**Fraternity Information Database**

**Description:**

This database helps Fraternities at Salisbury University organize data they use in their everyday life. This database will keep track of every member in each fraternity, alumni starting from the year we release the database, dues that each fraternity member owes/owed, community service events attended, potential new members in each fraternity and the attendance of each member in every fraternity. We will implement our website by using Linux, MySQL, php (LAMP), HTML and CSS for all topics listed above. All six fraternities at Salisbury University are the only targeted audience for the admin section of the database with the exception of a global admin. Our database will also contain a non-admin page that allows all users to preview information about every fraternity and their current active members. Those not in the fraternity will not have access to the admin pages of the selected fraternities. These features will allow executive positions in fraternities to organize data that is all in one place at one time without fear of losing data. Users can utilize the ability to search, delete, and insert everyday data into this database rather than less reliable methods such as paper/excel sheets.

The interface layout will be a webpage with outside customer features that do not ask for a login and admin features for the fraternities which will be provided by a login page. The homepage will be for outside users that are not in a fraternity. This page will provide information such as an “About Us” page and a list of active members for each fraternity at Salisbury University. They will also be provided with more information about Salisbury University fraternity life that allows non-active members to form interests in a certain Fraternity. The admin page will allow the admin of a specific fraternity to search, insert, delete and organize their members, alumni, community service events, finances, potential new members and attendance making their everyday jobs on one website. Ideally, since it is very difficult to receive information from every fraternity prior to releasing the website, all entries for data from each fraternity will be entered manually by the fraternity admin.

**Group Members:**

Ryan Poholsky

Matthew Demchuk

Nick Lautieri

Ikomi Moki

Sam Blaxberg

Dr. Jing

COSC 386-001

March 4, 2022

**Requirements:**

The database keeps track of a **FRATERNITY**, storing it’s unique name. Each **FRATERNITY** keeps track of it’s **ACCOUNT** information, **COMMUNITY\_SERVICE** events, and potential new members(**PNM**s).

**ACCOUNT** entity set will store each fraternities *name*, unique *username*, *password*, and *email*. This data will be encrypted through the use of a hash function to keep it secure from possible cyber attacks. This is related to **FRATERNITY** since each fraternity needs one **ACCOUNT** to access and use the database's functions.

**COMMUNITY\_SERVICE** entity set will store the information of community service events that the various fraternities attend and participate in. This is stored with each event’s *name*, *host*, *date*, *numMembers*, and *hours* worked at the event. This is related to **FRATERNITY** through the relation of **ATTEND**.

**ATTEND** is the relation between **FRATERNITY** and **COMMUNITY\_SERVICE**. This relation is used to keep track of which **FRATERNITY** attended which **COMMUNITY\_SERVICE** events by the primary keys *fname*, *csname*, and *date* of the event.

**PNM** also known as potential new members will store all of the new incoming students who plan to rush the various fraternities at Salisbury University. Within this table we will keep track of their *name*, *email*, *phone* number, *GPA*, and which *term* they are rushing in.

**RUSH** is a relation connected between FRATERNITY and PNM which contains the attribute *event* that keeps track of the different event names that each PNM rushed for each fraternity. Rush also contains primary keys fraternity *name* and pnm *email* so that they can be categorized in the table.

**MEMBERS** entity set will store each member of every fraternity in the database. It contains student ID (*sID*), the *name* , *phone number*, *email*, *grade*, *GPA*, *major* and *rank* of every member in each fraternity.

**ALUMNI** entity set will store each alumni’s *name*, *email* and previous *rank* in the fraternity. Once an active member in the fraternity graduates, they will be placed in the alumni table. Alumni will categorize its table through *sID* even when their member doesn't attend the University anymore.

**ATTENDANCE** entity set will store active members *date* of attendance, *event* attended to and a boolean *status* to whether they showed up or not. This is related to **MEMBERS** since each individual will be categorized by member sID to access their name.

**DUES** entity set will store the *current dues* that each member of the fraternity owes, how much each member has *paidOff*, the *term* of which they paid (Spring, 2022) and how much a member owes all together categorized by *owed*. This is related to **MEMBERS** since each member needs to be categorized by member *sID*.

**Entity Sets:**

Fraternity(name);

Account(fname, username, password, email),

foreign key (fname) references Fraternity(name);

Community\_Service(name, host, numMembers, hours, date);

PNM(name, term, phone, email, GPA);

Members(fname, name, rank, grade, GPA, phone, major, sID),

foreign key (fname) references Fraternity(name),

Alumni(name, email, rank, sID),

foreign key (sID) references Members(sID);

Attendance(event, status, date, sID),

foreign key (sID) references Members(sID);

Dues(sID, term, owed, currentDues, paidOff),

foreign key (sID) references Members(sID);

**Relations:**

Attend(fname, csname, date),

foreign key (csname, date) references CommunityService(name, date);

foreign key (fname) references Fraternity(name);

Rush(fname, email, event),

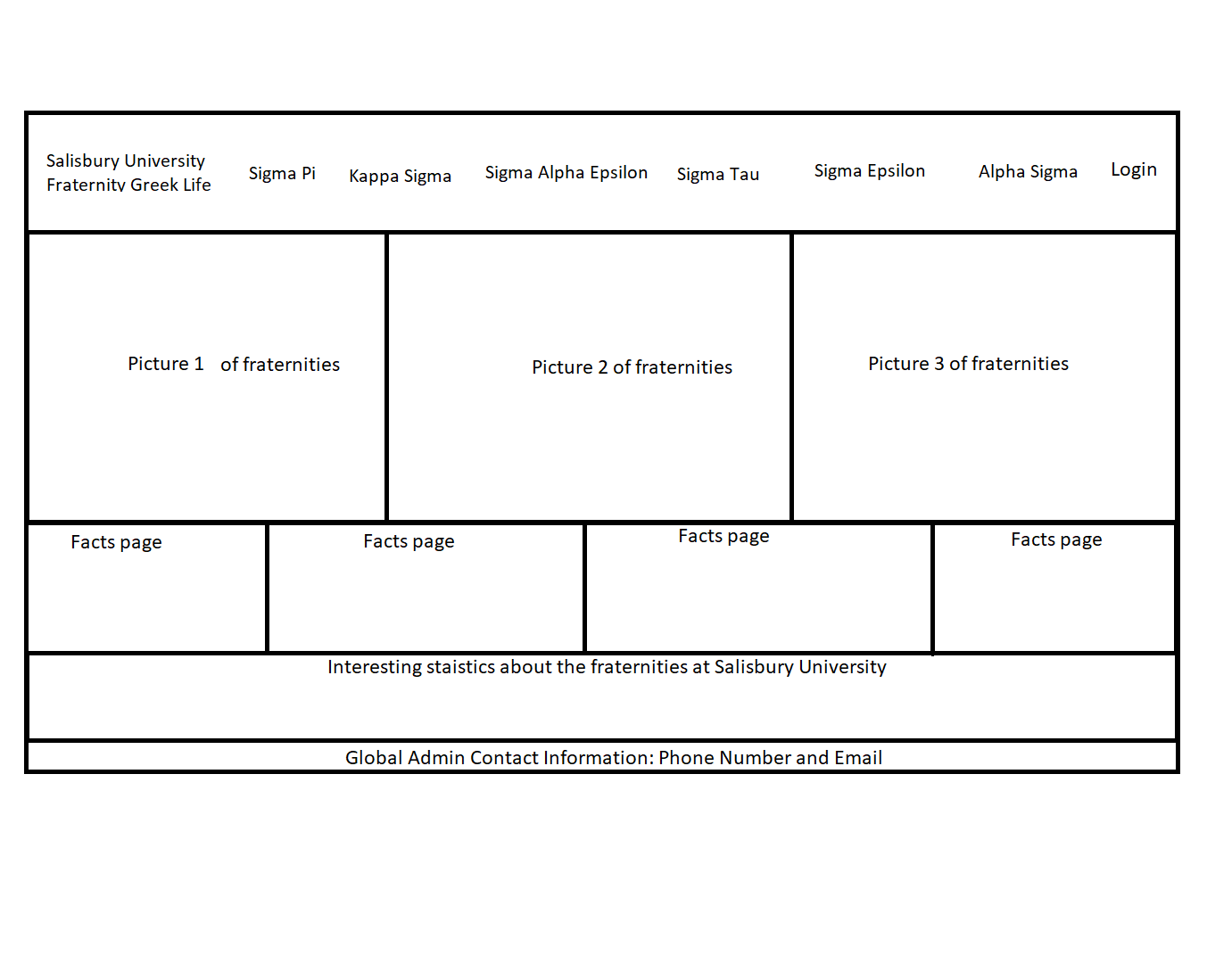
foreign key (email) references PNM(email),

foreign key (fname) references Fraternity(name);

### 

### **Interface Sketch:**

Home page:

Our home page is laid out such that we have five rows of various forms of information, navigation tools, and contact information.

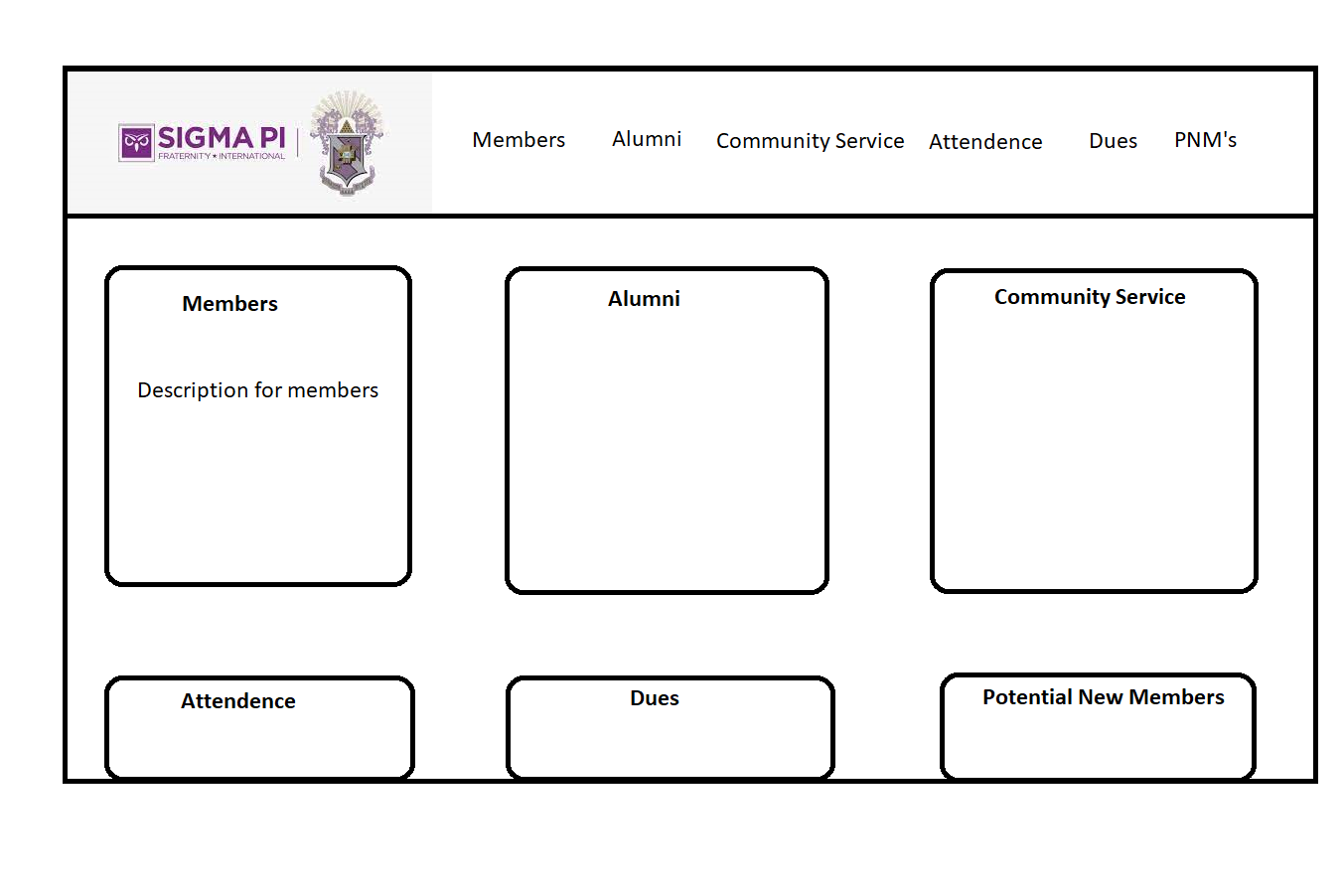
1. Starting from the top row we find our navigation bar which will contain the Salisbury Fraternity Greek Life Logo (tentative based on copyright/fair use compliance) followed by a list of the different fraternities available at Salisbury University, which can be clicked to divert the user/guest to a about us page describing the fraternity. As well at the left end of the bar we have our login button which can be used to access the login page.
2. The second row will be composed of pictures provided by the Fraternities to display as a slideshow to show visitors and users of the website Fraternity activities.
3. The third row will consist of facts about the various Fraternities.
4. The fourth row is composed of interesting statistics about Salisbury University such as member numbers, amount of community services events, etc.
5. The fifth and final row is composed of basic contact information to the global admin for more information or assistance.

Login page:



Our login page will consist of the various logos of the Salisbury Fraternities(currently set to one fraternity for simplicity). The user/fraternity will be able to enter their username and password designated for their fraternity to access their database of information.

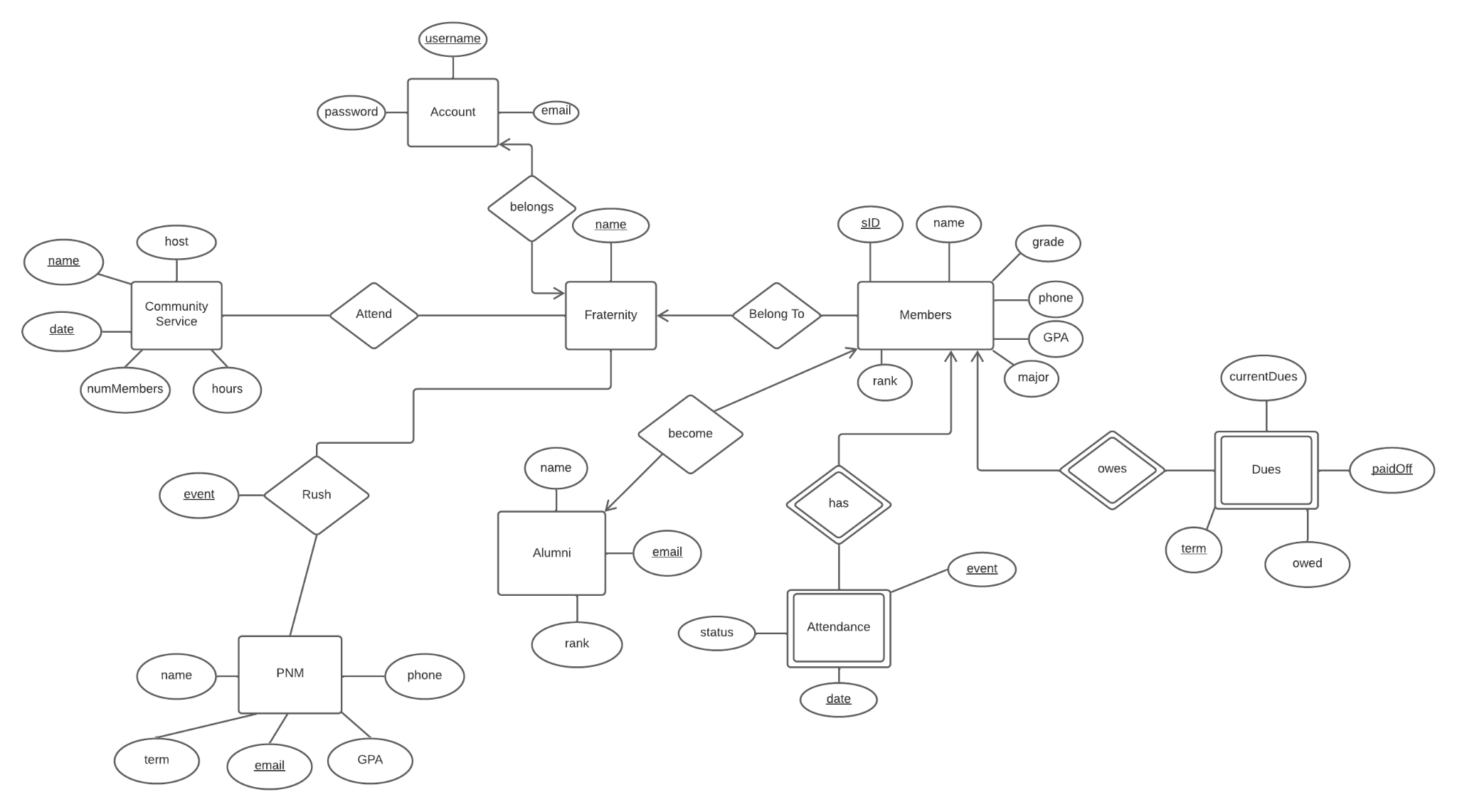
Admin page:



Similar to the homepage the admin page is where the fraternities will be able to keep track of information such as members, alumni, community service events, attendance, dues, and potential new members of the fraternity. Again this is laid out in two sections on the webpage:

1. Do The top row also known as the navigation bar will contain the fraternities logo based on which fraternity has logged into the database. Followed by buttons that will direct the user/admin to the respective page for each piece of information.
2. The bottom row will be composed of six buttons containing the title of where they will take the user/admin followed by a short description of the page. These similar to the navigation bar will direct the user/admin to said page in the database.
3. After selecting an option within this page the bottom section with buttons will be replaced with the respective page requested. Allowing the user/admin to search and modify the information in the database through filters and search bar found in the window.

### **ER Diagram:**



### **Group Work:**

* February 17th

We discussed in class’s lab and later met in the library to finalize our concept for the database. We decided on the idea of a database that tracks fraternities and their members and events, as well as related topics like alumni, dues, and attendance. Several members of the group are in / are interested in fraternities, and working on a project we are passionate about makes both conceptualizing the database as well as working with the actual code a lot more engaging.

* February 23rd

We met in the library to begin to conceptualize how the overall layout of the database would look and perform. Some questions that came up were how the different aspects of the fraternities' information should be translated and incorporated to the database, as well as what information should even be included. After this, we began constructing the schema for the database.

* February 24th

We met after class to finalize our schema for the database. During this time we started to think about how our database’s interface would look to a user. We decided on requiring a login for users, with limited information being available to non-logged-in users and a rundown of fraternity member information being available to users that are logged in.

* March 2nd

After finalizing our schema, we began to convert it to an ER diagram. During this time we worked out how to integrate some new attributes into our database, and updated our schema (and afterwards, our ER diagram) accordingly. We converted our ER diagram (from our sketch) to a digital ER diagram using Lucidchart and began to finalize our proposal document.

* March 3rd

We worked on the proposal document, adding the several required sections and filling out details for our database’s overview, including the outline sketch, entity set information, our translated Lucidchart ER diagram and our group info.

* March 4th

In class, Dr. Jing helped us resolve an issue we were encountering with our structural design. After class, in the library, we finalized some continuity issues with the database, making sure each element of the database was properly categorized and related to its fellow elements, and updated our ER diagram accordingly. We finalized our work on the proposal document, adding a database description and further elaborating on the entity set information and requirement rundown.