EventSquad iOS Mobile App

California Polytechnic State University, San Luis Obispo
CSC / CPE Senior Project
Advisor: Dr. Bruno C. da Silva

By: Rachel Lee, David Lim, Jed Chen

7 June 2018

Abstract

EventSquad is an event planning mobile application designed to help individuals plan for upcoming events. EventSquad allows users to seamlessly create an event and invite other users to help plan for an event. Everyone that is planning for an event should know all the details of the event; EventSquad allows all users planning for an event to know all the details involved with planning for the event so that things could go smoothly. We believe that planning for events should not be stressful and tedious but exciting and enjoyable!

Contents

1 Introduction **1**

1.1 Process Overview . . . 1

1.1.1 Design Work . . . 1

1.1.2 UML Diagram . . . 1

1.1.3 Planning Poker . . . 2

1.1.4 Acceptance Criteria . . . 3

1.2 IDE and Technologies used . . . 3

2 Features and User Stories **3**

2.1 Account Creation and Login . . . 3

2.2 Creating an Event and Event Management . . . 4

2.3 Calendar . . . 4

2.4 To-Do List . . . 4

2.5 Inspiration Board . . . 5

2.6 Budget Tracking . . . 5

3 Architecture **5**

3.1 Google's Firebase . . . 7

3.1.1 Realtime Database . . . 7

3.1.2 Authentication . . . 8

4 UI **9**

4.1 Screenshots . . . 9

5 Future Work **15**

6 Project Retrospective **15**

7 Conclusion **19**

1 Introduction

Planning for events can be a very stressful task. We wanted to develop a mobile application that would make planning for events fun and simple, so we came up with EventSquad! While there are other applications that can help plan for events such as Facebook's event planning application or GroupMe's calender events, we wanted to create something that would allow users to plan for events on the go with ease. EventSquad has several features which include: the ability to create an event and invite others to help plan for an event, a to-do list, a budget tracking list, a calendar with reminders, and an inspiration board. We believe that these are the core features needed to smoothly plan for an event.

Our main goal for creating EventSquad was to learn the process of developing a mobile application and have fun doing so. None of us had experience with developing mobile applications prior to EventSquad. We wanted to learn the process of developing the front and back end of a mobile application.

1.1 Process Overview

Initially, we were going to develop EventSquad on the Android platform using Android Studio. However, after discussion, we opted to switch our development platform to iOS and Swift because we believed it would provide a better learning experience for us as a group. One of us had some experience prior to developing EventSquad which eased us into the process of developing our mobile application.

1.1.1 Design Work

Before diving into code, our group did some design work to simplify EventSquad as much as possible. First, we planned out a very high level timeline for 20 weeks (2 quarters). We divided EventSquad into 4 major releases that were about 1 month apart from each other. Each release had around 8 user stories to be implemented and 2 of the user stories were major features. Having these releases planned out made it much easier to spread out the work and helped keep us on track. For our first release, we focused on account and event management. The second release was focused around getting the to-do list and calendar to function properly. For the third release, we worked on getting the inspiration board and budget to work. For the final release, we worked on getting the invites to users to work while also refining previous features.

1.1.2 UML Diagram

After planning out the timeline, we created a Unified Modeling Language (UML) diagram for EventSquad. The purpose of this diagram is to visually represent a system with its actors, roles, and actions. This helps organize a system that may be complex, which also makes it easier to understand. The UML diagram we created was really useful because it directly correlated to the Swift classes we wrote.

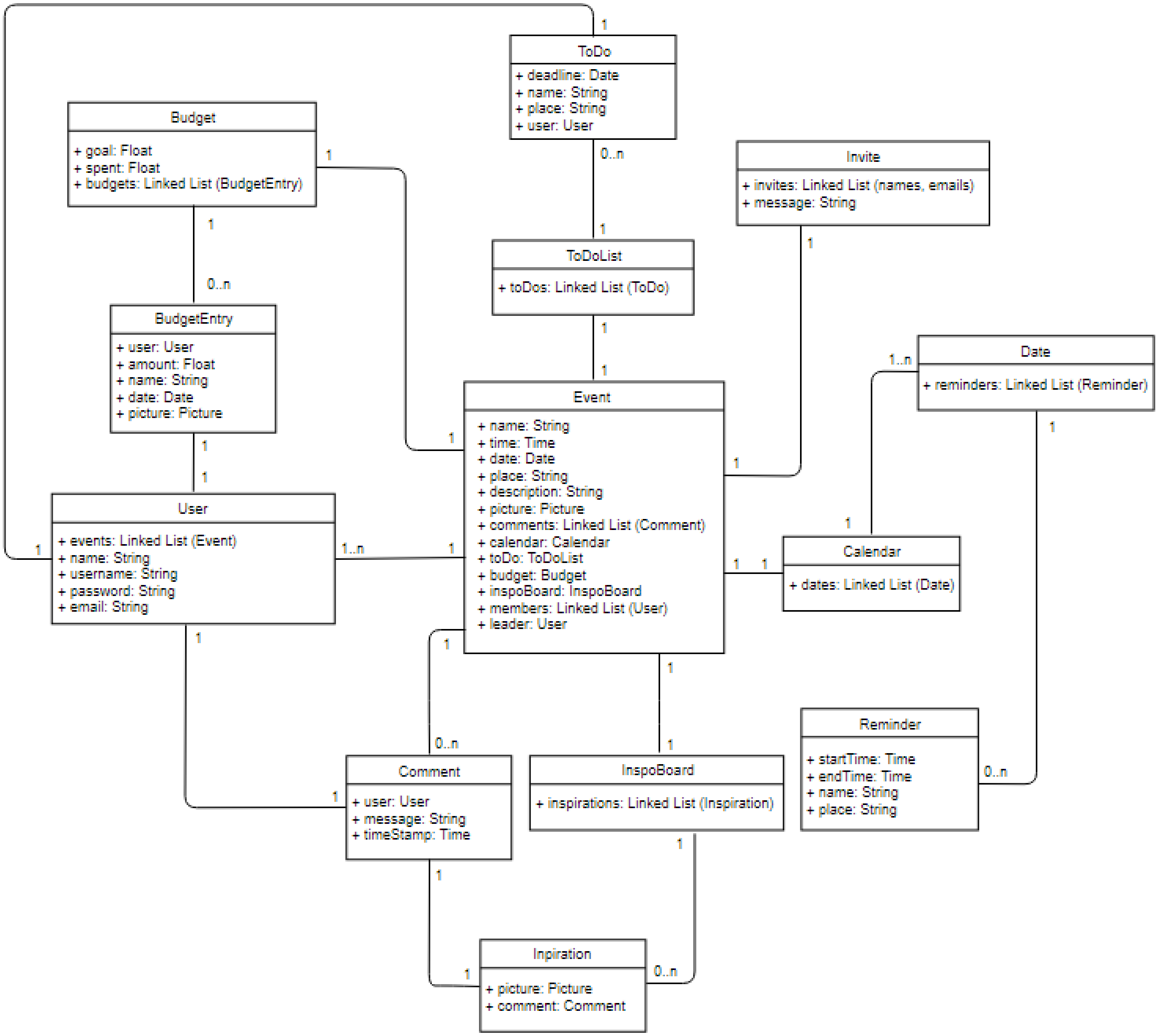


Figure 1: UML diagram for EventSquad

1.1.3 Planning Poker

Planning Poker is a useful tool to help estimate how long it would take to implement an individual use case on a project based on consensus of each team member. After the UML diagram was created, we used Planning Poker to help us estimate the amount of time needed to implement specific user stories.

1.1.4 Acceptance Criteria

Acceptance criteria is defined as a set of conditions that a software product must satisfy to be accepted by a user, customer, or other stakeholder. An acceptance criterion defines the boundaries and parameters of a user story, which determines when a user story is complete. We used acceptance criteria to help us determine if our user stories were working as expected. It also made debugging much easier since we knew the specific test cases that needed to succeed rather than random searching for a bug.

For example, if our user story was: As a user, when I sign into my account, I want to be able to create an event so that I could plan for that event.

Then our acceptance criteria would be:

- Show an "add" or "+" button on the screen to allow for the creation of an event
- Display list of events created by the user previously
- Display detailed event information (name, place, date, description, budget, members list, etc.) when a specific event is selected
- Any user that is invited to the event should have access to the event

1.2 IDE and Technologies used

EventSquad is built on Swift 4.0 and XCode 9.2 on Mac OS Sierra 10.12.6. The front end for EventSquad is built using XCode's Interface Builder. Google's Firebase serves as the backend for EventSquad. CocoaPods is the dependency manager used for EventSquad. Testing for EventSquad has been done with the iPhone simulator offered by XCode.

2 Features and User Stories

EventSquad has three main uses: user accounts, creating and maintaining events, and sending out invitations to others. Within an event, users are able to keep track of things to do, deadlines, budgets, and event details. Many of these features are standalone features for other apps, but the main goal of this app was to bring these various features together. This allows users to keep track of everything through one app rather than through multiple apps. Users can easily navigate the app with the bottom bar to access the different features for the app.

2.1 Account Creation and Login

When a user opens up EventSquad for the very first time, the user will have the ability to create an account which will require their first and last name, and an email address the user is willing to link to their EventSquad account. Once an account is created by the user, the user's information will be stored on the Firebase server and the user will be moved to the home screen. If a user already has an account with EventSquad, they will just need their email address and password to sign into EventSquad. If a user is logged in, the user will also have the ability to logout which will save their event information and bring them back to the log in screen. Refer to Figure 4a and 4b to see what the account creation and login screen looks like. The first time a user logs in after creating an account, they must also verify their email address. They should be sent an email with a link to verify their account and can click resend email if they did not receive one. Once the email is verified it does not need to be done again and the user can login in normally each time afterwards.

2.2 Creating an Event and Event Management

A user has the ability to create an event once they are logged in. When a user creates an event, the user will have to specify some details for the event. Every event has a name, a date, a place, a description, and a budget. Refer to Figure 4c for a UI screenshot of creating an event.

Once an event is created by the user, the event will be stored on our realtime database so that the user can access the event in the future. The user that created the event will also have the ability to edit and delete the event. Any edits will be pushed onto the database and overwrite any previous information stored in the same fields for the event.

2.3 Calendar

The calendar allows for users to add reminders for themselves about the various events they are involved in. This could be for group meetings or other deadlines that have to be met when planning an event. To add a reminder to the calendar, the user taps the "+" sign on the right side of the top bar. This leads to a new page where the user inputs a name, place, description, and a date. Once this is saved, the information is stored in the database and a red dot will be marked on the calendar for the new event. Clicking on the days that have the red dots will show all the reminders made for that specific date. An area of further improvement in the future would be notifying the user directly on their phone so they can be reminded without having to pull up the calendar itself.

2.4 To-Do List

Users that are involved in planning for an event will have access to a to-do list for that specific event. The to-do list is used to keep track of all the smaller tasks that need to be completed to make sure the event is a success and runs smoothly. All users can add items to the to-do list by taping the "+" sign on the right side of the top bar. Users can then add a new to-do item by inputting a name, description, and date. Once a to-do item is on the list, if a user completes that task, they can click on the circle to the left of the to-do item. A green check mark will appear in the circle which tells others group members that the task has been completed.

2.5 Inspiration Board

The inspiration board allows users to add ideas to a board so that they can brainstorm ideas for the event. An inspiration post consists of a picture and a small description. When a user wants to post an inspiration, the user will have to upload an image from their iPhone photo library and write a small description before posting it to the board. Once it is posted, all members for that event should be able to see it. Other users can then hit the heart button underneath the photo to show that they liked this idea.

2.6 Budget Tracking

Through the budget tab, EventSquad allows users to track the amount of money spent so far for an event. The budget tab has a list of all the expenses for an event. To add an expense, the user can press the "+" button at the top right hand corner. Each expense has the name of the person who bought the item(s), a short description and how much was spent. Once all the expenses are listed, the members can split the bill using the "Split the Bill" button. When pressed it will divided the total bill evenly among all the members involved with that event.

3 Architecture

Originally, our app was going to be an Android app because all of us had coded in Java and it seemed like the easier option. We had a more complicated architecture having a frontend, backend and multiple databases. We were going to code the backend in Java and the frontend using XML which is used specifically for Android apps. We were also going to use SQL Lite as a local database and the Google Firebase as our server side database. Initially, our architecture diagram can be seen in Figure 2.

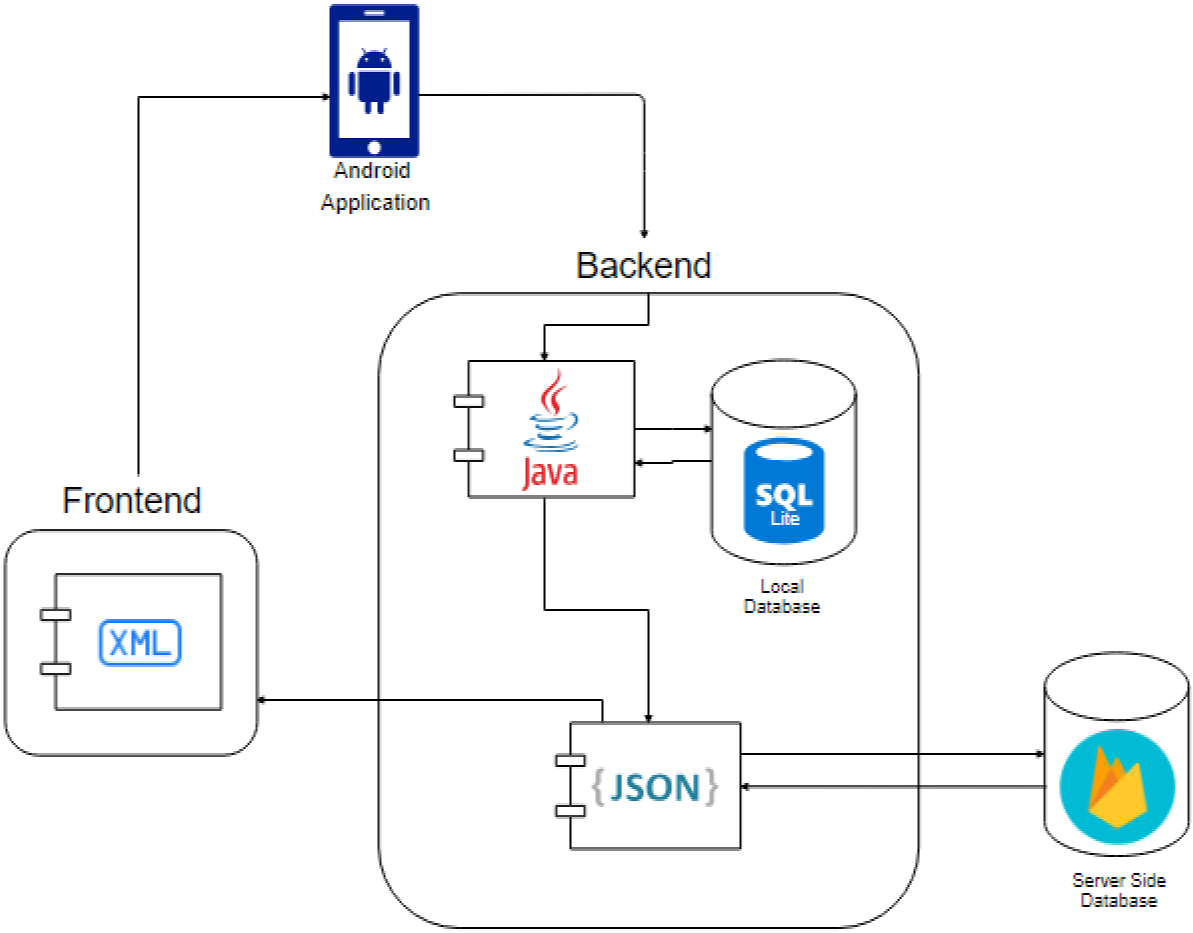


Figure 2: Android architecture diagram for EventSquad

After doing more research and considering other options we decided to switch to creating the app as an iOS app since we had one group member that was learning iOS development at the same time. This would make the learning process much smoother and we would have a better base to build off of rather than just trying to create an app without any experience. This also made our architecture much more simple. We decided against using a local database because we felt it was not necessary for our app and would not worry about any cases where the user did not have internet access. Also, we no longer had to breakdown frontend and backend since we could do both in Swift with various controllers and views. To make things simpler, we decided to use a Model View Controller architecture. Input would be read from the phone and processed. Then the information would be taken from the Google Firebase or uploaded/updated as necessary and then sent back to update the views. After switching our development platform to iOS, we came up with the architecture diagram seen in Figure 3.

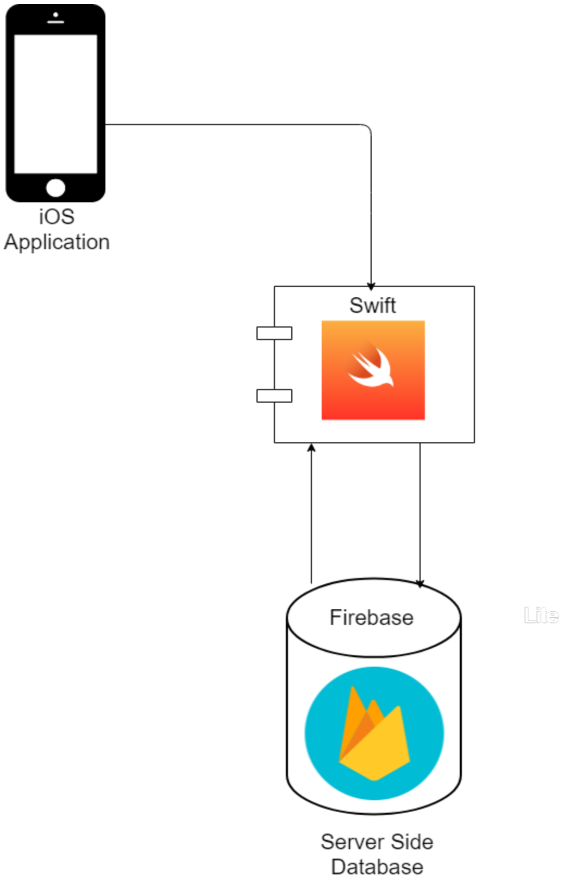


Figure 3: iOS architecture diagram for EventSquad

3.1 Google's Firebase

Google's Firebase is a platform that allows users to build mobile and web applications without a server side programming language. Firebase offers a realtime database which stores user data as JSON and allows user data to be synced in realtime to all connected clients. In addition to a realtime database, some other useful services that Firebase provides include: email invites, user authentication, and analytics.

We decided to go with Google's Firebase because it was free, easy to setup, and met all of our needs. Firebase was a learning experience for all of us. Although all of us had dealt with relational databases, none of us had any experience with realtime databases.

3.1.1 Realtime Database

Firebase's realtime database provided a means of storage for user and event information. Because Firebase stores data as JSON and synchronizes the data in realtime to all clients, it was easier for us to debug some of our problems when the backend was not working as intended. Firebase's realtime database is really simple, easy to understand, and extremely powerful.

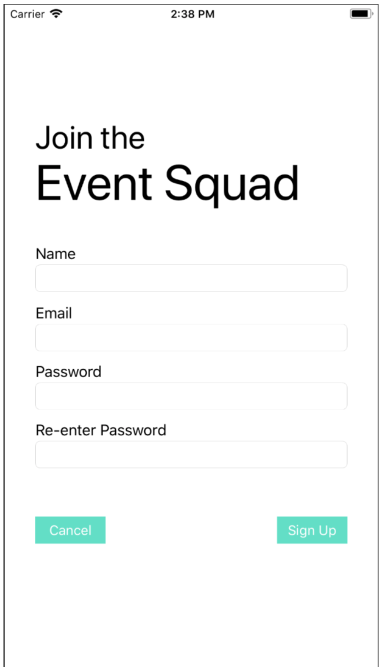
3.1.2 Authentication

Firebase Authentication was really useful for us as it provided a means of user authentication for our mobile application. Firebase Authentication supports different types of authentication using: passwords, phone numbers, and more. For our mobile application we used emails to authenticate our users. Even before using our app, users were required to authenticate their emails with an email verification that was sent to them after signing up to our app.

4 UI

We decided to create a fairly simple design for our UI. We wanted it to be clean and not over cluttered with information, pictures or designs to make it easier to use. We tried to make things intuitive like a "+" to add new items and appropriately labeled any buttons that the user had to click. A nice feature for the UI is that the list of events slides out so the user can access it from any screen. The UI layout has 5 tabs at the bottom that control almost all the pages the user can view. This includes the homepage which brings the user to the most recent event they clicked on and shows all the details for that specific event. The other 4 tabs lead to the various features our app provides which are the calendar, to-do list, inspiration board and budget. This allows for easy access to any of these features no matter what page you our currently viewing.

4.1 Screenshots



Carrier 2:38 PM

Join the
Event Squad

Name

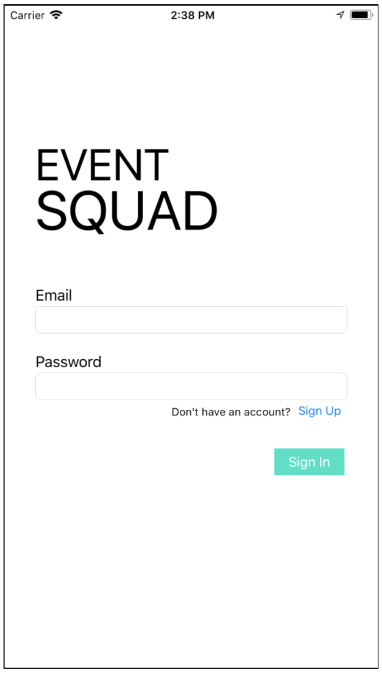
Email

Password

Re-enter Password

Cancel Sign Up

(a) CREATING A NEW ACCOUNT



Carrier 2:38 PM

EVENT
SQUAD

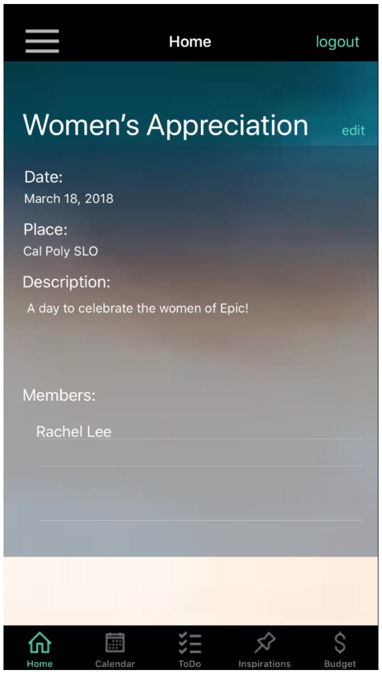
Email

Password

Don't have an account? Sign Up

Sign In

(b) LOG IN SCREEN



Women’s Appreciation

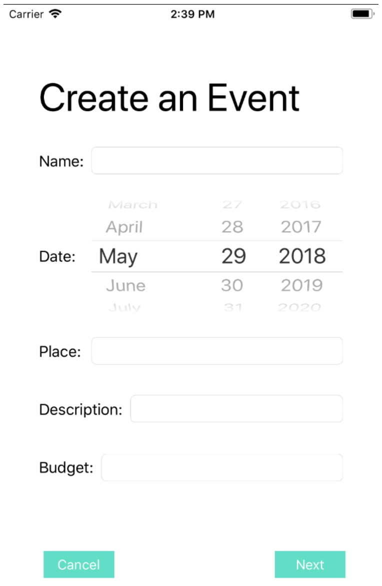
Date:
March 18, 2018

Place:
Cal Poly SLO

Description:
A day to celebrate the women of Epic!

Members:
Rachel Lee

(c) HOME PAGE WITH ONE EVENT SELECTED



Create an Event

Name:

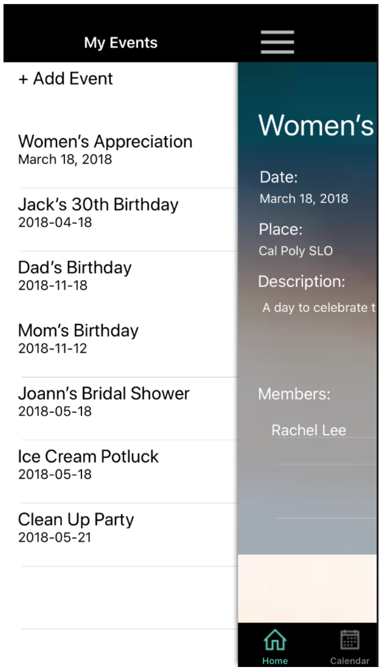
Date: May 29 2018

Place:

Description:

Budget:

(d) CREATING A NEW EVENT



Women’s Appreciation
March 18, 2018

Jack’s 30th Birthday
2018-04-18

Dad’s Birthday
2018-11-18

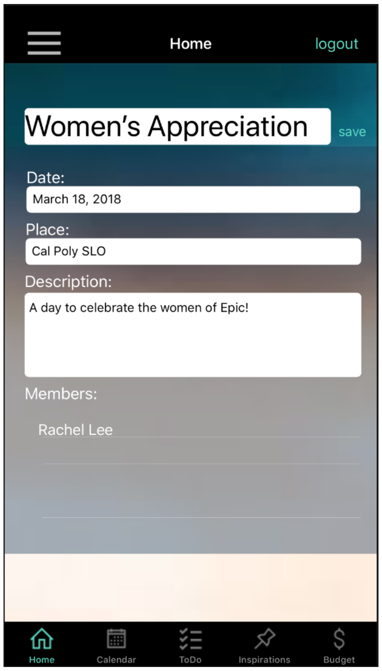
Mom’s Birthday
2018-11-12

Joann’s Bridal Shower
2018-05-18

Ice Cream Potluck
2018-05-18

Clean Up Party
2018-05-21

(e) LIST OF EVENTS



Women’s Appreciation

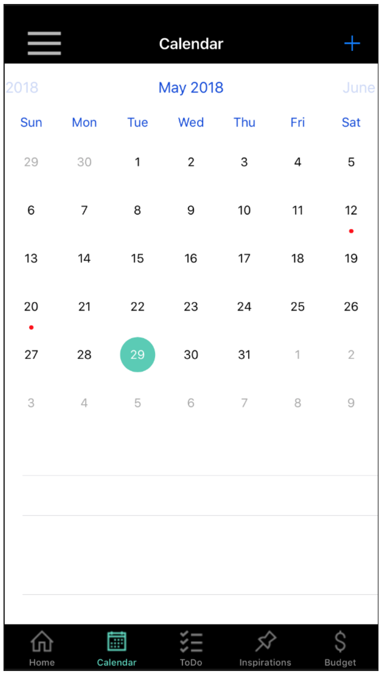
Date:
March 18, 2018

Place:
Cal Poly SLO

Description:
A day to celebrate the women of Epic!

Members:
Rachel Lee

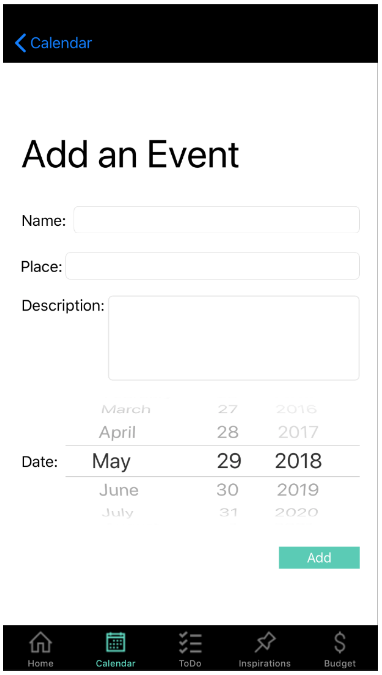
(f) EDITING AN EVENT



May 2018

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

(g) CALENDAR



Add an Event

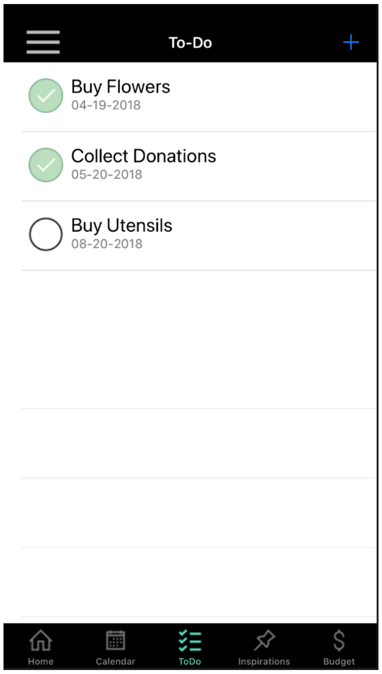
Name:

Place:

Description:

Date: May 29 2018

(h) ADDING AN EVENT TO CALENDAR

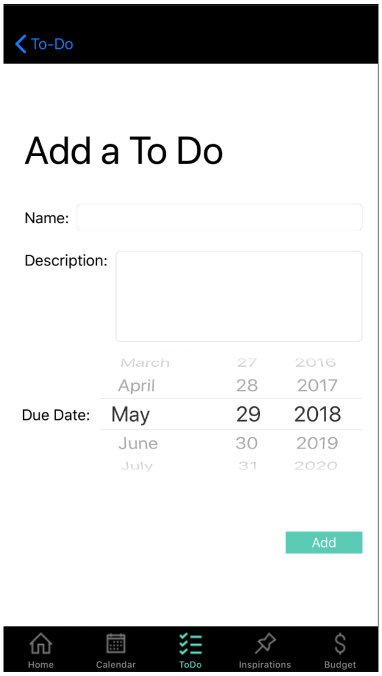


☑ Buy Flowers
04-19-2018

☑ Collect Donations
05-20-2018

○ Buy Utensils
08-20-2018

(i) TO-DO LIST



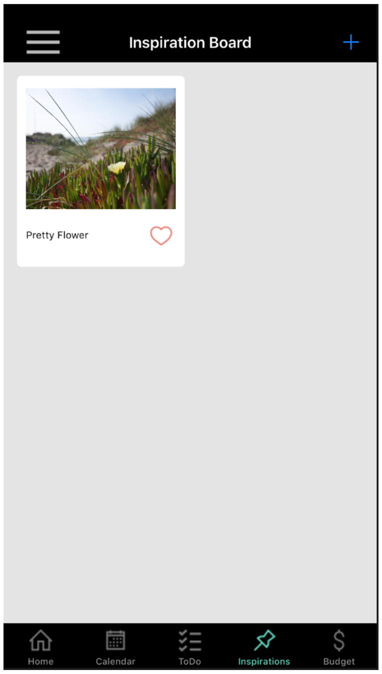
Add a To Do

Name:

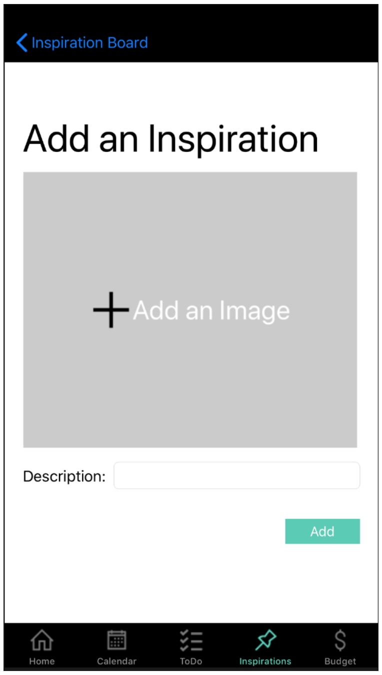
Description:

Due Date: May 29 2018

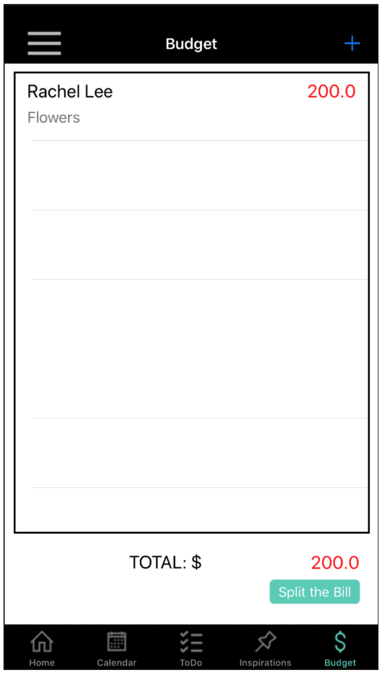
(j) ADDING A TO-DO ENTRY



(k) INSPIRATION BOARD



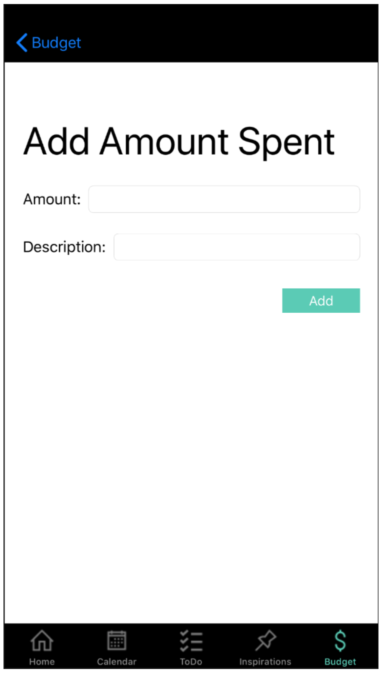
(l) ADDING AN INSPIRATION



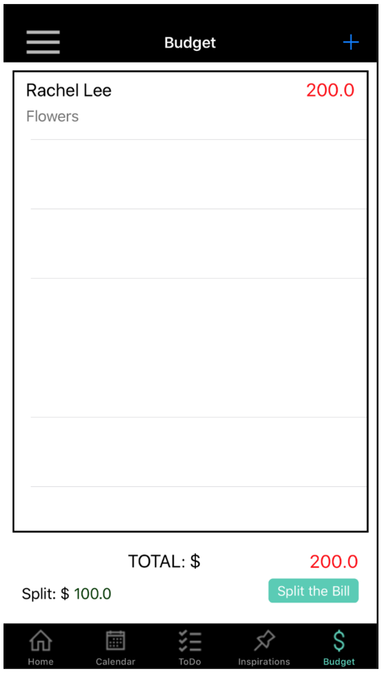
Rachel Lee 200.0
Flowers

TOTAL: $ 200.0

(m) BUDGET INFORMATION



(n) ADDING TO THE BUDGET

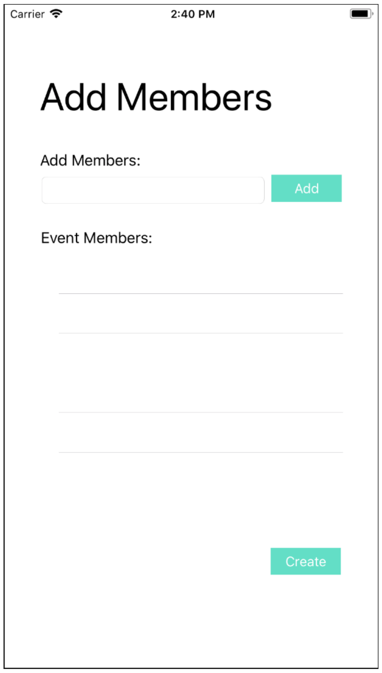


Rachel Lee 200.0
Flowers

TOTAL: $ 200.0

Split: $ 100.0

(o) SPLITTING THE BUDGET

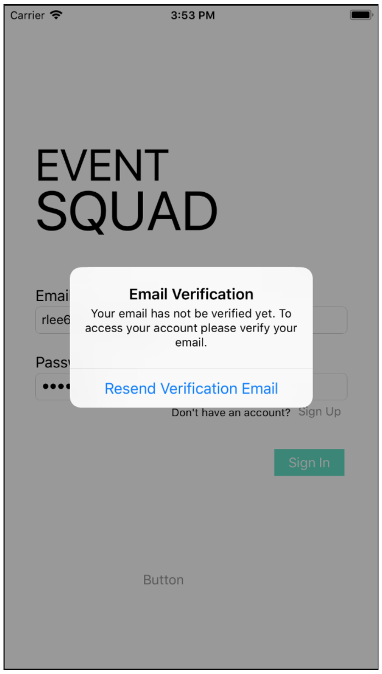


Add Members

Add Members:

Event Members:

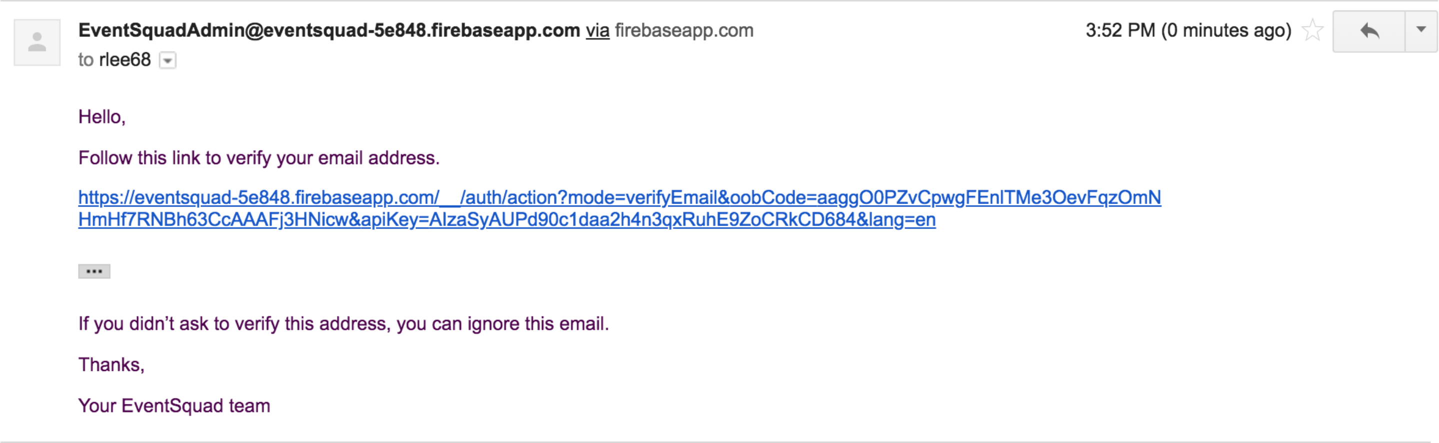
(p) ADDING A MEMBER TO AN EVENT



Email Verification

Your email has not be verified yet. To access your account please verify your email.

(q) EMAIL VERIFICATION SENT



EventSquadAdmin@eventsquad-5e848.firebaseapp.com via firebaseapp.com
to rlee68

3:52 PM (0 minutes ago)

Hello,

Follow this link to verify your email address.

https://eventsquad-5e848.firebaseapp.com/__/auth/action?mode=verifyEmail&oobCode=aaggO0PZvCpwgFEnlTMe3OevFqzOmNHmHf7RNBh63CcAAAFj3HNicw&apiKey=AIzaSyAUPd90c1daa2h4n3qxRuhE9ZoCRkCD684&lang=en

...

If you didn’t ask to verify this address, you can ignore this email.

Thanks,

Your EventSquad team

Figure 4: Verification Email for Account Verification

Figure 4 shows an example of what a user would receive in their email inbox if they register a new account with EventSquad. The user will not be able to use their account if they don’t verify their email by clicking on the provided link of the email.

5 Future Work

A few features we initially wanted to provide are not fully functional or were not released with this version of EventSquad because of time constraints like guest invitations. There are a few things we could also add in the future to make our current features better. Making the split bill feature be able to calculate how much everyone spent and then weight the prices they are charged appropriately. The calendar could also be synced to the users phone and could notify the user of reminders so they would not even need to look at it to see the reminder. For the inspiration board, we could add user comments so people could create discussions on ideas that may have been sparked by the pictures. We could also figure out a better way to invite group members to an event. There are also various background features that we could improve like deleting multiple accounts or events at once rather than one at a time. Some other features that would have been cool to address in this project is accessibility features. Accessibility features are designed to help those that may be color blind or have impaired hearing. If we had more time, we could have implemented different color schemes to help those that were color blind.

6 Project Retrospective

Since this was our first time creating any type of mobile application, we learned a lot of things both good and bad. Throughout the process of developing EventSquad, we learned how to use XCode, Google Firebase and code using Swift. We researched and learned a lot about the process of developing a mobile application and how the front and back end interact with each other. One of the major takeaways we had was just how important the planning process is before even starting the coding part and making sure to update this plan as the project moves along. Especially when creating something we had very little experience in, it made the project more focused when we decided early on exactly what features we wanted to try and implement. This was the first time we had a project span multiple quarters and we had to divide it into multiple releases. This was actually very helpful because it taught us to breakdown the project into smaller chunks and created a schedule we had to follow. We learned about sprints and often used them as deadlines approached.

Another important lesson we learned was to use acceptance criteria for our testing. By listing out clearly what we were going to test for, we thought about what we wanted exactly in a feature and we could know when we were done with developing it. It also made us think of possible bugs and ways we could better design that feature. This made testing easier as we were could test specific inputs rather than inputting random things until we found bugs. This is something we will be sure to use in future projects.

One more lesson we learned was estimating time need for various user stories. Since we had very little experience with mobile development, many of our time estimations were off for the first release. Somethings took way less time than expected, while other times we ran into bugs that would cause delays into the next release which was unfortunate. Even though our estimations got better as we understood mobile development more and how much time it would take to do specific tasks, the initial setbacks and bugs carried out throughout all of the releases which was impossible to predict. Overall, it was not a pleasant experience being behind schedule and trying to figure bugs out. However, in the future we will have better estimates of how long it takes to complete similar tasks and will have answers to similar bugs if we run into them again. Things should go much smoother if we were to continue on this app or any futures mobile apps we choose to create due to this experience we had and can build off of. Figures 5-8 shows the amount of time spent on each release.

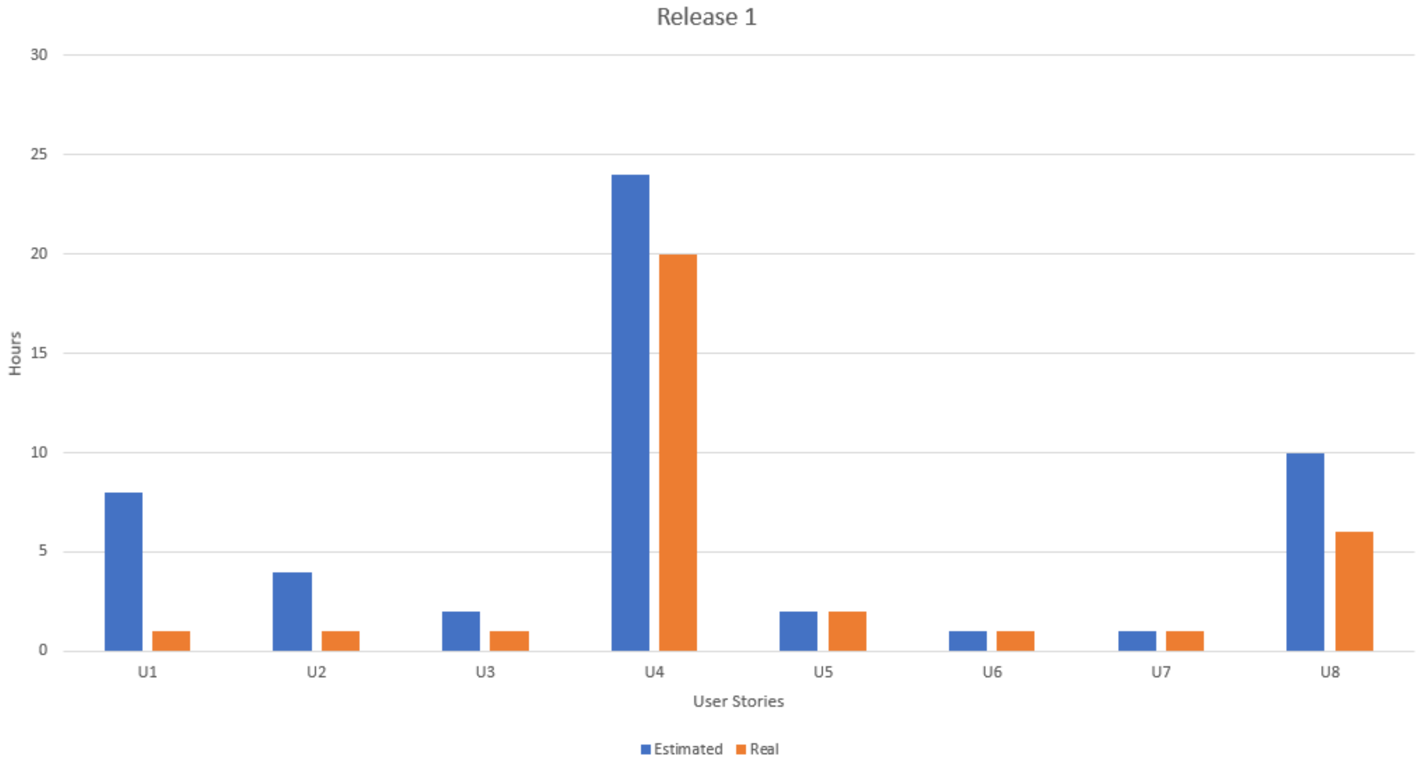


Figure 5: Time spent on Release 1

For release 1, our user stories were the following:

- U1: Creating an Account
- U2: Signing into an Account
- U3: Logging out of an Account
- U4: Creating an Event
- U5: Editing an Event
- U6: Deleting an Event
- U7: Removing a member from the group
- U8: Bug fixes and testing

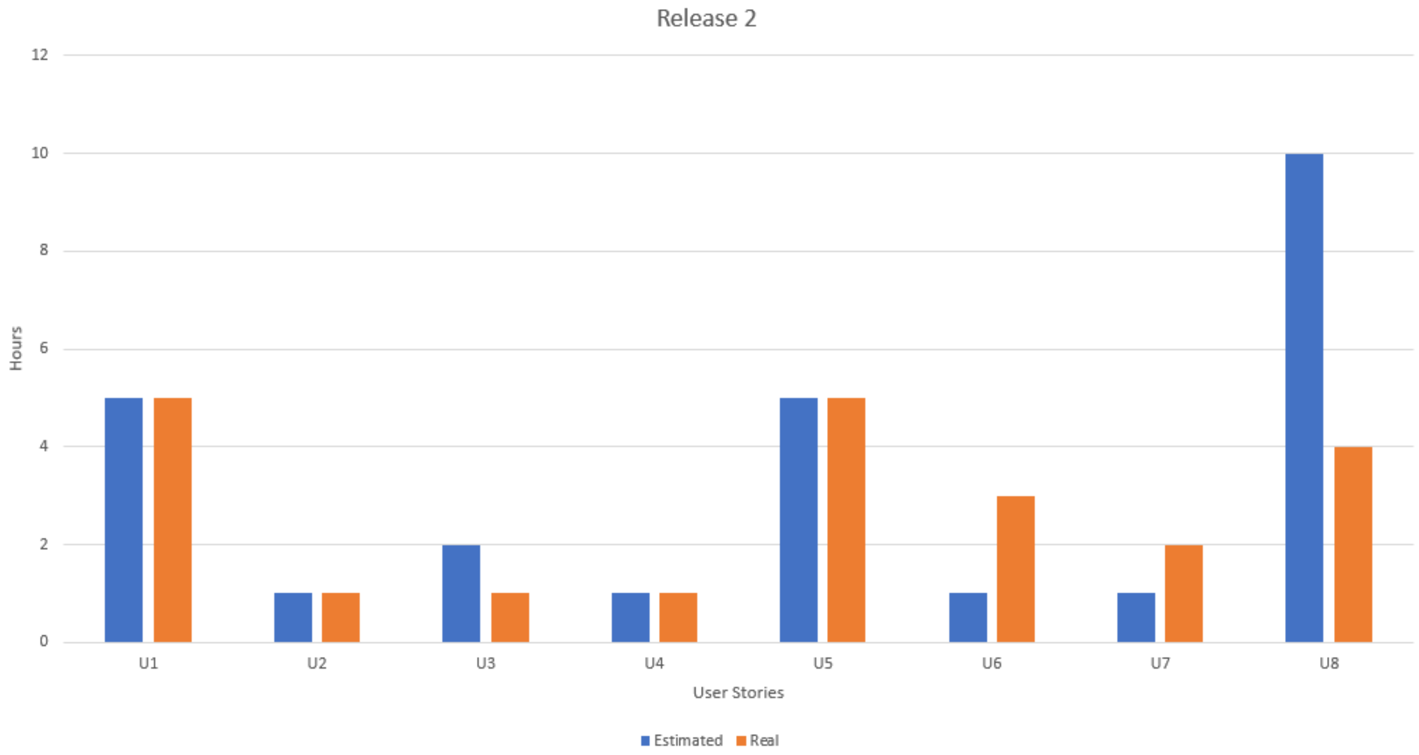


Figure 6: Time spent on Release 2

For release 2, our user stories were the following:

- U1: Adding a To-Do entry
- U2: Completing a To-Do entry
- U3: Deleting a To-Do entry
- U4: Viewing a To-do entry
- U5: Adding a reminder onto the calendar
- U6: Viewing a reminder on the calendar
- U7: Deleting a reminder from the calendar
- U8: Bug fixes and testing

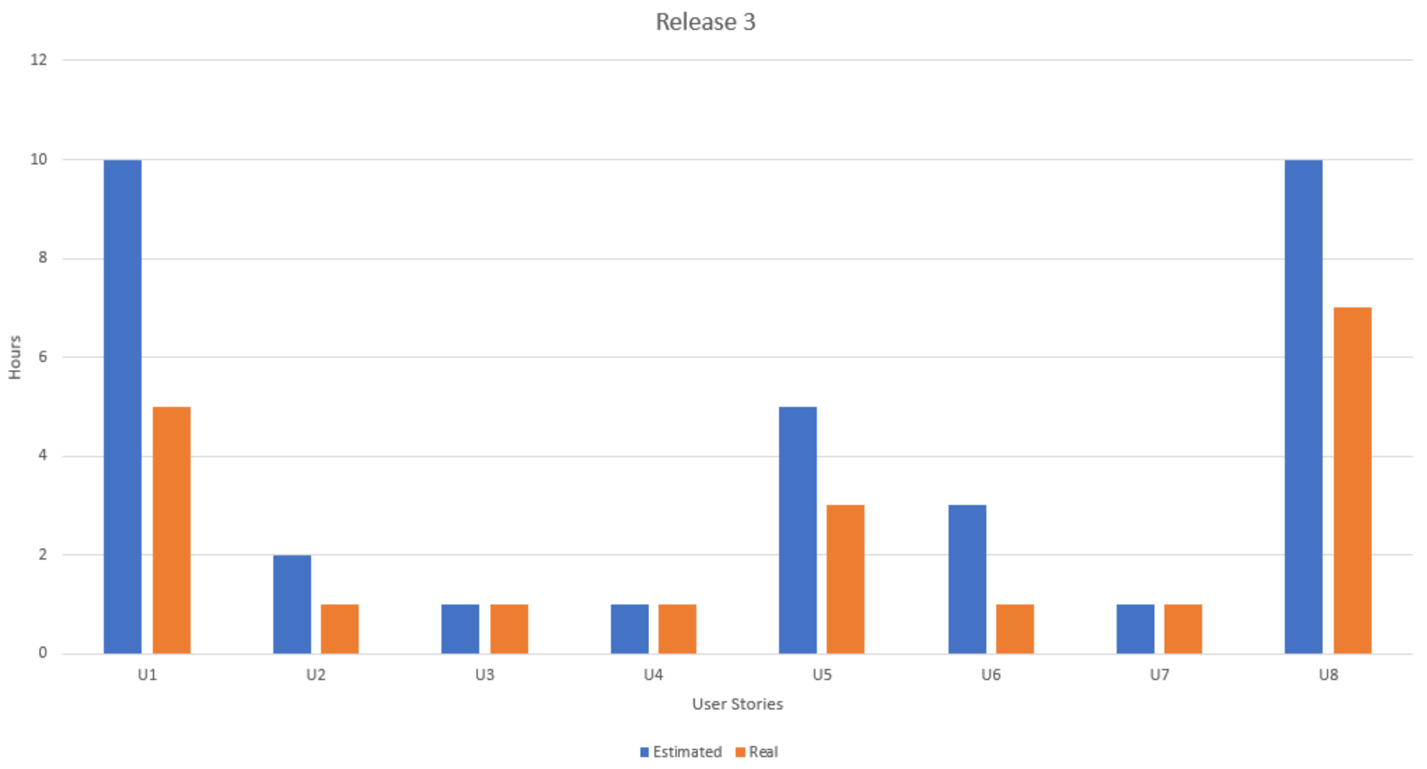


Figure 7: Time spent on Release 3

For release 3, our user stories were the following:

- U1: Adding to the inspiration board with access to local photo library
- U2: Access local photo library for inspiration board post
- U3: Deleting an inspiration board post
- U4: Liking an inspiration board post
- U5: Inputting a budget to an event
- U6: Splitting a budget of an event
- U7: Deleting a budget from an event
- U8: Bug fixes and testing

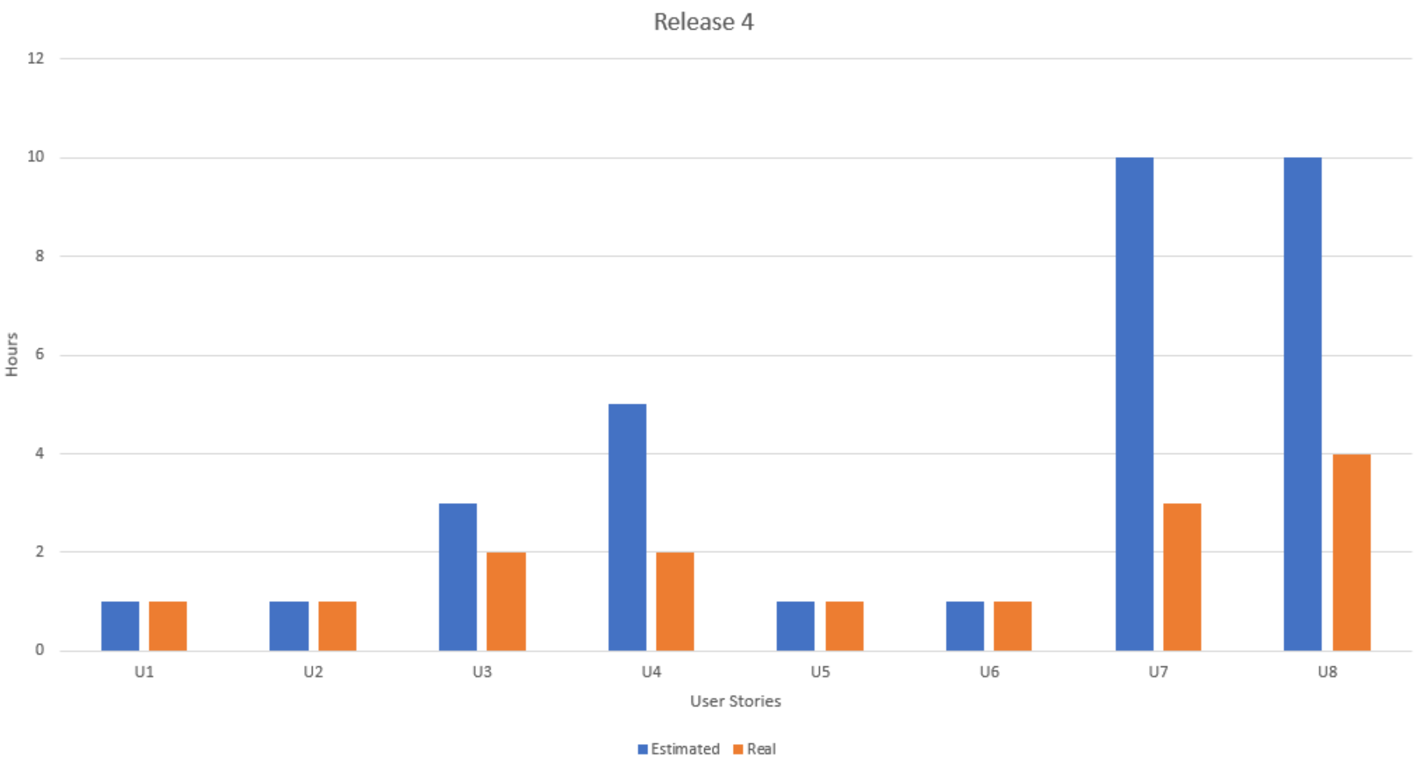


Figure 8: Time spent on Release 4

For release 4, our user stories were the following:

- U1: Email verifications when signing up
- U2: Signing in once email is verified
- U3: Inviting other group members
- U4: Sending out email invitations
- U5: Deleting events for an administrator
- U6: Deleting users for an administrator
- U7: Refining previously implemented features
- U8: Final bug fixes and testing

7 Conclusion

Despite many challenges we ran into when creating a mobile application for the first time, it was a great learning experience for all of us. We learned how to use new tools, IDEs and languages. We were able to learn a lot more about the software development process and how important it is to map out a plan and keep to it. The plan helps ensures that the people working on the project are held accountable and helps keep them on track when working on longer projects that span multiple months. Many of the processes we learned will probably be used at our future workplaces so it was beneficial to have some experience with them in a much lower pressure situation. We hope that we can further fine tune our application and maybe eventually release it on the App Store.

References

[1] “What Characteristics Make Good Agile Acceptance Criteria?” *Segue Technologies*, 26 Apr. 2018, www.seguetech.com/what-characteristics-make-good-agile-acceptance-criteria/.

[2] “69,300 Free Icons (SVG, PNG).” *Test Passed Icon - Free Download, PNG and Vector*, icons8.com/.

[3] “Estimates Made Easy. Sprints Made Simple.” *PlanningPoker*, www.planningpoker.com/.

[4] ThornTechPublic. “ThornTechPublic/LeftSlideoutMenu.” *GitHub, 15 Nov. 2017*, github.com/ThornTechPublic/LeftSlideoutMenu.