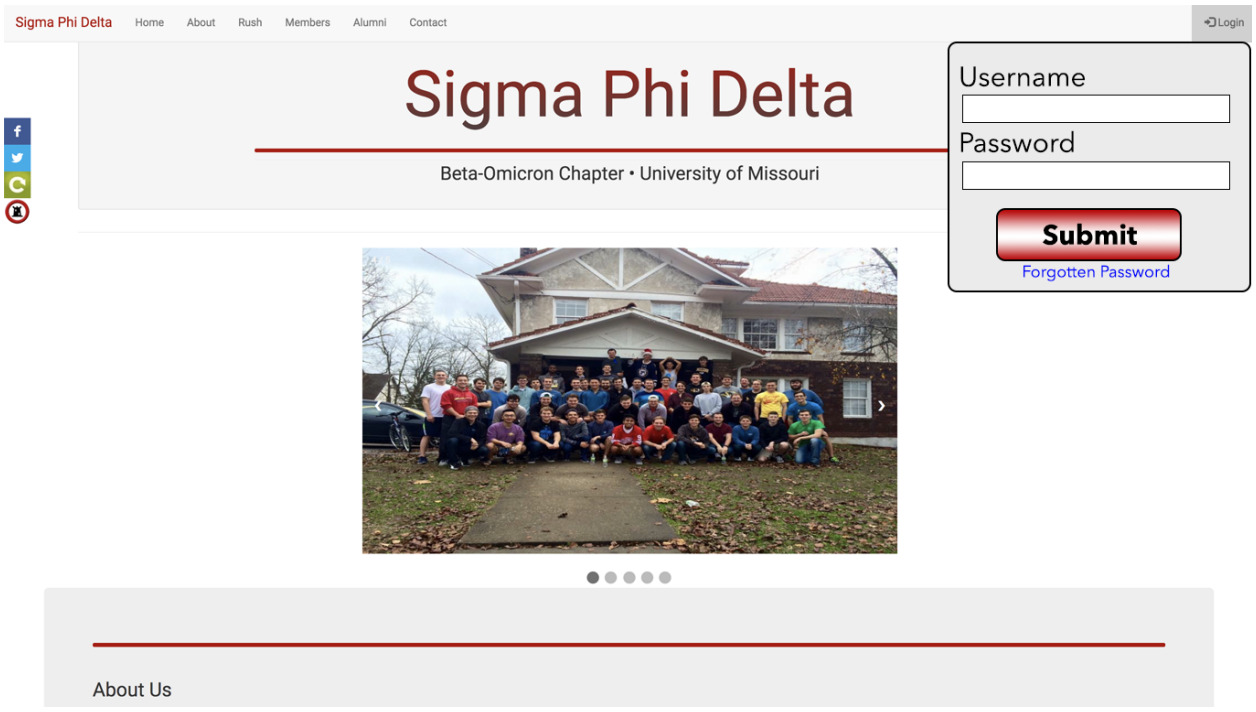
This case uses the same file archive as that used in UC-12.

4.1.10 Use Case UC-10: Managing Events

Users can use this page to create an event and request an RSVP. The ‘Who’ dropdown menu will have the names of members and an option to select all of them. Users will select the names of the people they are inviting to the event. The ‘What’ text entry is for giving a description of the event. The drop down menu for ‘Date’ will be a standard calendar popup for the user to select the date of the event. The ‘Time’ dropdown will be a list of times separated into 15 minute intervals. The ‘Where’ textbox is for entering a location or address. The user will then select the submit button and an e-mail will be sent to each member that was selected and request an RSVP.

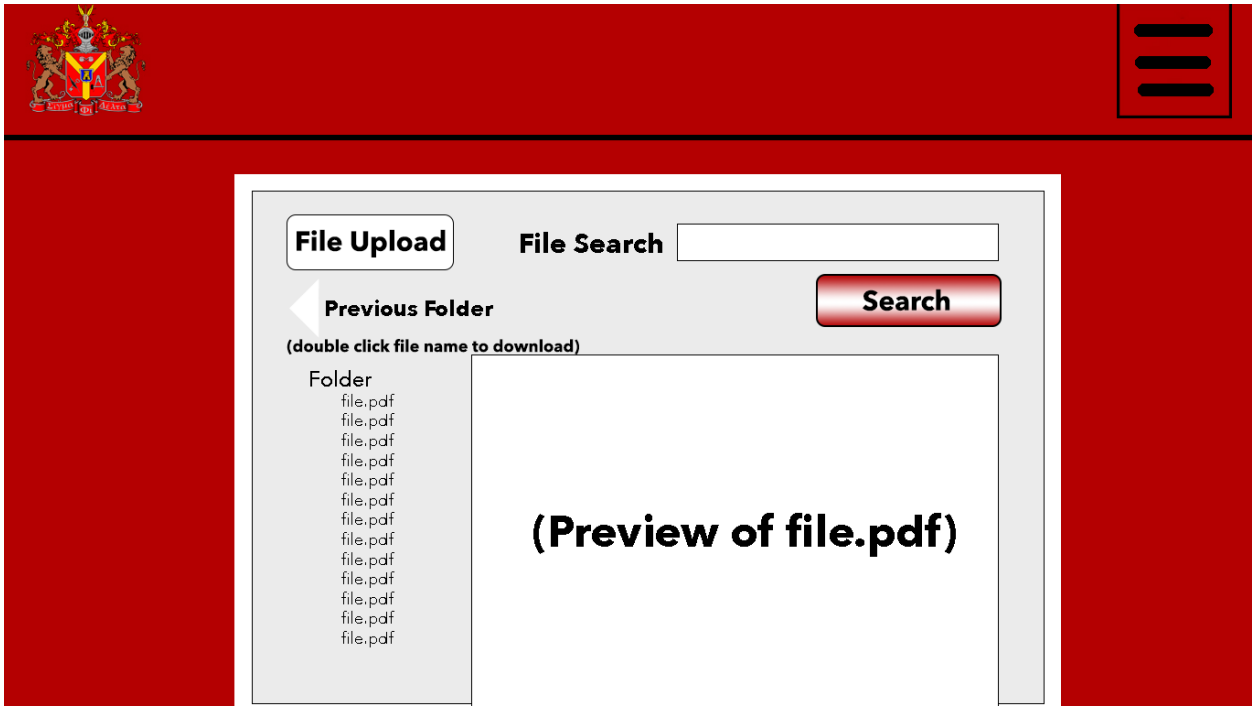
4.1.11 Use Case UC-11: Logging into the System

The user login will be a drop down window on the main fraternity page the user will be challenged for their username and password with a submit button. A link will be below the submit button for account recovery detailed in UC-5.



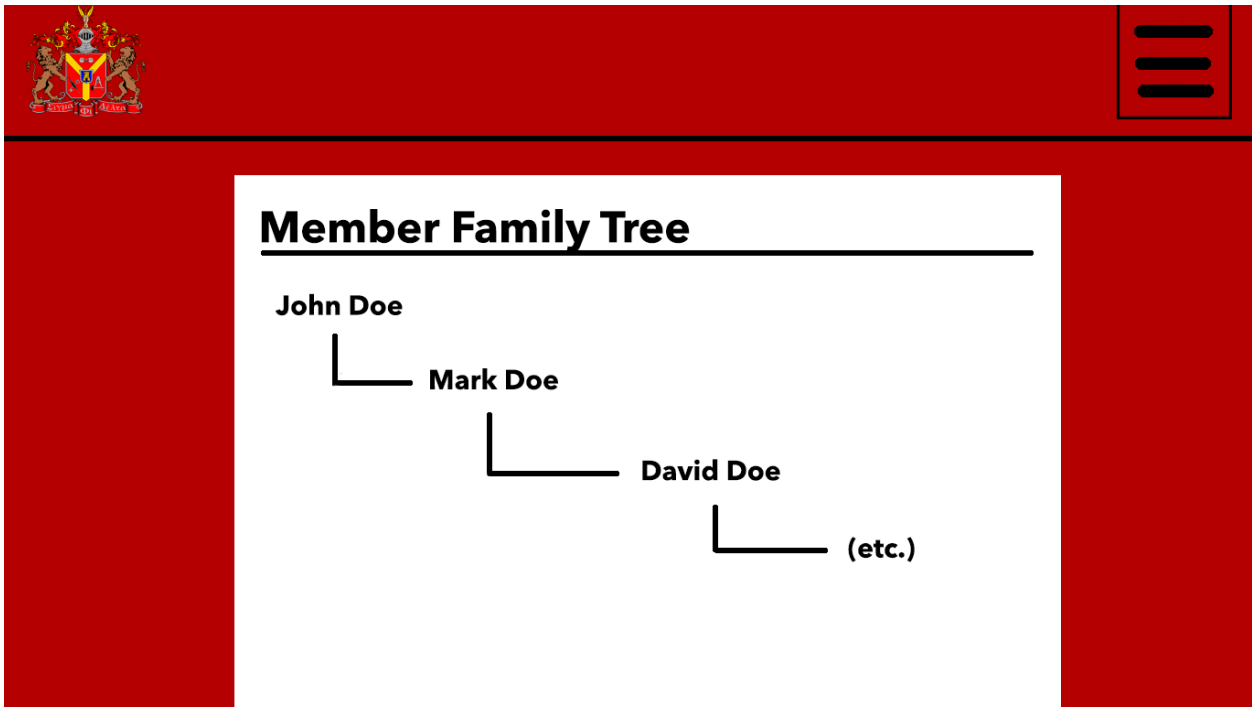
4.1.12 Use Case UC-12: Accessing File Archive

The file archive will be presented with a number of options. The first is a file upload button. When pressed a standard upload prompt will be displayed allowing a user to upload a file, from their computer, into the folder they are currently viewing. There is an option to search inside the folder that the user is in. They fill in the information they are looking for and then click the search button. The user will navigate in the file archive by selecting what file or folder they would like to view on the lower left list. To navigate back they will select the left facing arrow or the words just above the list. A display on the lower right will give the user a preview of the file they are selecting and a double-click of the left mouse on the file name will initiate a download of that file.



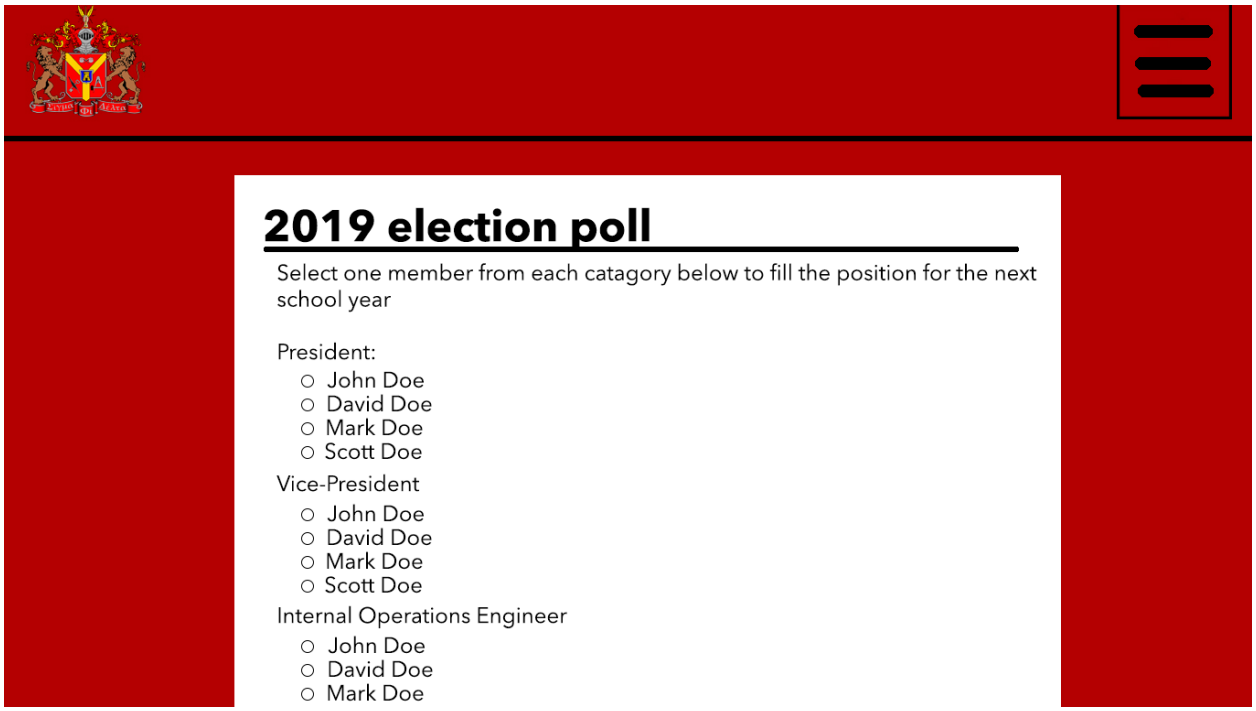
4.1.13 Use Case UC-13: Generating Family Tree

The family tree will be created automatically when the user selects it from their drop down menu. The only interaction users will have is selecting a member from their tree and be taken to that members profile.



4.1.14 Use Case UC-14: Voting in Elections Polls

Election polls will will be created by an administrator. This will be the page presented to users for their votes to be tallied. They will select a radial button coinciding with the person they wish to register a vote for (only one radial per section will be able to be selected). The bottom of the page will have a submit button for sending the form to the server to be retrieved when the results are to be calculated.



4.2 User Effort Estimation

4.2.1 Scenario 1

Generate List of Active Members' Phone Numbers (UC-1):

1. Select "Member Search" from the navigation menu.
2. Select "Multi Member Search" tab.
3. Choose "Current Members" from the "Group" dropdown list.
4. Choose "Phone" from the "Info" dropdown list.
5. Choose "Last Name" from the "Sort" dropdown list.
6. Press "Search" button.
7. Press "Generate Spreadsheet" button

   Clicks: ~10; Keystrokes: ~0

4.2.2 Scenario 2

Adding a Member (UC-2):

1. Select "Member Management" from the navigation menu.
2. Click the "Add Member" button.
3. Click the "pawprint" field.
4. Entre the pawprint.

5. Click the email field.
6. Enter the email
7. Click “Create” button.
8. Click “Confirm” button.

   Clicks: ~6; Keystrokes: ~30

4.2.3 Scenario 3

Send Text Announcement that Chapter has Moved (UC-3):

1. Select “Announcements” from navigation menu.
2. On “Announcements” page, enter your message in the “New Announcement” field.
3. Select a topic from the dropdown list.
4. Click “Urgent Text” button.
5. Click the “Confirm” button.

   Clicks: ~5; Keystrokes: ~150

4.2.4 Scenario 4

User Login To System (UC-11):

1. From spdmizzou.com homepage, select “Login”
2. Click the “username” field
3. Enter username
4. Click the “password” field
5. Enter password
6. Press <Enter> to login

   Clicks: ~3; Keystrokes: ~20

5 Project Management

Each member of our team will be assigned functional features that they are responsible for developing as listed below. Responsibilities will be be divided so that the overall difficulty of the assigned features is equivalent for each team member. Consideration will also be made to each team member's strengths and interests, but will not be the only deciding factor. Each member will manage their code using the github repository.

Group communication will be done using GroupMe and group meetings are held weekly on Tuesday nights at 6:30 when all team members can be in attendance. In the weekly meetings, each group member will provide an update on their assigned tasks, including any difficulties keeping them from completing their tasks for the week and projected completion dates. During the meetings, tasks will be assigned to be completed by the next meeting and progress on course deadlines will be assessed. Meetings will be conducted according to the meeting agenda document and recorded by Christopher Whetsel. An archive of the meeting agendas will be kept on the blog.

Code contributions will be managed with the project's public Github repository. Team members will work on their individual and use git to share and merge everyone's work. Diagrams were created using Draw.io and Abode Photoshop was used for the interface designs.

Reports are created using a shared Google Doc. The theme and style used for each report is the same and the content of each report follows a common format for spacing, headers, and font. Once completed, the report is converted to a PDF, turned in on Canvas, and made available for download on the project website.

5.1 Breakdown of Responsibilities

Jason Pulis will own the features to allow sending urgent announcements as text messages, to facilitate secure login and user access to features, and to create the announcement feed. Jason will be in charge of managing the project website and blog. He will also get a https certificate for the client and development server.

Christopher Whetsel will own the features to generate a graphical fraternity family tree, to allow users to search the database and generate spreadsheets, to create and manage events, to provide voting polls for fraternity elections. Christopher will also work on the Documentation of how to set up the project for other Greek Organisations.

Mike Winkelmann will own the features to have an organized file archive, to add and edit information to the database from the web interface, to allow fraternity position

holders to view documents related to their position. Mike will also work on creating a UI theme for the website.

References

1."Use case", Wikipedia http://en.wikipedia.org/wiki/Use_case

2."System requirements", Wikipedia http://en.wikipedia.org/wiki/System_requirements

3."User interface specification", Wikipedia
http://en.wikipedia.org/wiki/User_interface_specification

4. “System Sequence Diagram”, Wikipedia
https://en.wikipedia.org/wiki/System_sequence_diagram

5."Software Engineering book", Ivan Marsic
http://www.ece.rutgers.edu/~marsic/books/SE/book-SE_marsic.pdf

6. “Sigma Phi Delta National - Home” https://sigmaphidelta.2stayconnected.com/

7. “**ΣΦΔ** Beta-Omicron” http://spdmizzou.com/