Product Design

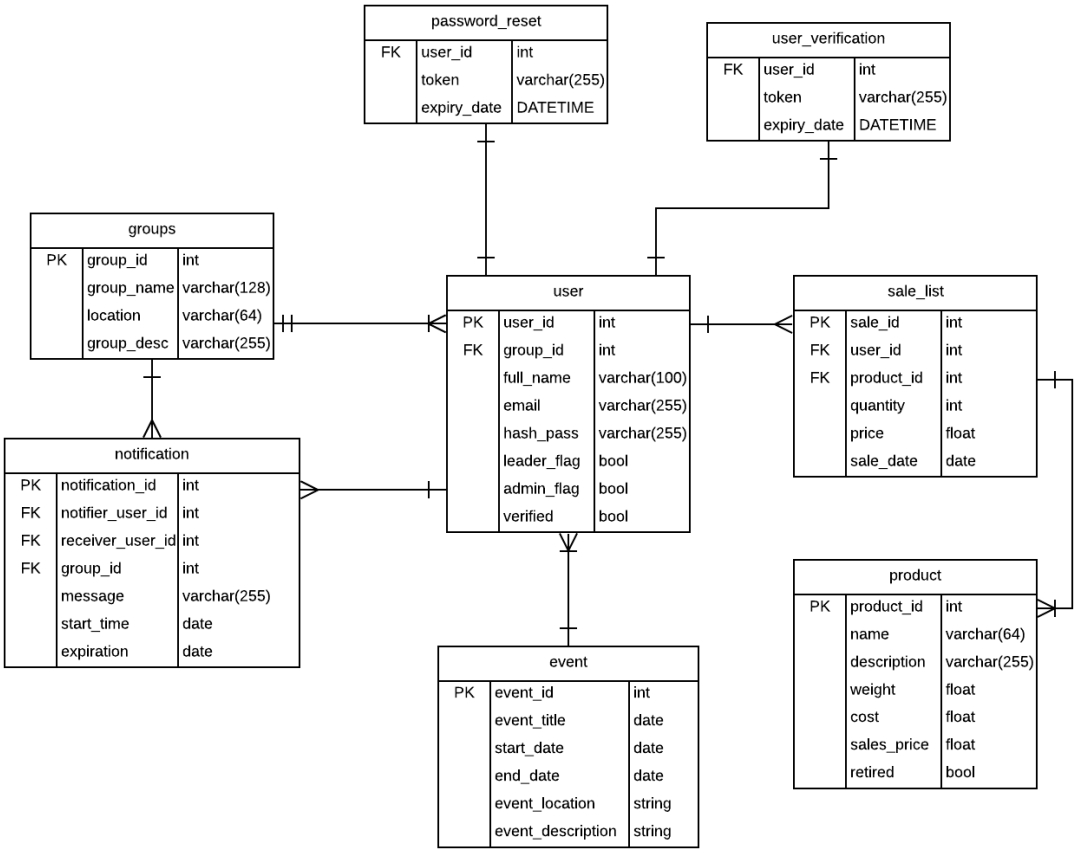
Team 406 Not Acceptable

Mohamed Rahaman, Khang Tran, Jacob Roquemore, Natnael Tsegaselassie, Sam Smetana

Revision Number	***Revision Date***	***Summary of Changes***	***Author(s)***
0.1	02/18/2020	Initial design summary,	Sam Smetana Khang Tran
0.2	02/18/2020	Initial Interface Wireframes	Khang Tran Mohamed Rahaman Natnael Tsegaselassie
0.3	02/19/2020	Initial ER diagram	Sam Smetana Jacob Roquemore
0.4	02/20/2020	Initial Architecture Diagram	Khang Tran Mohamed Rahaman Jacob Roquemore
0.5	02/21/2020	Designed and updated Architecture Diagram	Khang Tran Jacob Roquemore Mohamed Rahaman
1.0	02/22/2020	Completed the summaries, design rationale and UI wireframes.	Mohamed Rahaman Khang Tran Natnael Tsegaselassie Jacob Roquemore Sam Smetana
2.0	03/28/2020	Added reference to info-architecture diagram pdf. Added screenshots for leader dashboard and other pages.	Jacob Roquemore Khang Tran

			Sam Smetana
3.0	04/18/2020	Updated design rationale, updated screenshots, updated ERD	Sam Smetana Jacob Roquemore Khang Tran
4.0	5/2/2020	Update ERD, update database rationale, updated UI rationale, updated UI screenshots	Mohamed Rahaman Khang Tran Natnael Tsegaselassie Jacob Roquemore Sam Smetana

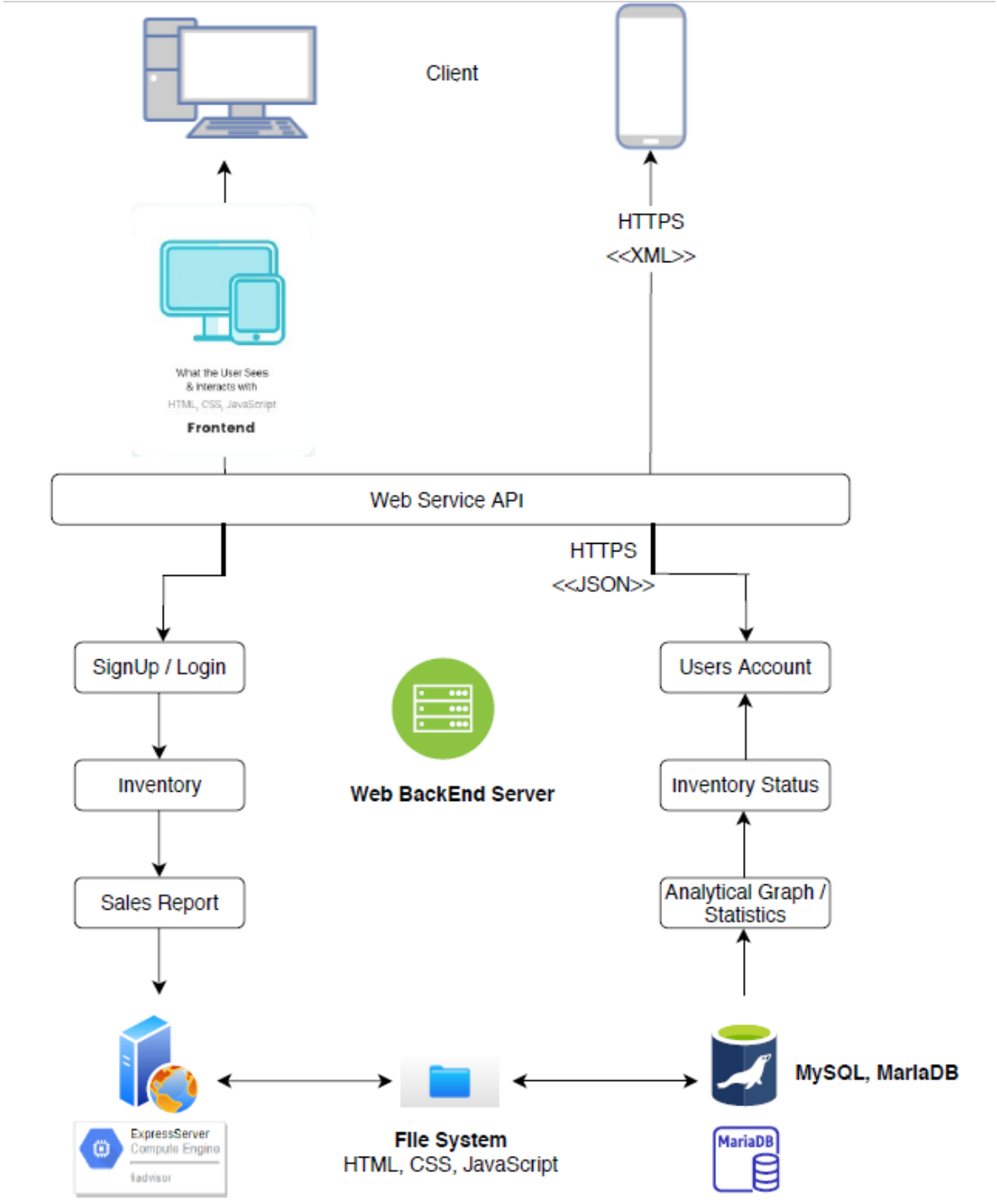
ER Diagram



Information Architecture Diagram

(refer to info_arch_diagram.pdf in canvas submission)

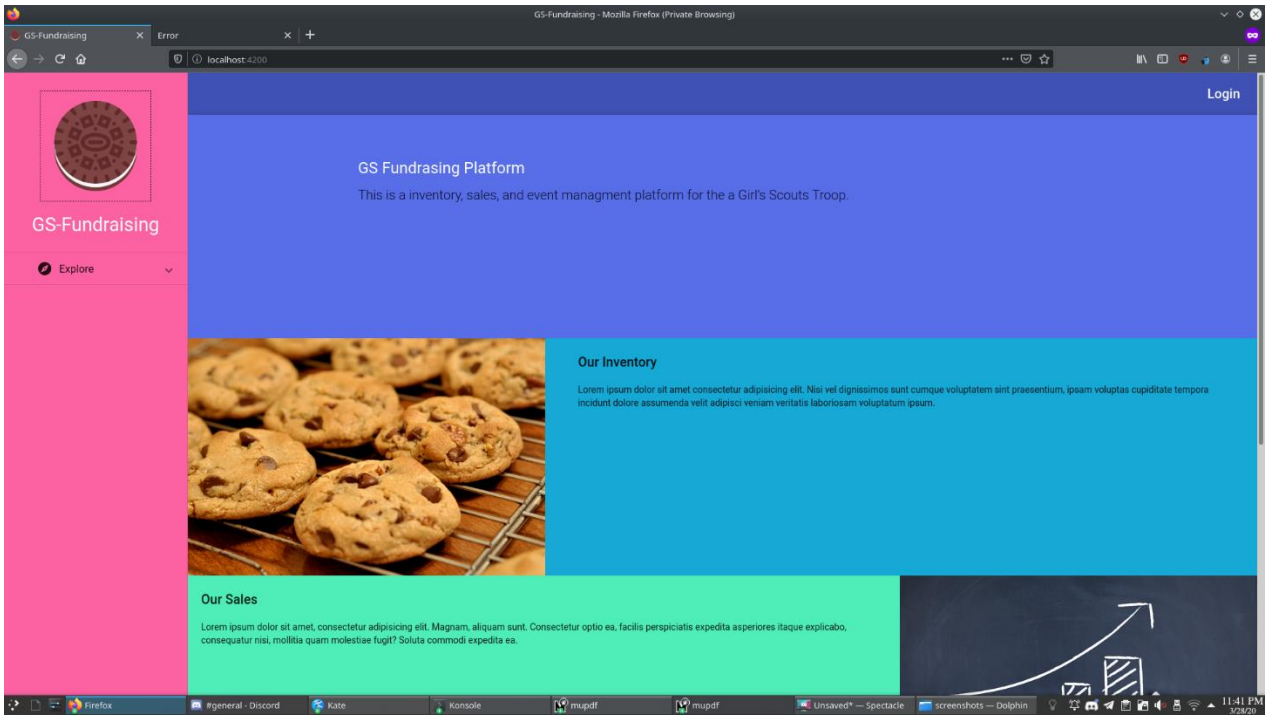
Systems Architecture Diagram

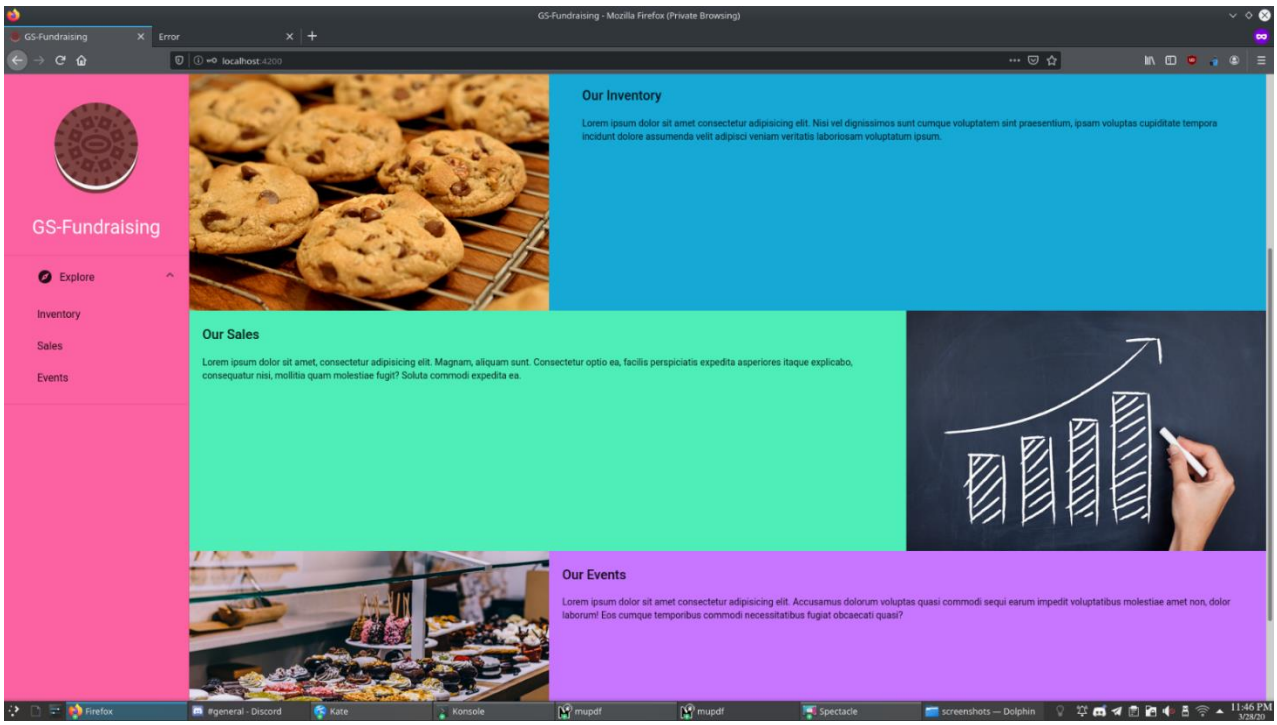


Web Application Architecture

User Interface Wireframe(s)/Screenshot(s)

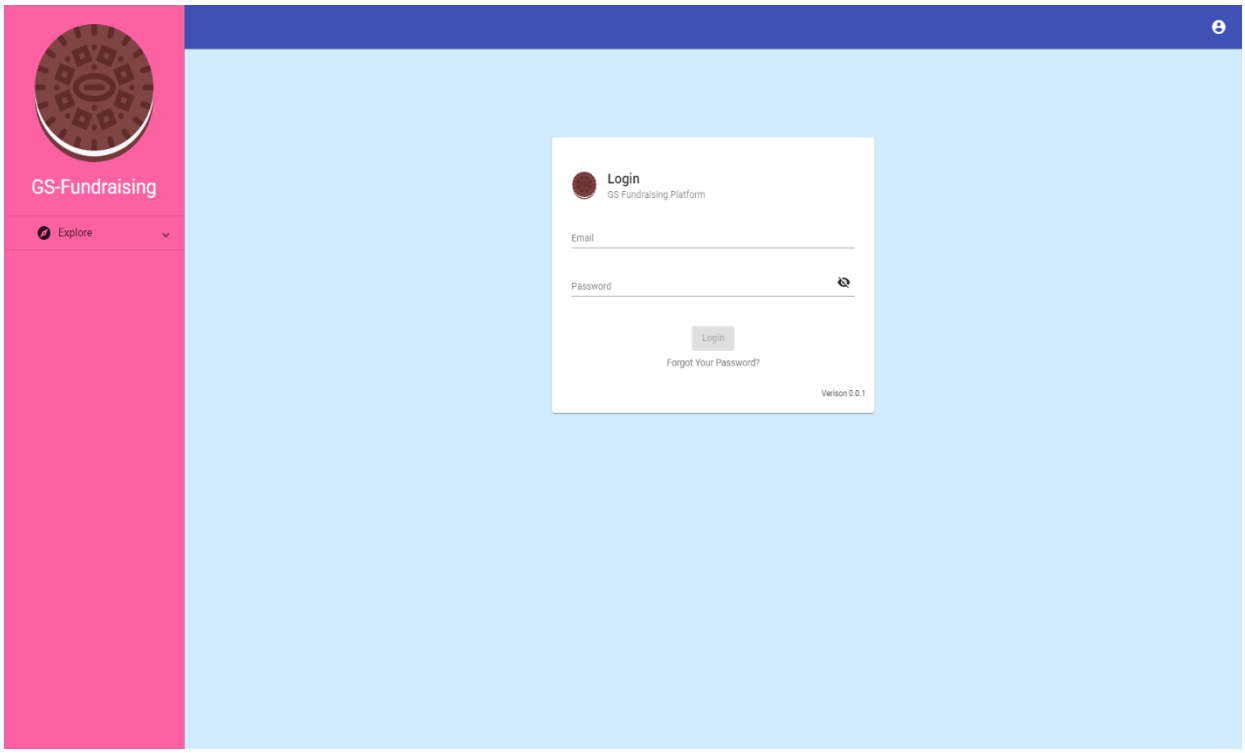
- **Homepage**





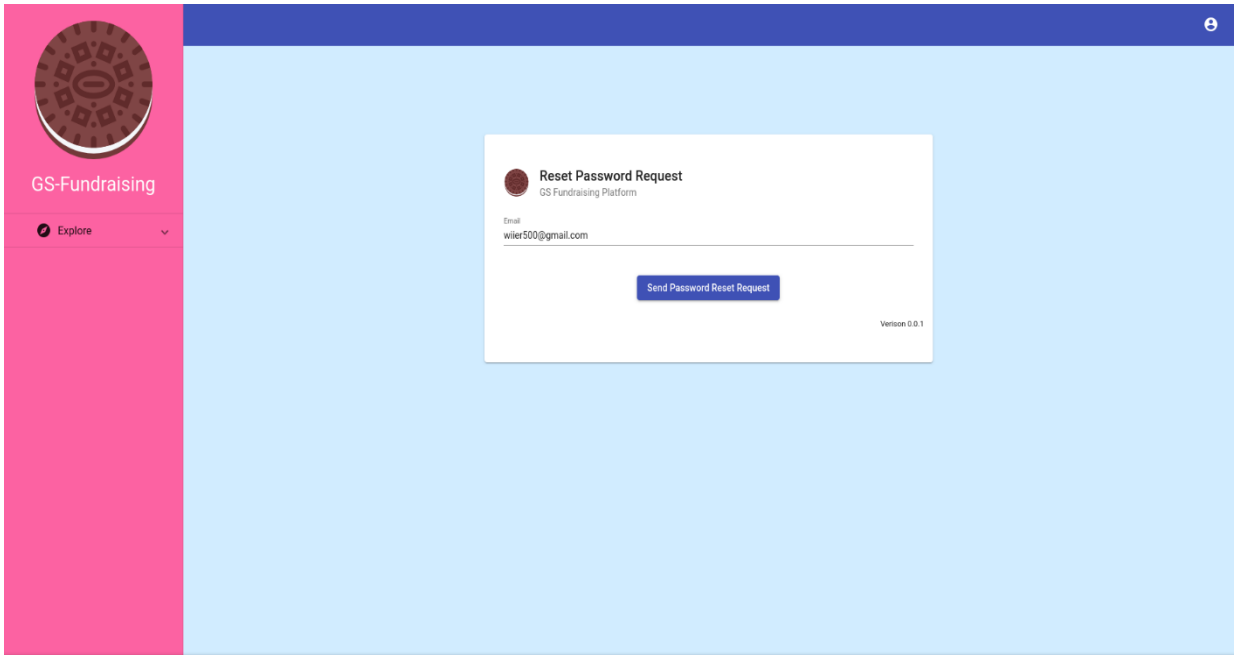
This is the homepage of the web-app. This will be the first page presented when a user enters the URL. Images and other informational data will be inserted later in the open tabs.

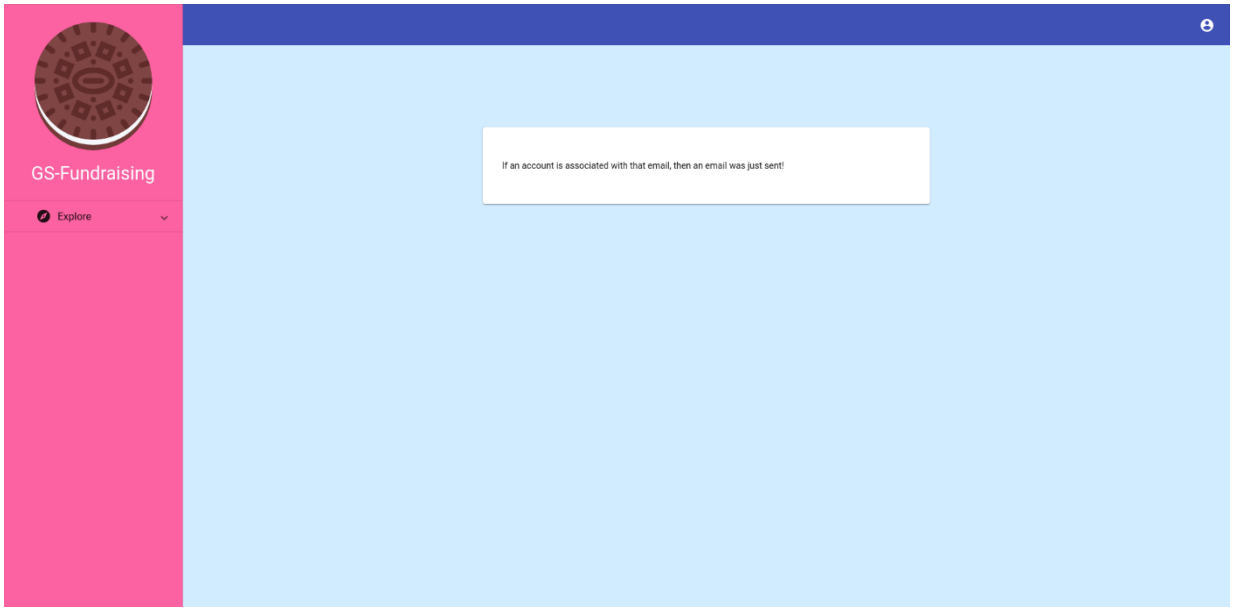
- **Login page**

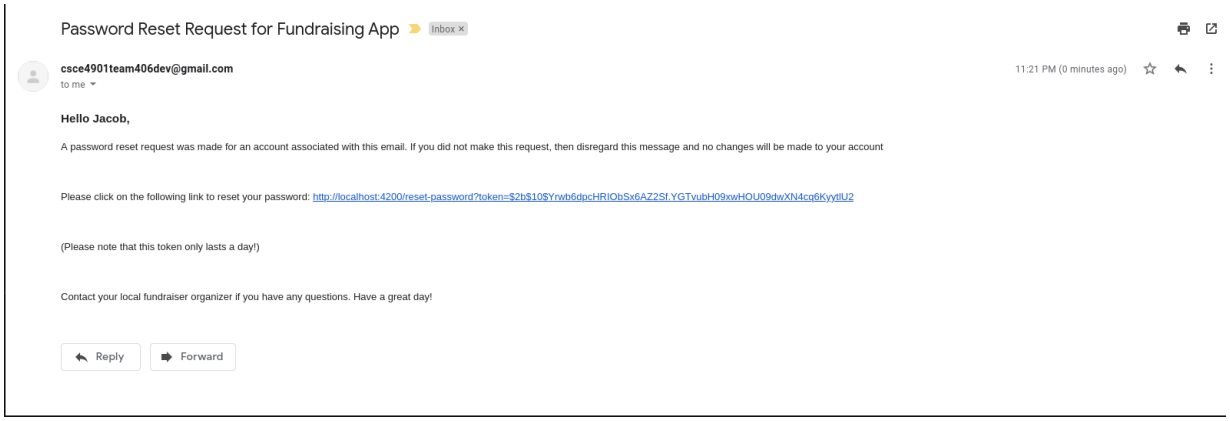


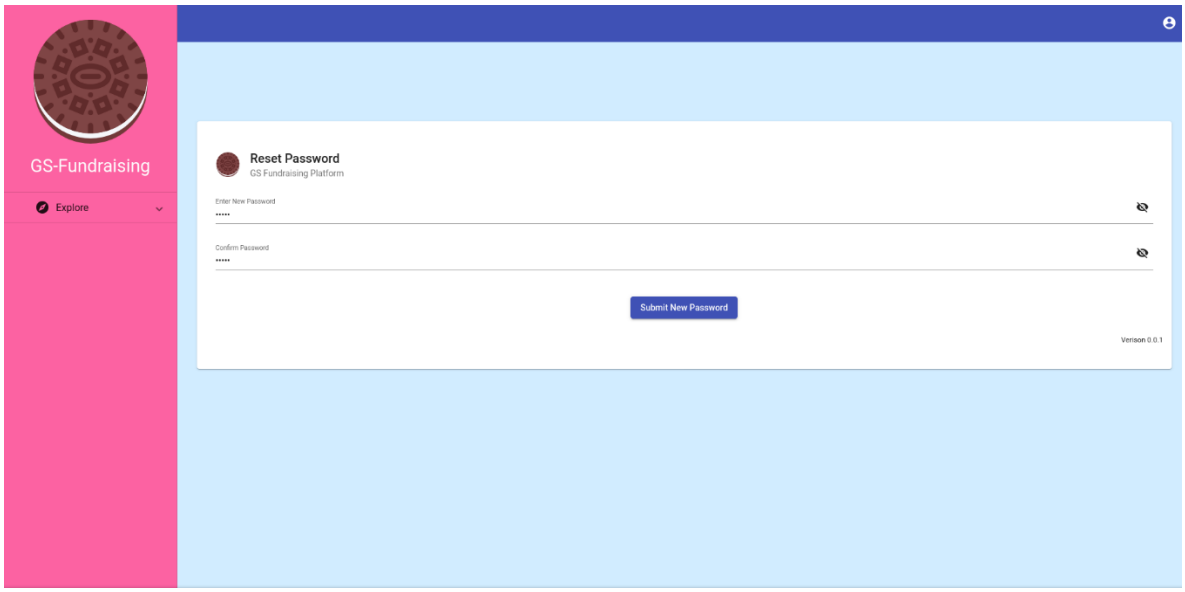
This is the login page of the web-app. The user can use their email and password to login to the system. They also have the option of recovering their password in case they forget it.

- **Password Reset pages**



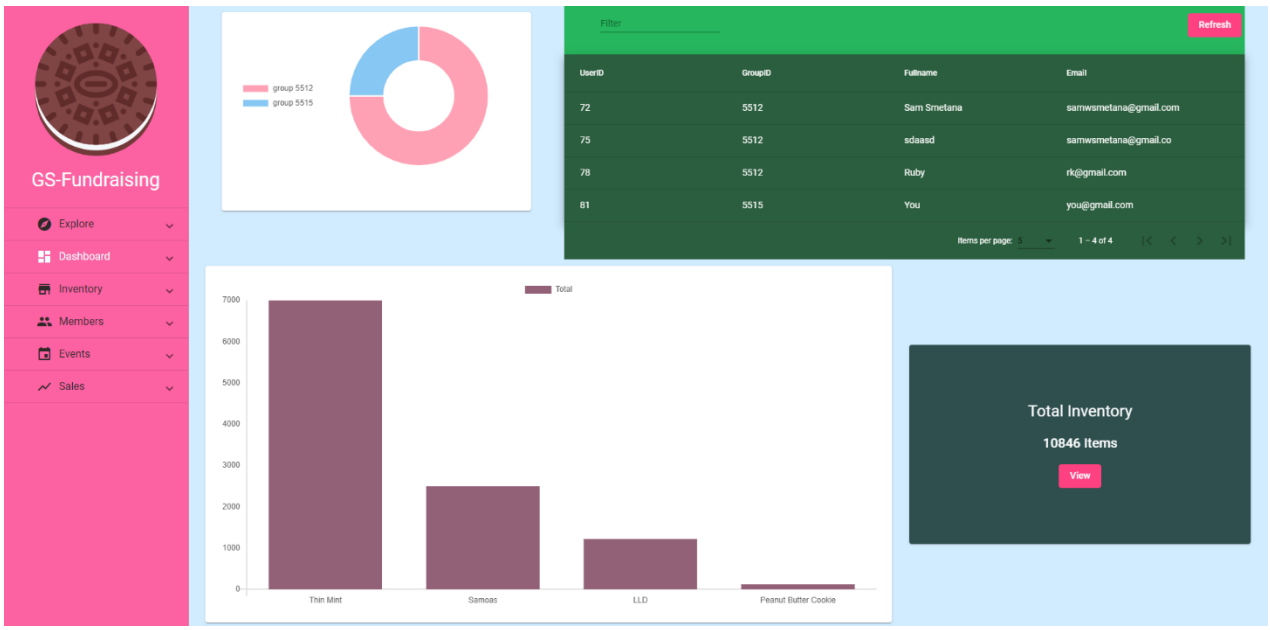

GS-Fundraising
Explore
If an account is associated with that email, then an email was just sent!

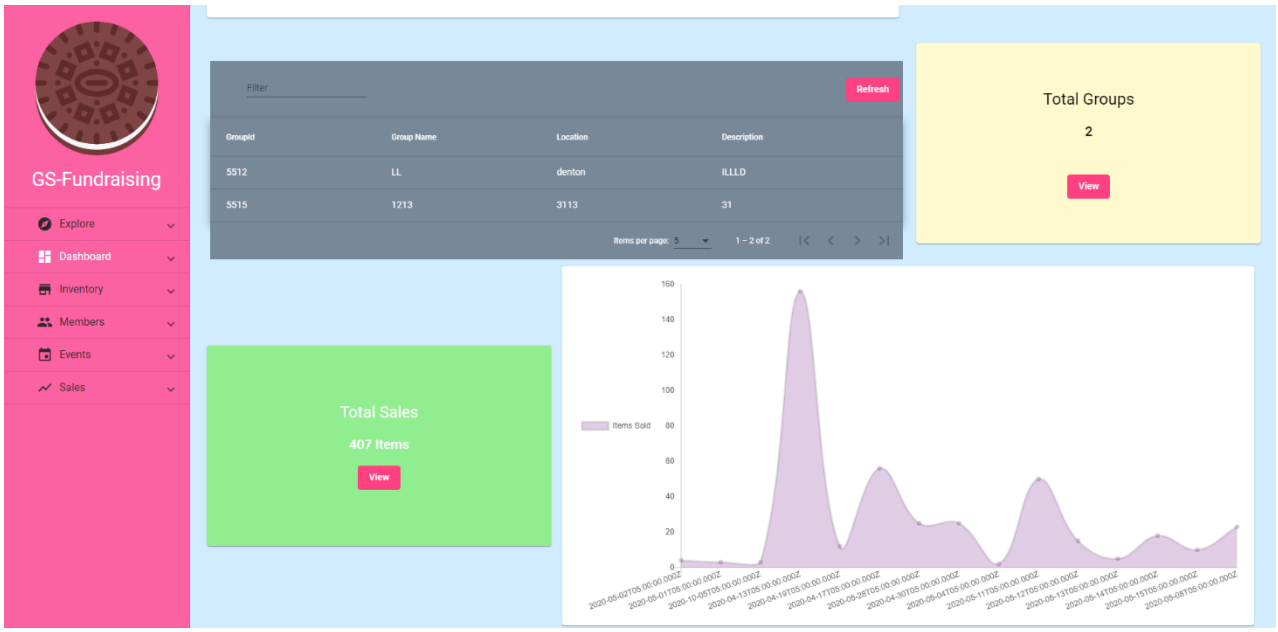

Password Reset Request for Fundraising App
Inbox ×
csce4901team406dev@gmail.com
to me
11:21 PM (0 minutes ago)
Hello Jacob,
A password reset request was made for an account associated with this email. If you did not make this request, then disregard this message and no changes will be made to your account
Please click on the following link to reset your password: http://localhost:4200/reset-password?token=$2b$10$Yrwb6dpcHRIObSx6AZ2Sf.YGTvubH09xwHOU09dwXN4cq6KyytlU2
(Please note that this token only lasts a day!)
Contact your local fundraiser organizer if you have any questions. Have a great day!
Reply
Forward



These screenshots show the process a user will typically take when they first select, “Forgot Your Password?”, on the Login screen, enter their account email, and then open the url in their email to form a new password for their account based on the token they receive.

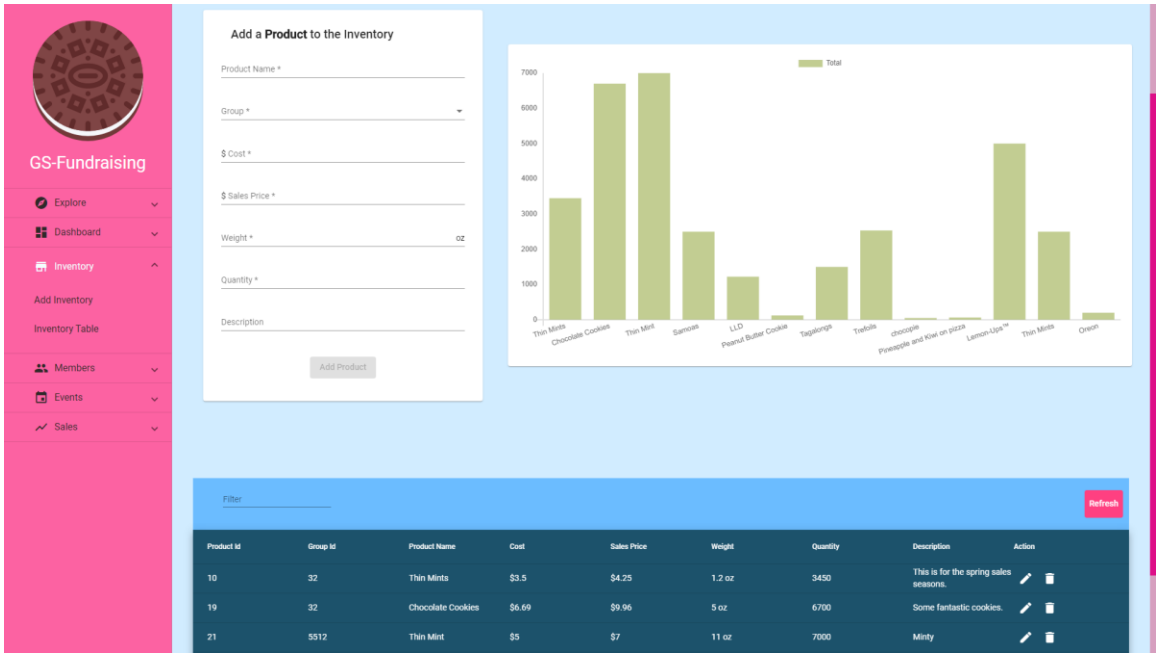
- **Troop Leader dashboard**





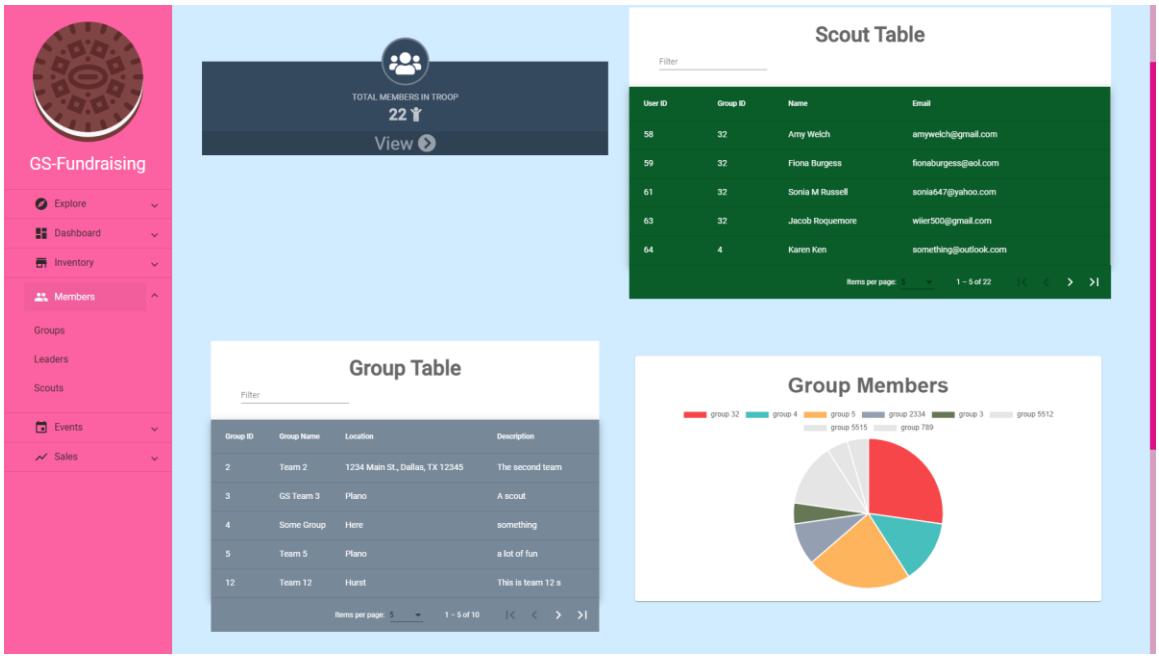
This is the dashboard for the troop leader. In this page, the troop leader has the option to create a group, add scouts to the group, manage inventory, assign certain specific tasks, and track sales made by the scouts of that specific group.

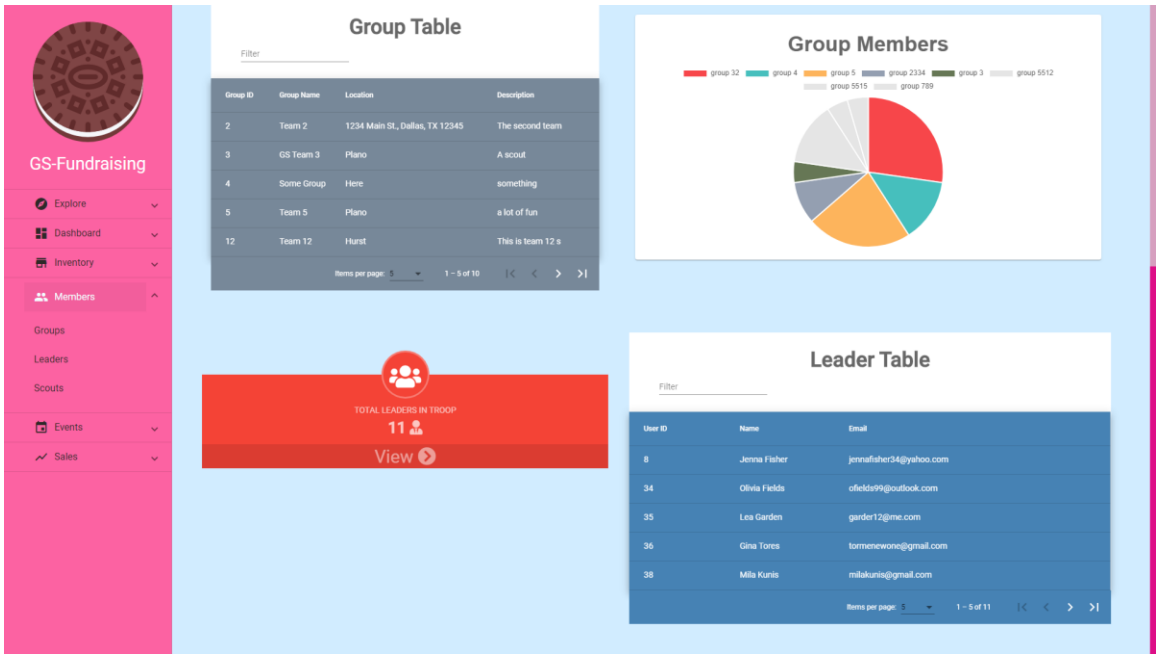
- **Inventory management page**



This is the inventory management page. This is where the troop leader can access data about the inventory of the selected group. It will include information on the quantity, cost, and price of an item. It also has the option of editing that information or adding a new item altogether. There is a bar graph available that provides a visual representation of the amount of inventory left for each item.

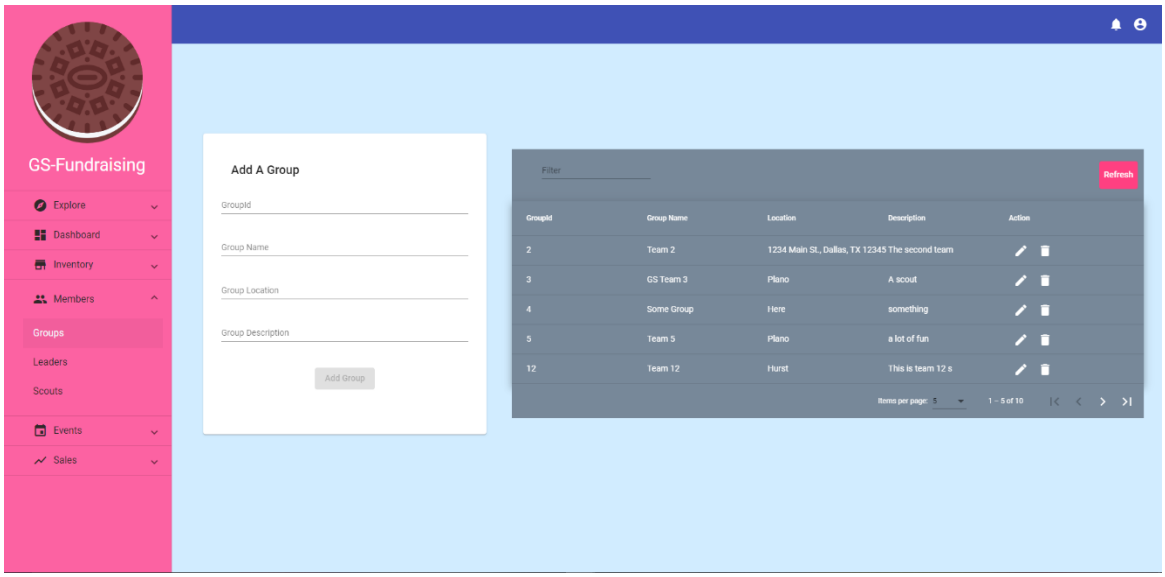
- **Troop members page (on Leader/Admin dashboard)**





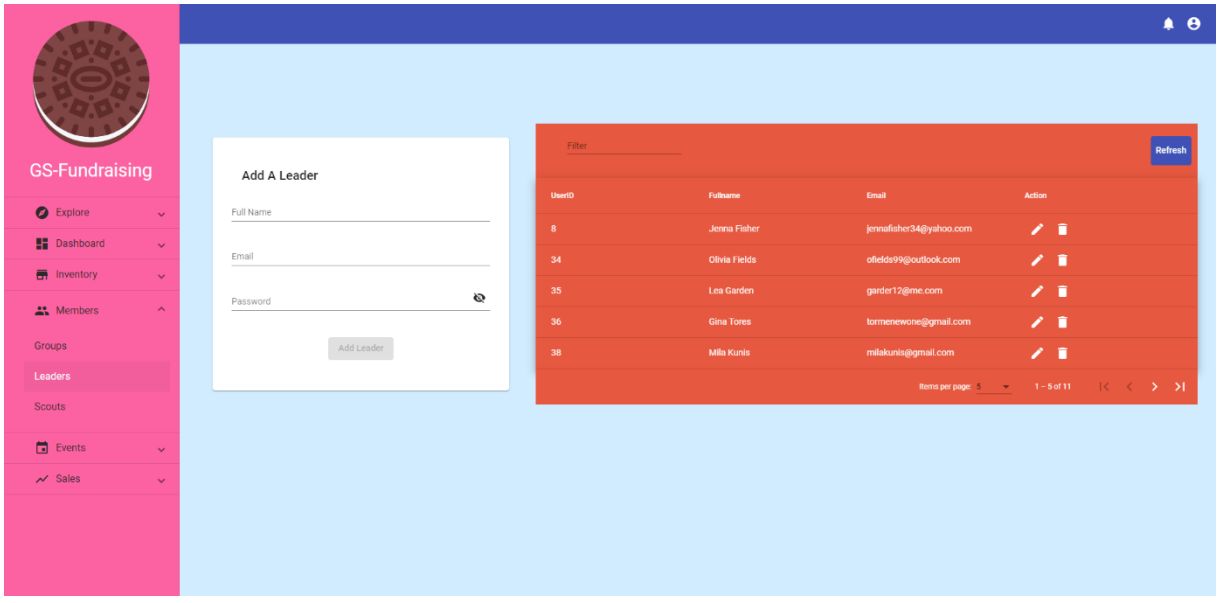
This is the troop members display page. This is where the troop leader views the scouts in their particular groups. It also includes statistical data of the sale made by each scout as well as how much inventory they carry. In addition, this page allows the troop leader to assign inventory to individual scouts.

- **Members[Groups] page**



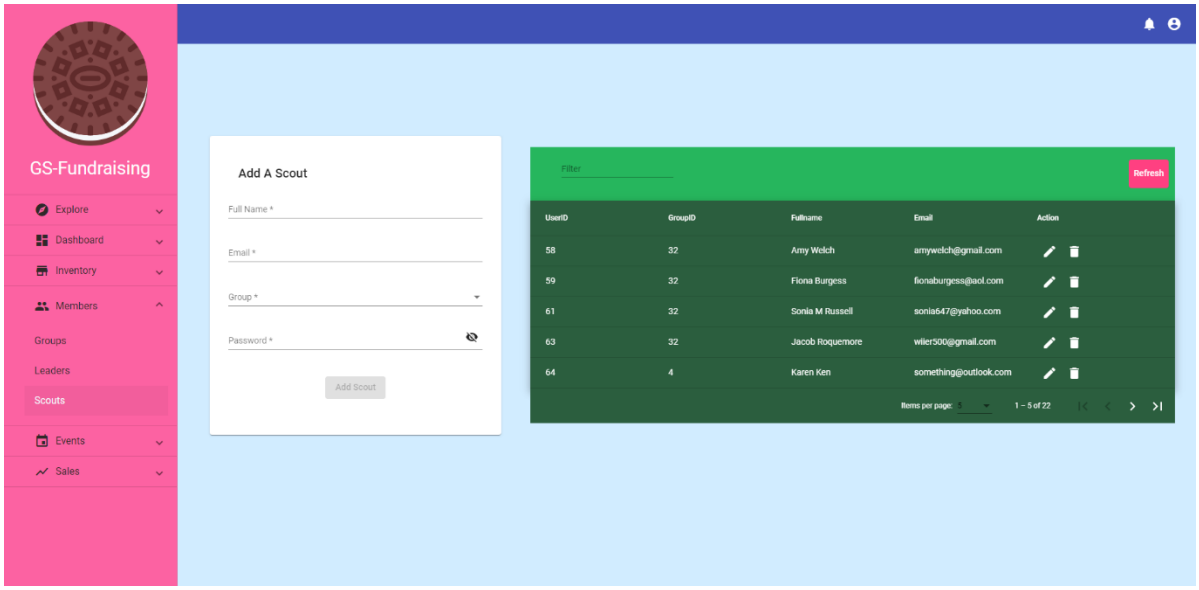
This is the page that allows admin/leader to add a new group into the database and display the data. In addition, the backend also checks the user privilege for authentication purposes. If the user does not have the leader/admin privilege, they will not be able add a new group. The input field also checks whether or not the input is valid.

- **Members [Leaders] page**



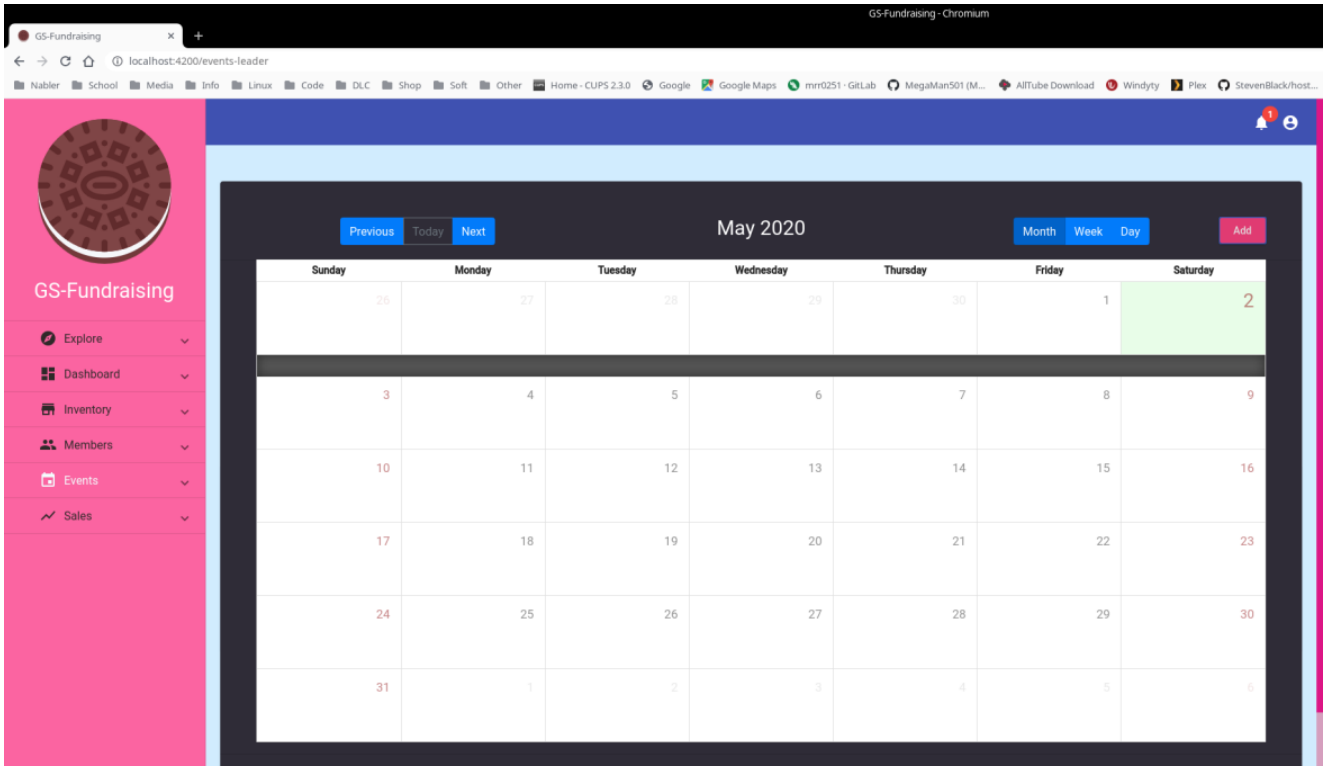
This is a page that allows the super user(admin) to add in a new leader and display who is currently the admin on the list on the right. In addition, it will also check whether the user is authenticated to perform any action on this page. The input field also checks whether or not the input is valid

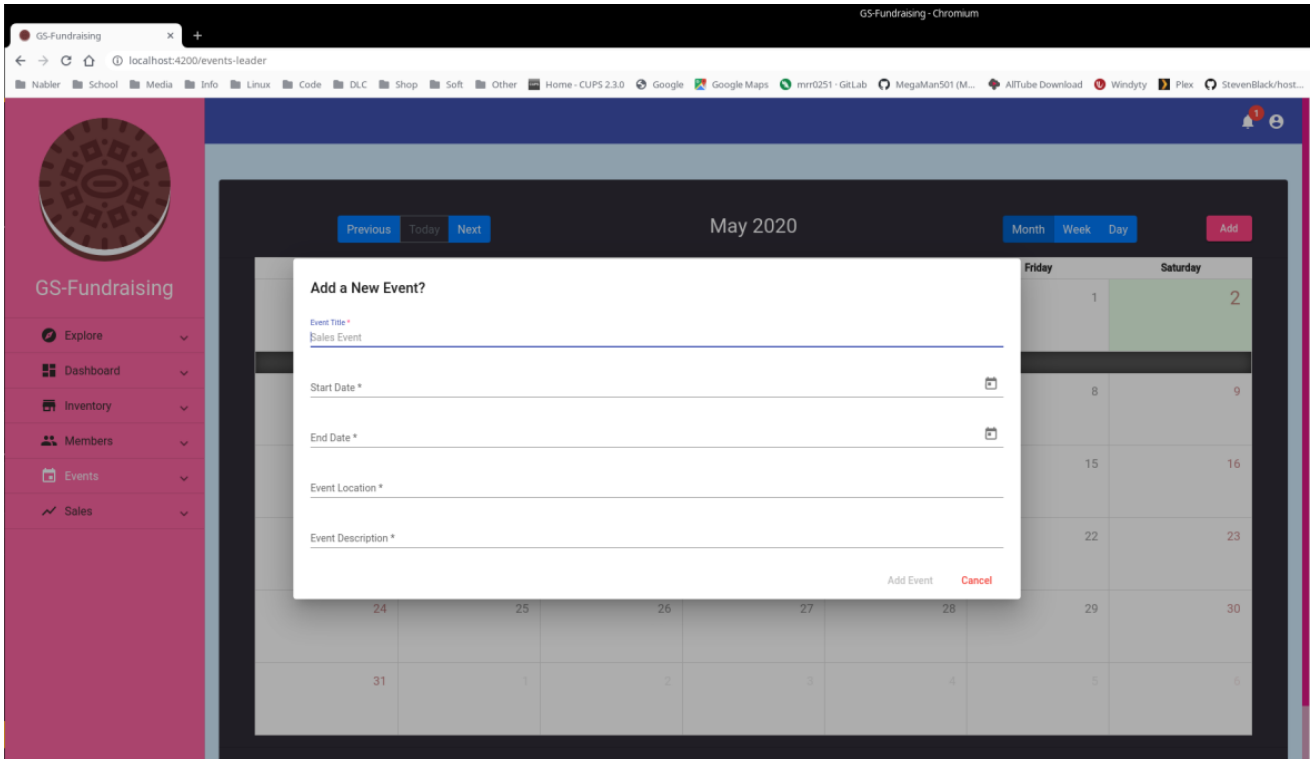
- **Members [Scout] page**

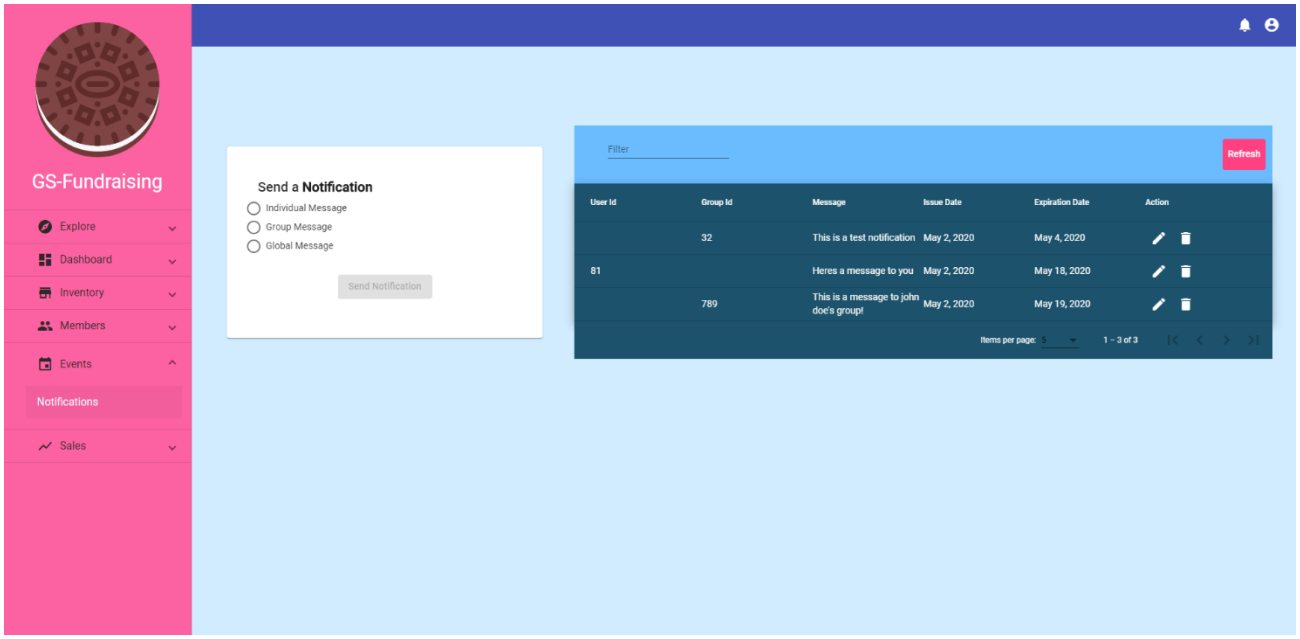


This is a page that allows admin/leader to add a new scout member into the database and display the data. In addition, it will also check whether the user is authenticated to perform the action of adding a scout member on this page. The input field also checks whether or not the input is valid.

- **Event / Notification management page**





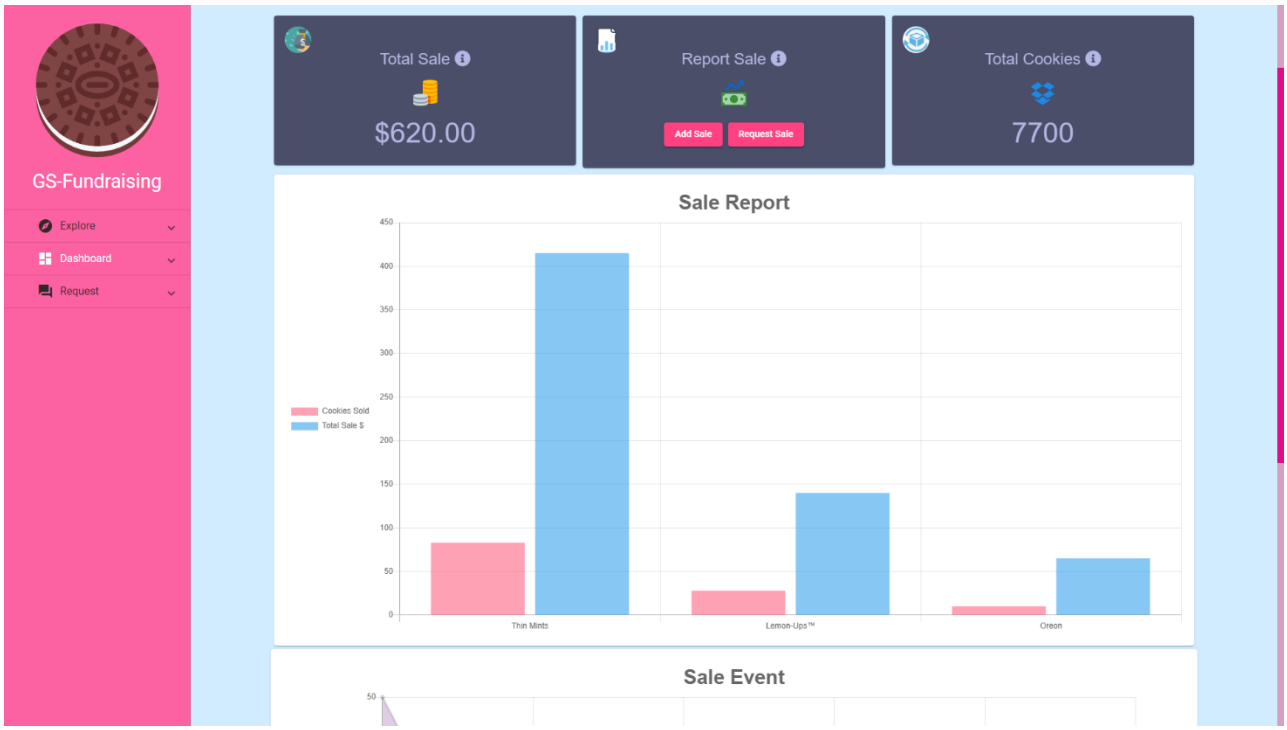


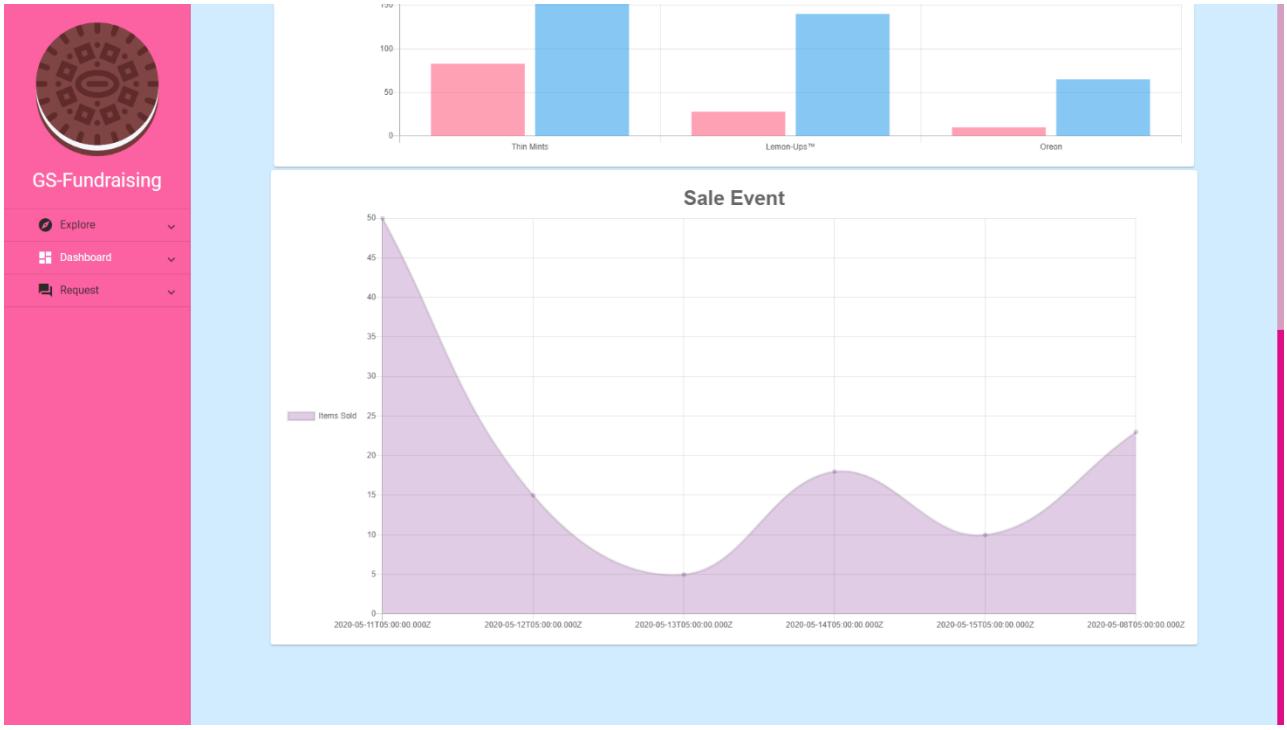
This is the event / task management page. This is where the troop leader creates and manages specific events in the selected group. The troop leader can assign scouts to a certain event and allocate inventory in this page. The descriptions of the events are included in the table.

- **Sales management page**

This is the sales management page, from both the Leader Dashboard and the main Sales page. This is where the troop leader can view the sales and actual revenue made by the troops. In addition, it includes statistical data that shows which items make the most money and are the sold the most.

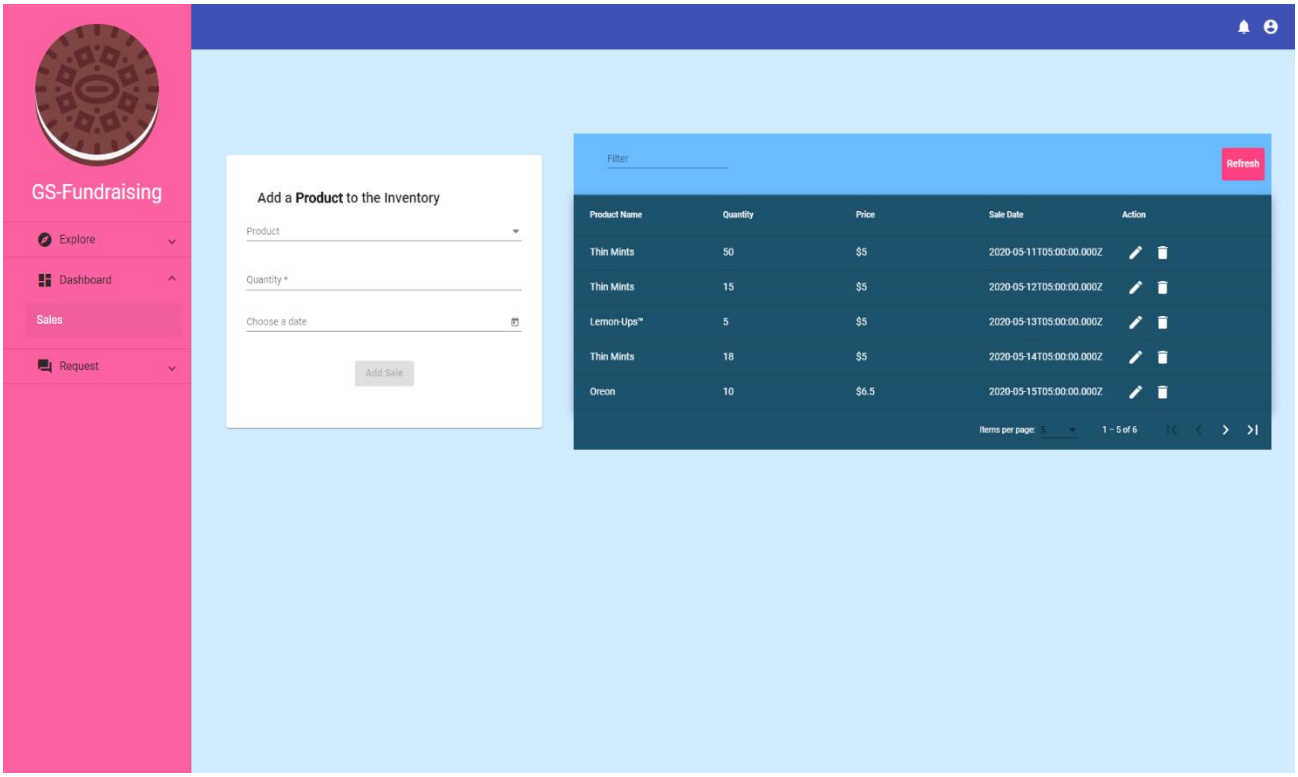
- **Scout Dashboard**





This is the scout dashboard page. This is where the scout can view their upcoming events, the total sales they've made so far, their inventory, and additional sales statistics.

- **Scout Report page**



This is the scout sale report page. This is where the scout can report to the system how many cookies did, they sell and at what date. Essentially, they can also edit their report and delete it.

Design Summary

In order to design the system, we broke our team into smaller groups, one to work on the front-end UI, and one team to work on the database design. We chose to split the team like this because these are the two key aspects of a web-app. The front-end designers focus on the homepage, dashboard and functionalities interface. The back-end designers focus on the quantity and type of data the application requires to design a suitable database. The front-end team incorporates with back-end team to ensure the compatibilities for future integration.

In developing the database, we started by designing basic versions of obvious tables, such as users and groups, which we extracted from the requirements. We then took a deeper look at the requirements to add any more tables that we would need, which were not as obvious. After generating a group of tables which seemed to account for all required functionality, we then began making sure all the tables would function well together. We examined each table and made sure each one made sense and tried to reduce redundant information. Our overall entity relationship diagram's (ERD's) goal was to reduce coupling and increase cohesion among our database tables as much as possible in order to form simpler queries in our code and allow for flexibility. We would only use coupling when necessary, such as through the use of join tables to prevent unnecessary duplicated entries as seen in the "event_attendee", table.

The primary goal in the UI development was to meet all the basic requirements, but another extremely important goal was making the UI easy to use. The web app is intended for users who are not technical; therefore, every aspect needs to be easy to understand. To that end, we tried to design the UI in the most intuitive way possible.

Design Rationale

1. Database Design

Our entity relationship document uses crow's foot notation in order to represent the relationships between any related database tables, and it further uses foreign keys and primary keys to help form proper relationships whenever we will perform JOIN queries in our code. Our sales table was originally going to have unique ids for each sale, but we instead decided to rely on foreign keys from other tables, which is an aspect seen in the industry standard star database schema. Lastly, we decided to use table names such as, "group", and, "product", rather than specific terms such as, "troop", or, "cookie_box", helps to keep the app generic and applicable to other forms of fundraising, and allows groups to have different types of products for fundraising rather than be restricted to a certain type of product.

In the database we have also included a new table called “sales_list”. In the original design, sales did not have an id, and multiple sales could not be grouped together. We added this table so that the sale input pages could more closely mimic the cookie order form which has multiple types of cookies listed in one sale. In addition, we have added tables for

password reset and verification so that passwords can be reset through email. Another change we made, was that we added a retired flag to the inventory list so that whenever cookies are retired, they are not completely removed from the system, so sales listings from the sale table will not display incomplete data.

For sprint 3 we reduced some of the complexity of the database. In sprint 2 we added "sales_list" to the database so that the UI could more closely mirror the sales form, but this complicated the UI too much. We opted to revert the database back to the original sale table because the app is supposed to be for users who may not be very good at using computers. The functionality is still there, but we chose simpler UI with more steps over a complex UI with less steps. In addition, we renamed the tasks table to notification since we got permission to simplify the requirement. The notification table is basically a simpler version of the task table. We also added stored procedures to the database to allow us to put more security in and so that we could perform more complex queries. We greatly reduced the complexity of the database for two reasons: 1) to make the queries easier; 2) reduce the amount of space required for the database.

2. Architecture

We decided upon the client server architecture by crossing off other unsuitable architecture types. First, a group leader needs to ability to check on the status of other users, which means the app needs to be connected to the internet, this crosses off many non-network centered architectures. Among the popular network-centered architectures is peer-to-peer (P2P). We also decided against this style because even though a large enough network may provide faster download speeds, download speed is not as important for an app that uses so little data. In addition, P2P introduces a whole host of other issues such as maintaining connections with multiple hosts. There are other network centered architectures, but they focus on more specific use cases; in summary, the client-server model best suits this application because it requires a database to store data and a server that can service requests from clients and access the database, and it is easy to maintain and scale.

3. UI design

The rationale for the user interface design according to requirements would be to make the UI elements pop out and avoid using a monochromatic color scheme. Moreover, every UI element should be separable, in other words, no two UI elements should cross each other's boundaries. Crossing boundaries both in space and colors would confuse user's attention. Additionally, having the UI interactions clearly label and spaced appropriately would allow for easy navigation between the different pages of the website. Lastly, labeling the top navigation bar with labels for the group ids and other relevant information would allow users not to get confused with which group they are making updates for. This applies to scout leaders because they may manage multiple groups at the same time and having some way of knowing which group is currently selected would reduce any errors in the inventory systems for each of the groups.

Another UI design decision we made to reduce errors is the usage of form selector type inputs rather than text input boxes as a way of mediating input. For example, when

inputting sales, scouts can only use drop down menus and a date picker. The troop leaders do have text input because they have to add new information and it is unavoidable that they need to be able to input raw text.

For sprint 3 we basically just kept adding more input forms that are mediated to prevent users from putting in malformed data. We also made the UI for crud operations on many tables more user friendly than they were last Sprint. There are pop ups to ask a user if they are sure about deleting an item before doing so, to prevent accidently deletes. There are also confirmation messages when editing or deleting items from tables.