Andy Chaney

Carroll university  Fall 2020 – Spring 2021

Seam: a social media application

Contents

[Project Introduction – Update from CSC650 2](#_Toc70771489)

[Project Diary 3](#_Toc70771490)

[Time and Work Log 7](#_Toc70771491)

[Software Architecture and Design 9](#_Toc70771492)

[Use Cases and Functional/Non-Functional Requirements 11](#_Toc70771493)

[Project Design Evolution and Code Samples 12](#_Toc70771494)

[Testing and Risk Mitigation 15](#_Toc70771495)

[Users' Manual 17](#_Toc70771496)

[CSC651 Final Reflections 18](#_Toc70771497)

# Project Introduction – Update from CSC650

This application is designed to help people remember important dates and gifts for friends and family and will also function as a sort of small social media community application. The application (entitled "Seam" to emphasize the shared interests that bind communities together) will function for several different uses: 1 in which it will allow the user to enter calendar dates and notes on others as functional reminders, 2: the application will attempt to use social media buzzword searches to gain suggestions for gifts or reminders that might not be immediately obvious, and 3: because the users are registered to certain community names, it is creating a sense of exclusivity, similar to other social media apps that use private groups to ensure appropriate membership and interaction.

The idea for the project was inspired by my own real-life issues with not remembering certain important dates and not really knowing what to get them for gifts for those events. I wanted to build something that wasn't open to all invitations but would keep a small community connected. My goal is to create something attractive and easy to use that provides a business purpose that would justify investment or scalability if the application would grow as more users begin using it. Ultimately, the project aims to create something positive and unifying that brings joyful experiences to all, like community bulletin boards you might find in public spaces. It could also be an application that a company or organization could use if they were splitting members into teams and entering reminders or tasks that needed to be completed and removed from the board as listed.

Since the initiation phase of the Fall semester, the Spring semester was focused on implementation, testing, and documentation. The project began the Spring semester in a state of having login and authentication working. None of the functionality related to communities or events had been built or implemented. The plan for the Spring semester was to continue following the strategy from the Fall: a waterfall process that relied on 2–3-week sprints and was based off the completion of the earlier modules before moving onto the later ones. The layering aspect of the project was a necessity since many of the later modules could not be implemented until earlier modules were completed. For example, building the calendar application was not possible without having a solid Firebase event creation system that would successfully log events that were viewable and clickable. Thus, the development plan was structured that way and was arranged with project management principles building in designated slack time each module to allow for errors, testing, and educational skill building issues. The included documentation in this report will discuss how the project reached a successful development process and ultimate completion for the Carroll University capstone requirements.

# Project Diary

**9/1/2020**-Project began development over the summer with requirements research. I checked out many different apps that have similar functionality as to get an idea to what worked and what didn't. Also began to think about the business uses of the app. Is it going to be a social media application or a calendar? I like the idea of doing something that involves reminders of presents or gifts to get someone else. My thoughts were that I'm often unsure of what gifts to get a person so having some way to scan a person's social media for keywords or phrases would be a good way to connect with others and understand them better. Discussion over the usages of hashes or some other means of searchable tags is ongoing as I'm not entirely sure what direction I want to take it, what platform or how I want to do this. I know I want to do a mobile app as that’s a personal development challenge I want to have for myself as I don't have the opportunity to take a mobile applications course at Carroll during my time there.

**9/25/2020**- The past few weeks have been primarily learning and deciding upon requirements and process that I want to follow in developing the application over the next several months. After comparing many different platforms and what I want to do, I decided on Kotlin and Android Studio as an IDE for the application. The main reason was that I wanted to learn a new language and I wanted there to be enough resources available for me to realistically be able to do the project. The android community is enormous and there are large amounts of documentation and frameworks out there that you can implement into your project to make attractive and polished. I also know several other folks who have developed on that platform so I have resources available to answer questions and help with any programming difficulties I might have in the project. I have been taking Android Studio courses and doing the Google developer examples they have many good videos explaining what everything is in the platform and language. I've learned the Kotlin language, so I'm able to do java-style functions and imports of widgets and necessary items from the language library as needed. Some of the things to work on is to further learn how everything goes together, I've only been doing simple projects like having pictures listed or learning how to create themes and simple UI processes.

**10/10/2020**-This cycle of development was spent on learning how to implement themes, colors, and building the virtual device used for editing and playing around with the application to make sure everything is working. I was able to create a Google pixel emulator which will be used as the virtual device during development. I learned how to insert an app logo on the menu of the phone, as well as practice using recycle views for the calendar. This didn't really work out the best compared to what I originally wanted to do. My thought was using a recycler view would have the different calendar months and the user would scroll between the months by swiping, but I found a simpler Google widget for that that already has it built. I felt it was easier to just import that then try to re-invent the wheel myself. The trick will be learning how to highlight days with events that are clickable and able to add or delete events on the calendar. I also came up with the idea of when users log in, they join "communities" that are exclusive to their account. A user for example would have to know the specific name of the community (this is based off Mastodon using "instances") that would generate a feed of all users' events. So, for example, if you joined the "csc" community, and all the events you placed on your calendar for reminders would be visible to the rest of the community, you could see your classmates' items including their birthdays which would generate the ability to do a search for their social media for ideas for gifts or to even just learn more about them.

**10/25/2020** Current development cycle has seen the implementation of the different widgets involved with the application. I feel like progress is coming around quickly. I was getting a bit discouraged in having some difficulty in getting the database helper to work and getting the login info to save the SQL command to the tables, etc. But after watching some YouTube videos and looking into the Android developer libraries I think I've figured that part out. That is a big part of what I need to accomplish going forward. I didn't know exactly how Kotlin, and Android Studio worked exactly in terms of how the different resource and pages worked. As I learned more, I learned for example that a function call was necessary to move throughout the app and it doesn't involve separate pages like html and web design does, but it involves making different resources visible. It's a lot like Angular in certain ways in that you have different folders for items involved with the same page or same part of the website or application. Going forward into the next sprint and development process will be to get the database working 100% so user profiles are created upon sign in. Making sure data created is visible in the landing page, and then making sure the drop-down menu has the different clickable items that will create function calls for the main app design and uses.

**11/8/2020**- I decided to shift gears in a sense for the last month of the project and focus on research of the functionality and on completing documentation to a high standard for the final presentations for the semester. After the meeting, this past week it was clear in listening to Professor's advice and hearing about Clay's project the need to have solid software engineering fundamentals in place going forward as time becomes scarcer and the ultimate deadline in the spring approaches. I don't have any fear of not completing the project, but I also want to make the development process and design process easier than it's been so far. This semester has involved a lot of trial and error which meant that I would look up frameworks or watch tutorials on how to do something but since the Android community is so diverse and widespread there's a lot of alchemy in a way in terms of getting everything to fit together. A lot of time was spent going back and forth because something I was implementing wasn't going to do what I needed to in the end. For example, there was a calendar widget through Google, it wouldn't allow clickable date popups that would allow you to add/delete or view your reminders, it was a date selection function for other types of uses, such as selecting a date and entering into a form, but in this case I want the dates themselves to have the user event data entered into it, so I just need to take the time to find the right frameworks and right way to put them together. I also shifted gears a bit and switched from SQL lite database to Google Firebase. It seems to work a little better and I'm hoping that it'll provide more scalability and will store the user data on their cloud instead of requiring it to be stored on my machine or to have to get a server that would store it. Firebase also has the potential to add authentication and login functionality that I won't have to code myself.

**11/28/2020-**Spent the last week or two just going through Android Studio tutorials and doing project research with the goal of setting myself up for a solid month or so of programming during the Christmas break when the schedule is lighter. Went through the developer guides at <https://firebase.google.com/docs/database/android/start> to learn the database and programming needed to implement this. Maybe it was just a little bit of nerves and wanting to make sure the project works as I want it to, but I took a step back the last few weeks and sort of wanted to get my ducks in a row and get my documentation and everything looking good for capstone. Began coding the Firebase login functionality based off of the above-mentioned developer guides.

**Coding Development Cycle – 12/1/2020 to 12/15/2020**- The first real coding for the project has begun, and the project database functionality has been implemented. After some experimentation to ensure that it will do what I need it to do in the end, I decided to go with Google Firebase and, after a bit of learning and trial and error, developed a no-SQL JSON database library and relational tables that, I feel, effectively links via relational tables the users to events to communities. I needed to take a bit of time to ensure that the tables were saving data the way that I intended, and that the data was retrievable and made sense for a logical structure for the application. These saves of data are done through a specific setValue type of method call which navigates the node tree setup and finds the specific area to save. We would then make a getValue function call to retrieve the information and store it in a text field holder. At this stage in development, I just have basic functionality of setting and storing values. I haven't built any forms or displays or anything like that, so that will be the next step in the process to learn about the different views that are available.

**Coding Development Cycle – 12/16/2020 to 12/31/2020**-Time was spent on this cycle building out the activities that would retrieve and set data in the database. I created an activity that would insert records for a user to create a community. The user would save the name and a unique identifier number for community and it would have a list of all the users in the community. I also created a list view/search view for the list of the communities, any user could log in and click on one of these communities and it would join the user to the community and thus make the community events visible to them. I also built in an event adding activity items to the database, that associate each event to the user and community they are currently logged in and assigned to. At this point in development these are all still very much in the prototype stage and haven't been tested or debugged, but for the first time in the process I can see how all of this will work. Next iteration will be spent on designing different activities and navigation between them. I am starting some requirements research for how the layout will look in the end, and how I can build different widgets in that are easily accessible.

**Development Cycle 1/2/2021 to 1/25/2021-**The prior development cycle was a success as the application now boasts sign in, registration, and event creation database structures. The user can now navigate through button click events to the different activities on the application. This development cycle was spent on beginning research into an algorithm, program, or method that would provide information or a social search of gift ideas or interests to those within your community. I don't quite know the best way to do this yet. I'm looking at maybe doing it via an email report that would do a social search on the user and then send an email with ideas. Or possibly a third-party widget that I can somehow implement into the app. The other main goal that I'm working on during this period is researching frameworks for the best way to create attractive UI and layout for the user. Specifically, I am looking at ways to make pop ups, card views, and scrolling feel intuitive and attractive. As the project moves towards the implementation phase of the Spring semester, I want to make sure that requirements are refined, and processes put in place for a smooth development schedule and stress-free coding period.

**Development Cycle 2/1 to 2/14-** For this cycle, the goal was to implement the date picker and notification delivery system for setting reminders. There was also some research done into whether a recycler view would make more sense for listing the events that the user created or some other kind of widget. I also created logo icons for a top navbar as well as activities for the profile view and editing. As a result of this work, I successfully implemented the recycler view widget for viewing events through scrolling up and down, I was able to build the notification system to notify the users of events at a certain date or time, and I found several UI themes that have worked well. Some of the earlier iterations and cycles have bled into the current one, but it was an extremely productive 2-week period in which several items came together and the project heading forward towards completion. The notification system works by using a custom helper class that takes the date added from the event creation and then times it, so it posts the reminder at the given time and date. This will take some refining to make sure that the date is correct as there is a fair amount of date formatting involved with this part of the functionality.

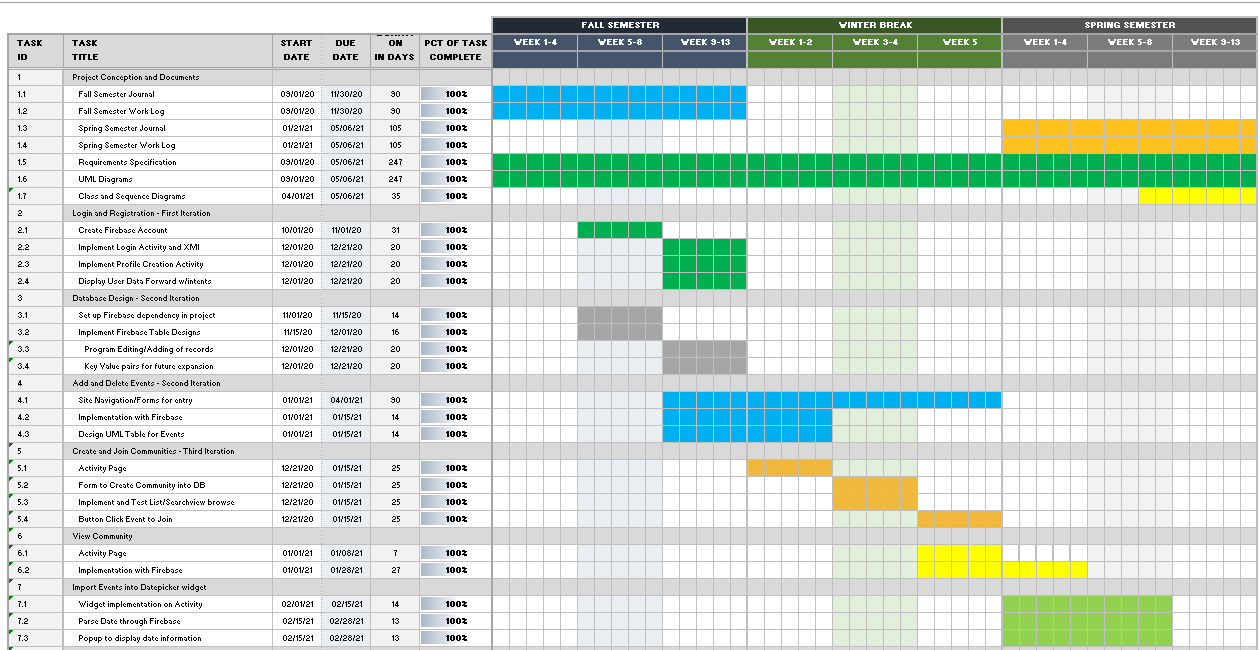
**Development Cycle 2/15 to 3/1-** With the event creation working as it should, development in this cycle continued with the UI design and the date picker navigation, which included working on button widgets, aligning linear and constraint layouts to look more attractive and to build in a "dark mode" setting for the application, for those using it after hours or with a dark theme. Also work started on debugging and testing different facets of the application. Making sure conditionals checked and worked for items such as logging in and not overwriting prior entries that were in the database. Also making sure that user can delete items on the community via the button, organizing the community list to show only future events and can connect to different communities as needed. Some errors were found with the dates not sorting correctly that will be addressed in a later iteration as development here is moving towards implementing profile editing and deletion in the next sprint. Some errors found with the event deletion not picking the right position to delete, so further research will be required there as well to figure out how to do the adapter view item deletion.

**Development Cycle 3/1 to 3/15-**At this stage of the project, I continue to wrap up the debugging and testing, and completing the last handful of functionalities needed to get the project up and running. The UI is mostly done at this point, all activities have constraints built and are aligned for different devices and different types of phones. I remain stuck on how to approach the gift idea/social media aspect of the project, so my plan is now by the end of March to have everything built and completed for the calendar aspect and then I can take the final two cycles of the project to figure out the best way to do that. I'm thinking I'll use the first part of it for requirements elicitation and then use the remained of the time to iterate through building something that works. I'm not sure if it'll be successful but that is the plan at this point. Also continue to research the best way to do event deletion on the recycler views that I have built, which uses the same adaptor for both community view and for specific date view.

**Development Cycle 3/15 to 3/30-**Great success this cycle as the profile view and editing activity is now working, the date picker/calendar day viewer is now essentially done at this point, everything is working as it should, updating and deleting as needed in the database. For the next couple of cycles, testing and documentation are the main thing to get done outside of the social media algorithm and method. The project also saw a successful implementation of profile deletion activity, so the user could delete their account should they wish to leave the application. I would peg the coding completion to be around 85% at this point, and moving forward it's refining, testing, and fixing errors, and mistakes in the code. I will also try to clean up and make my earlier code more elegant. There is a lot of code I wrote earlier in development that I found I could for example wrap in interfaces or shared methods. So, to make the project code more attractive and scalable, that will be the approach.

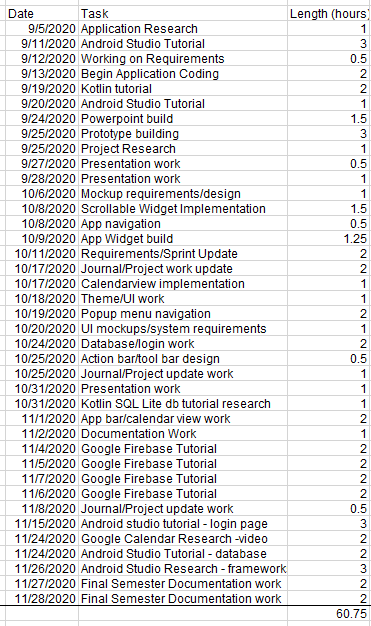
**Development Cycle 4/1 to 4/15** A lot of positive developments during this cycle. I finally got the event deletion working through the adapter as it should, and I also built a few customer comparator and formatters to fix the earlier date parsing issues. So now everything is sorting, displaying, and deleting as it should. These were huge headaches that I had for the most part completed, but I wanted to eliminate any out of place data that would reflect poorly on the project. After struggling for quite some time to figure out how to use the social media aspect of the application (and cycling through several iterations of requirements "changes") I believe I finally settled on a solution to the issue. There is an open-sourced algorithm called JSoup that works with Kotlin and android devices that can scrap a selected web page (in this case, someone's HTML twitter page) and then display the results. So, my thought here was to collect the data from the HTML scraper, parse it through a non-common word counter and display the most frequently used words in the textbox. If the user decides that those words make sense, they'll file it as a comment of an event on the calendar and make it visible to the community. I haven't done any testing or development of this so I'm not 100% sure it's going to work, but it's a positive step in the right direction. I also sent the program via zip file to a few users to start testing and playing around with it, to make sure there are no unforeseen issues or errors that I hadn't detected earlier. I also exported the application to a physical android device to play around with it and it works. I was really shocked to see it in physical form on a device, and I have to say it was a moment of pride to see this finally come to fruition. There were a lot of moments of self-doubt this Spring where I felt that this wouldn’t come together so to see it reach its goal was a great feeling.

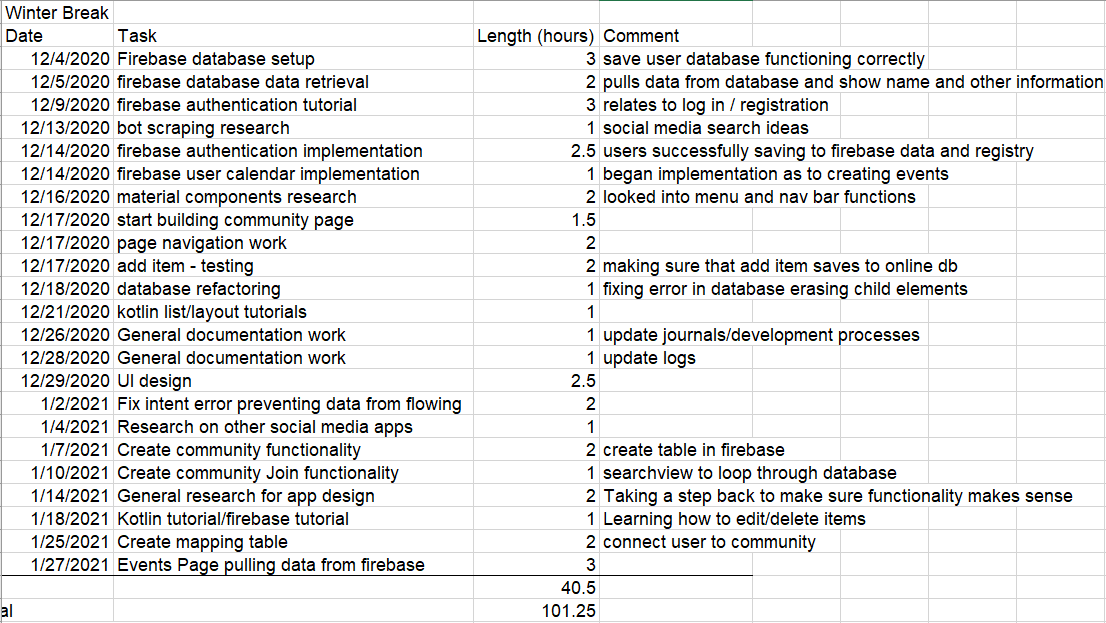
**Development Cycle 4/16 to 5/6** The final iteration! The program is reaching completion at this point, all functionality is working, and user interactions reported back positive experiences with the app. Via these interactions in which I also had them test out testing scenarios that I designed, all testing is complete, and because of this feedback several issues with verification and data entry were fixed. This cycle will focus on fine-tuning design and display of XML, documentation for final submission and final demonstrations, requirements updating, and diagram completion. I've drawn some conclusions on the nature of this project, mainly that Android development is super powerful and useful, and Kotlin is a great language, but that my hope and wish is that the documentation and community can make it more accessible to those that might want to learn. Even those going through the official Google Developers platform can have some struggles and the learning curve can be steep. But, in the end, the project was done and completed, and we stayed within the development cycles and project management plan, and we can turn in a working project with solid documentation to support it.

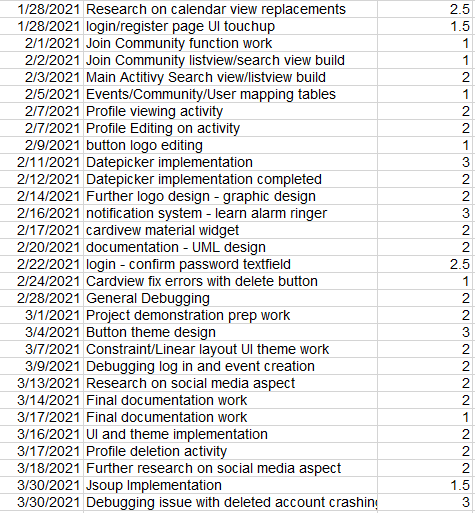
**Project Management Development Chart:**

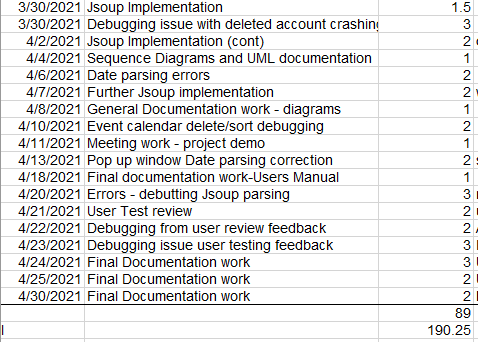
# Time and Work Log

Fall Semester 2020



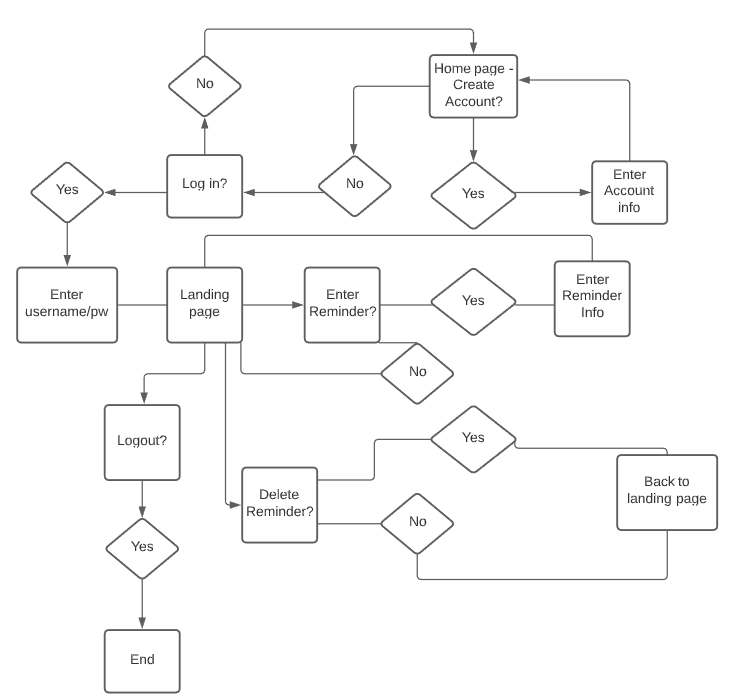


Spring Semester work log – project came in at 190.25 hours from start to finish.

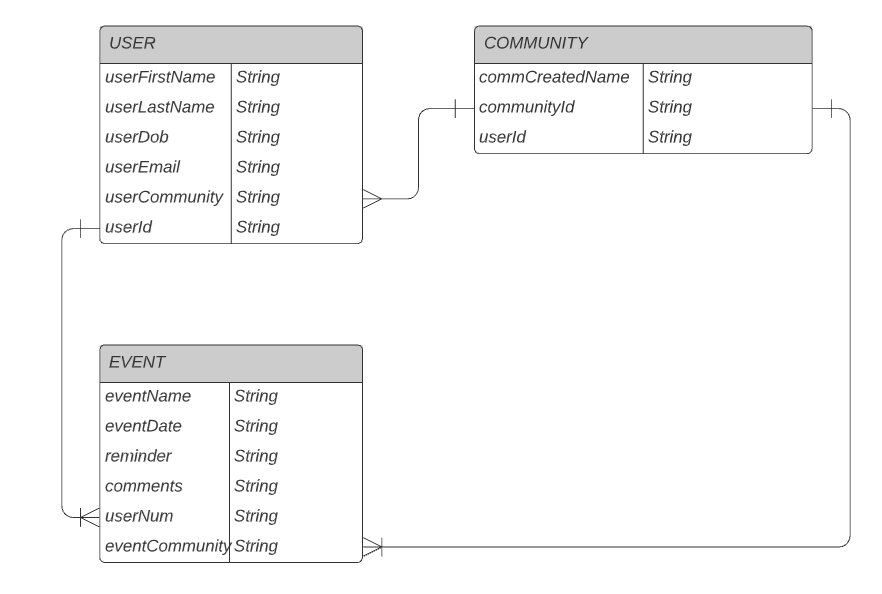


# Software Architecture and Design

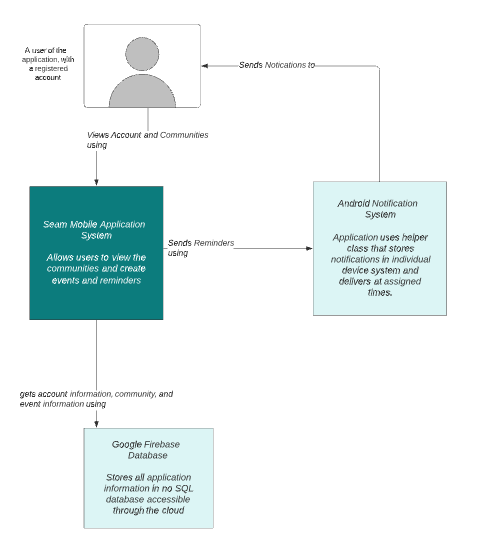
Sequence Design of User Experience-

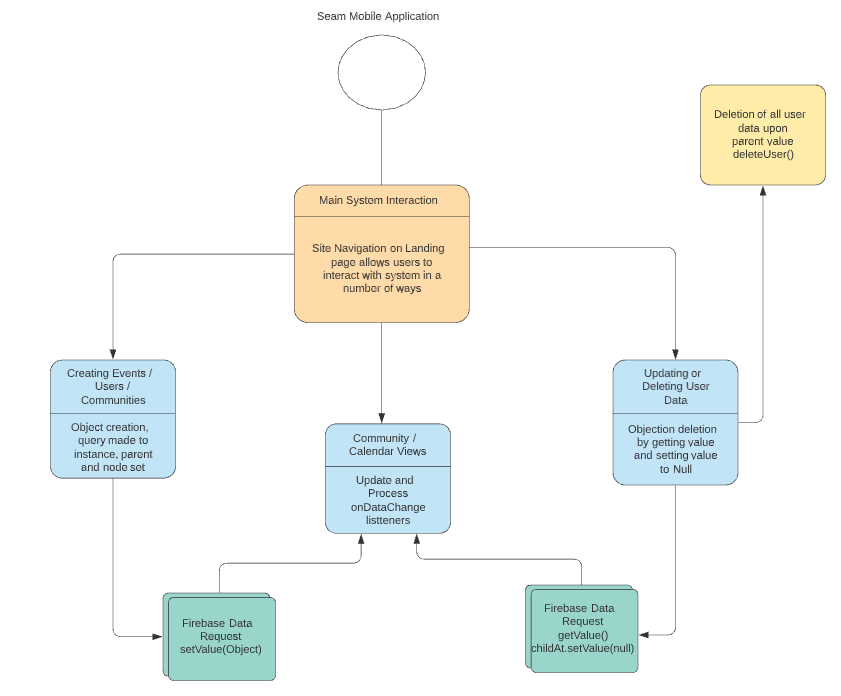


Entity Relationship Diagram showing database structure-

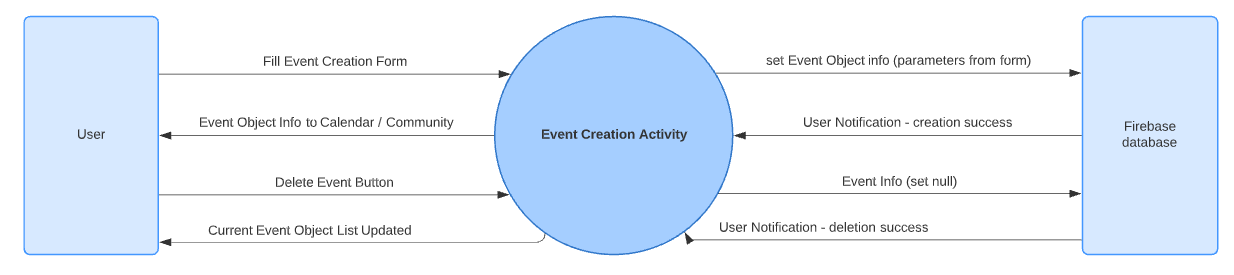


C1 and C4 Context Designs: User Experience:

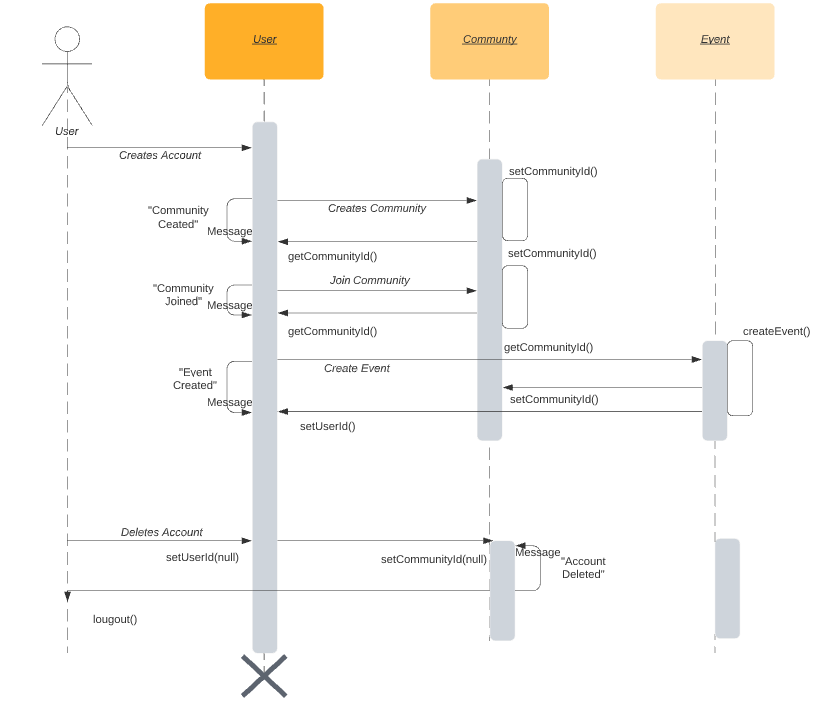




Dataflow diagram, component level (event activity):



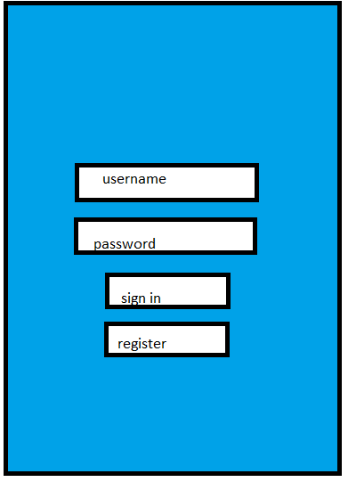
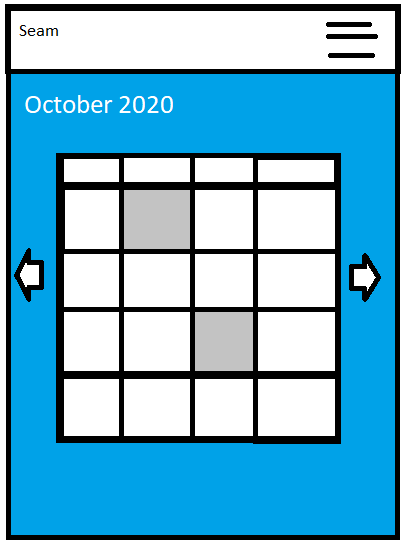
Sequence Diagram of Program Use:



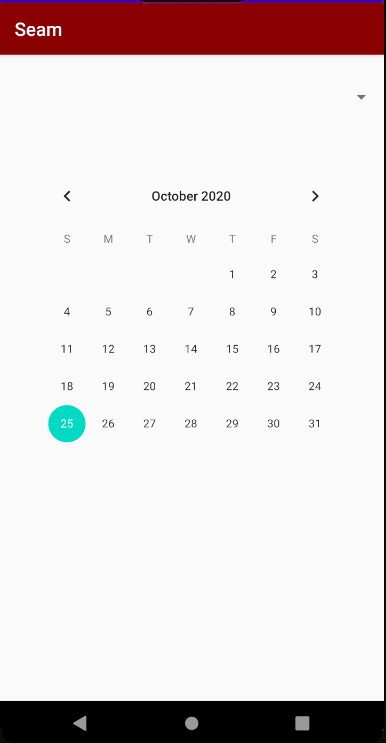
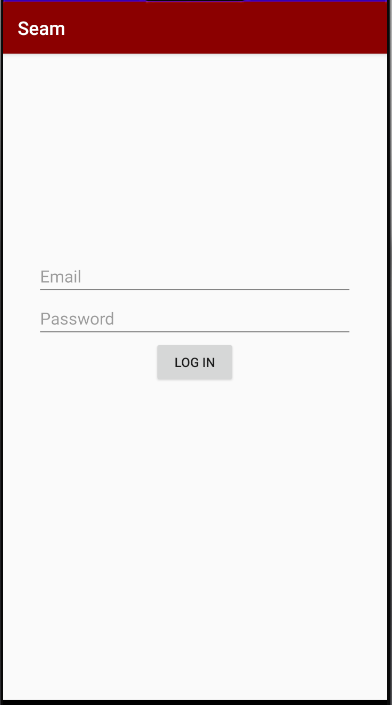
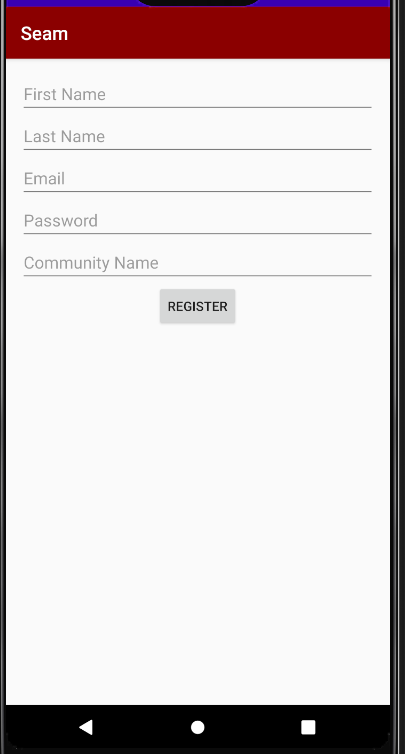
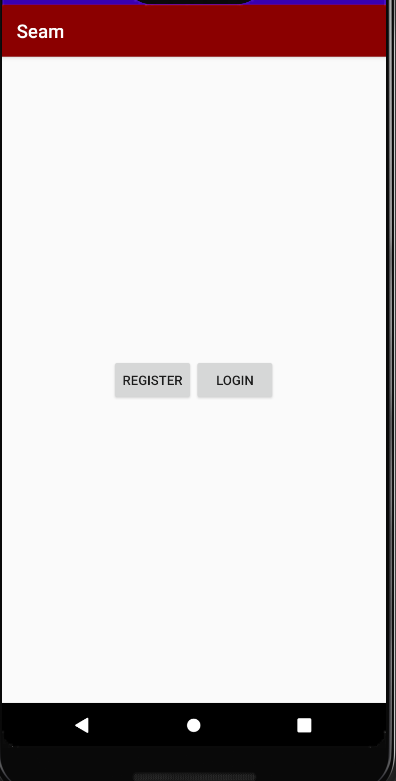
# Use Cases and Functional/Non-Functional Requirements

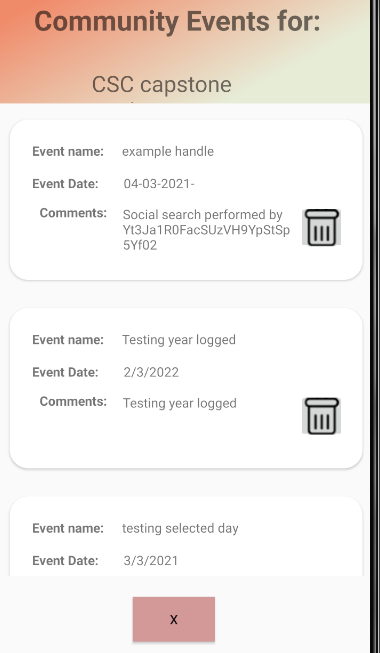
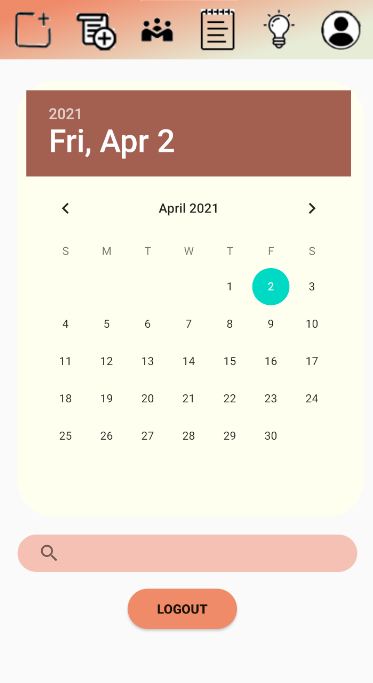
* Create Username/Register: System shall accept input from user in the form of email address and password creation. System shall also accept data in DOB form as well as community name. System shall validate that the input for username is in email format by ensuring that there is an @ and domain in the email address inputted. Will output error message if validation fails. Will also run validation to check if there is anything returning a null value, if data check returns null value, system will output an error message. If validation successful, system will output success screen and return to log in page.
* User log in: System shall accept input from the user and check it against the registered user database. System will run a loop through database to find matching username, then check input of password against associated password connected to username in the system. If username and password are accepted, system will output a success message and direct the user to the landing page which contains user specific calendar and event actions and activities. System shall also access and display username in upper left corner.
* Create reminder: System shall accept input from the user creating an event that will be stored and accessible on the calendar. System shall create a unique event ID# for each event and will store associated information regarding the event in each database column. Output will be a visible event not only on community board but also on personal landing calendar.
* Delete reminder: System shall accept input from user through click event that erases the reminder from the user calendar. Output will be that the system accesses the event id and nullifies the data surrounding that event/reminder so that the reminder is no longer visible or clickable.
* Search calendar: System shall accept input from the user and search the database for instances of that input. System will run a loop through all calendar and community events and display an output activity page showing all data returned.
* View community: System shall connect user to all other users in the same instance of the community's name. System will search for community name that the user is logged into and will run a loop for all events that have a deviation of 1 day from current datetime. This function call will display all community events and reminders on that day and the next day and will run a loop, outputting and displaying each event that shows up. System will sort events chronologically and list them on that activity page.
* Logout of system: System shall, when user starts the click event, clear the user data, and return the user to the registration/login activity screen. System shall output a message stating that the user is successfully logged out.

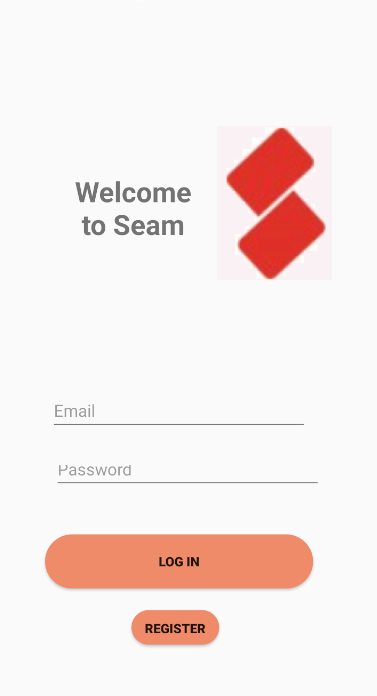
# Project Design Evolution and Code Samples

Project Mockup Oct 2020:

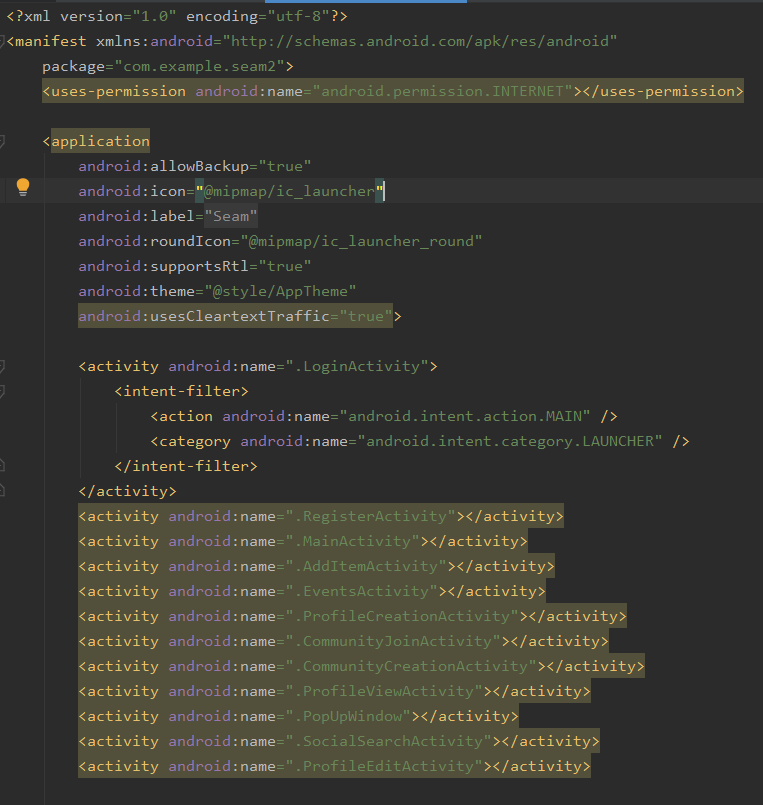
Project Progress Nov 2020:

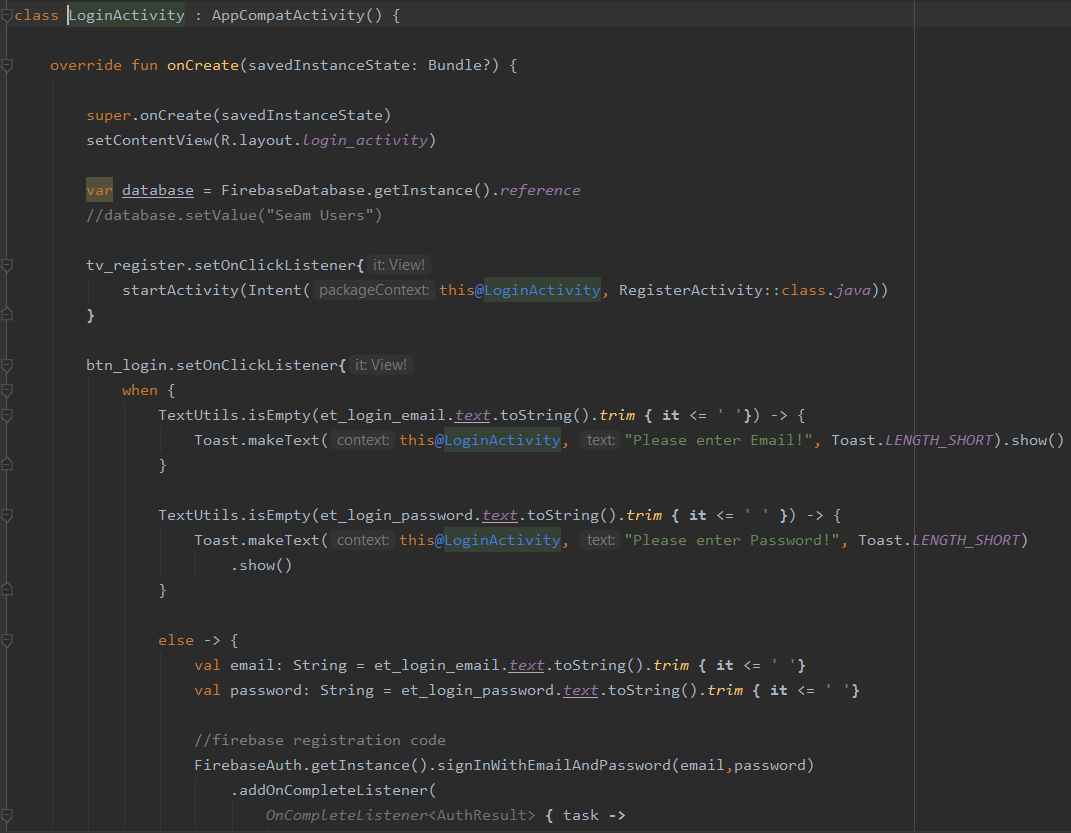


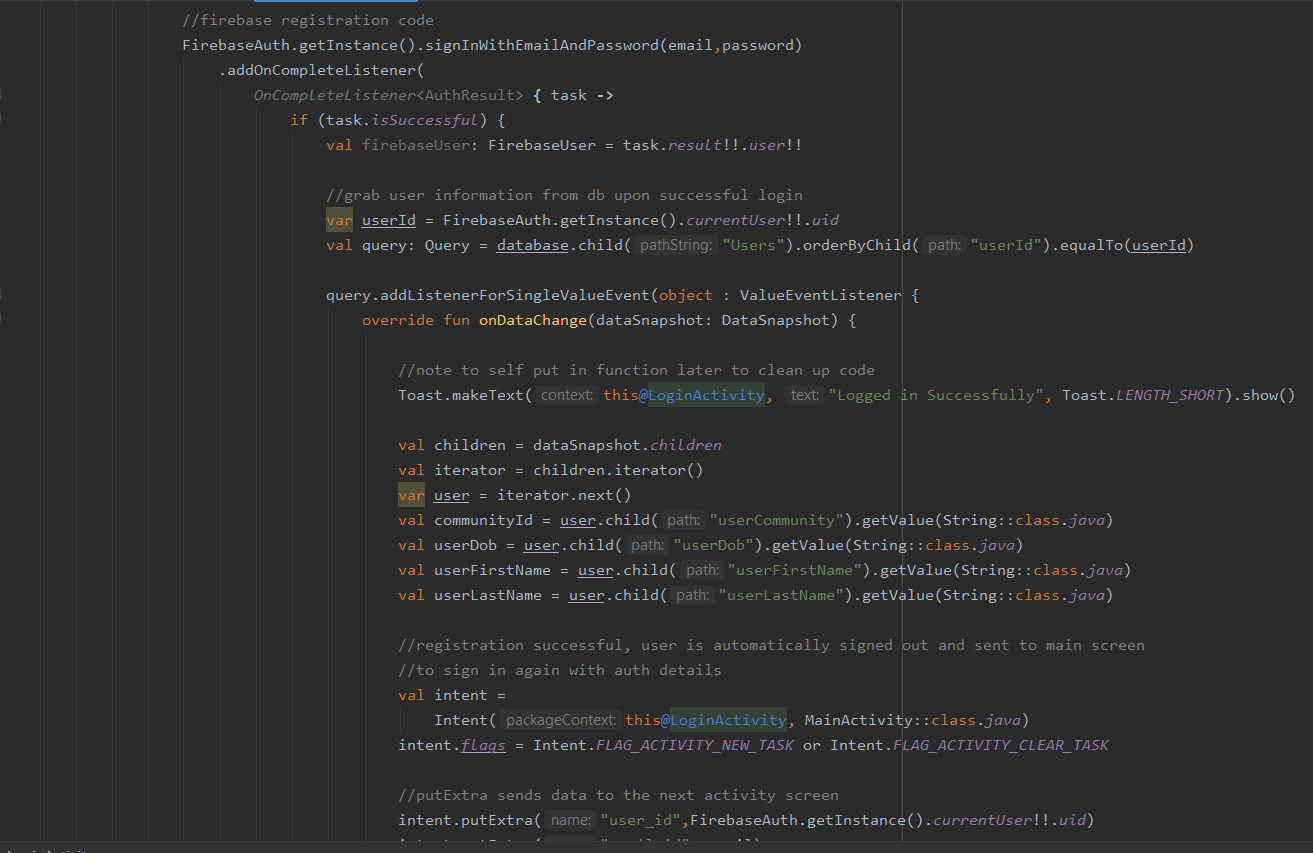
UI storyboard design as of March 2021-

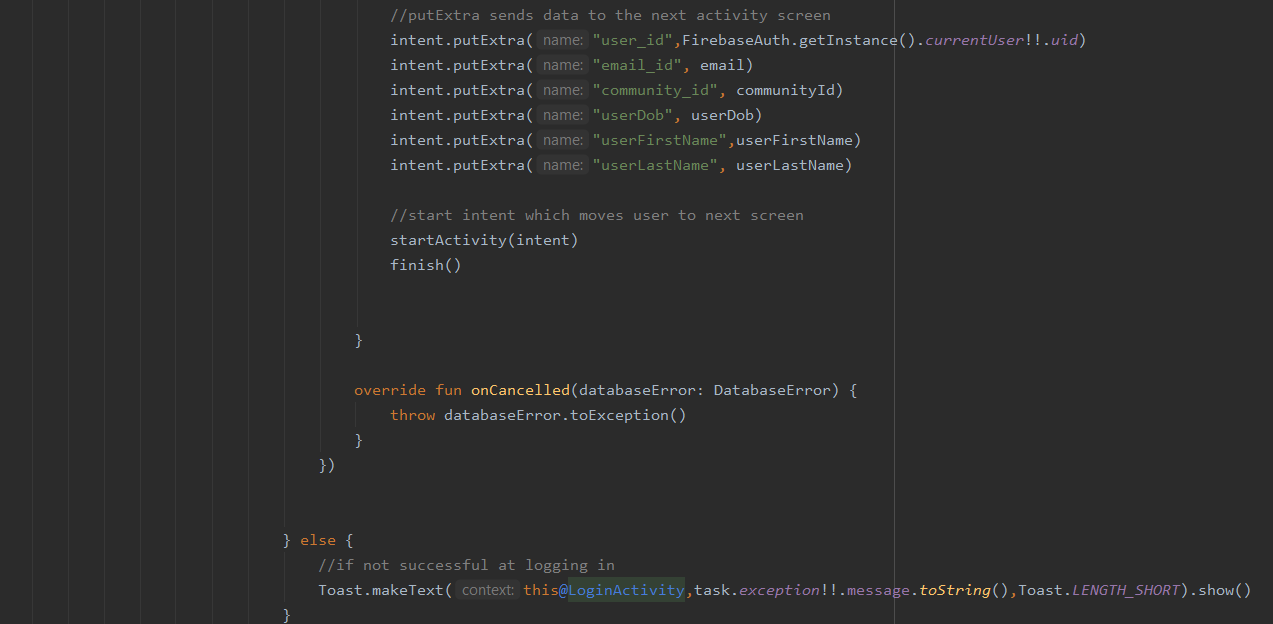


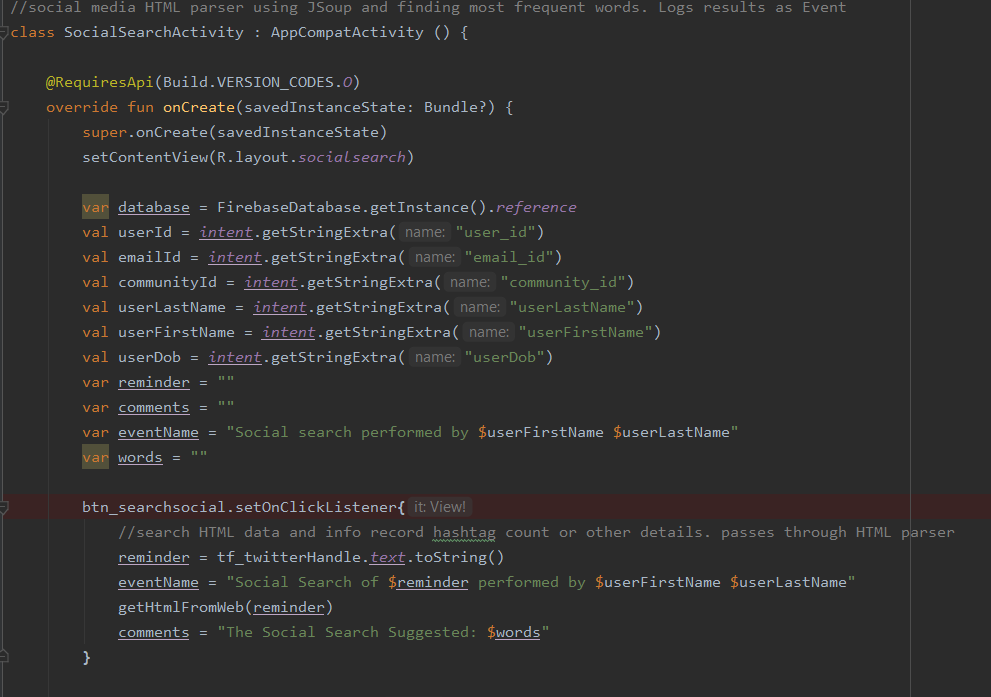
Final Code Samples – These code samples will demonstrate the Notification System and JSoup Implementation (showing test URL), they will also demonstrate the basic Firebase functionality and adapter views that work with associated XML files, as well as how the Firebase authentication system works for user login and verification. The way that the program works is by storing certain information into intents that they then pass back and forth between activities. We use queries to do database instance calls and the program has Event, User, and Community objects that are created that we use the constructors for to create the appropriate data to interact during the application.

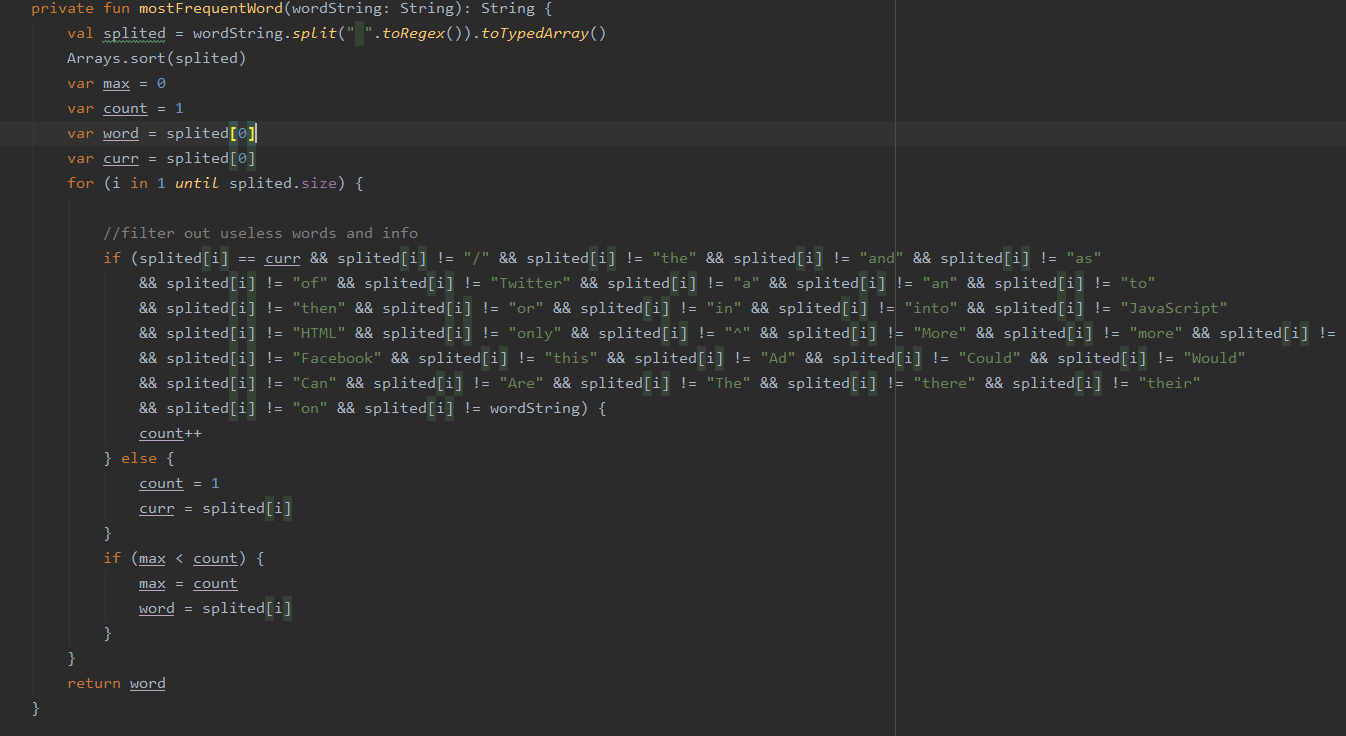


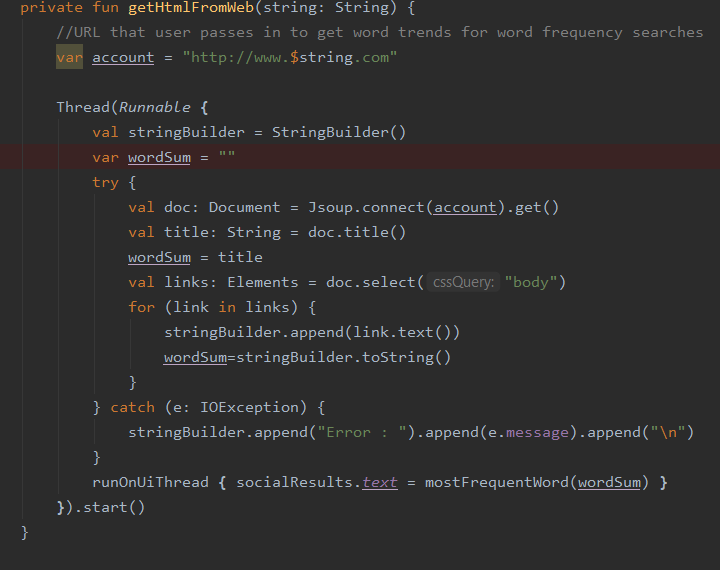


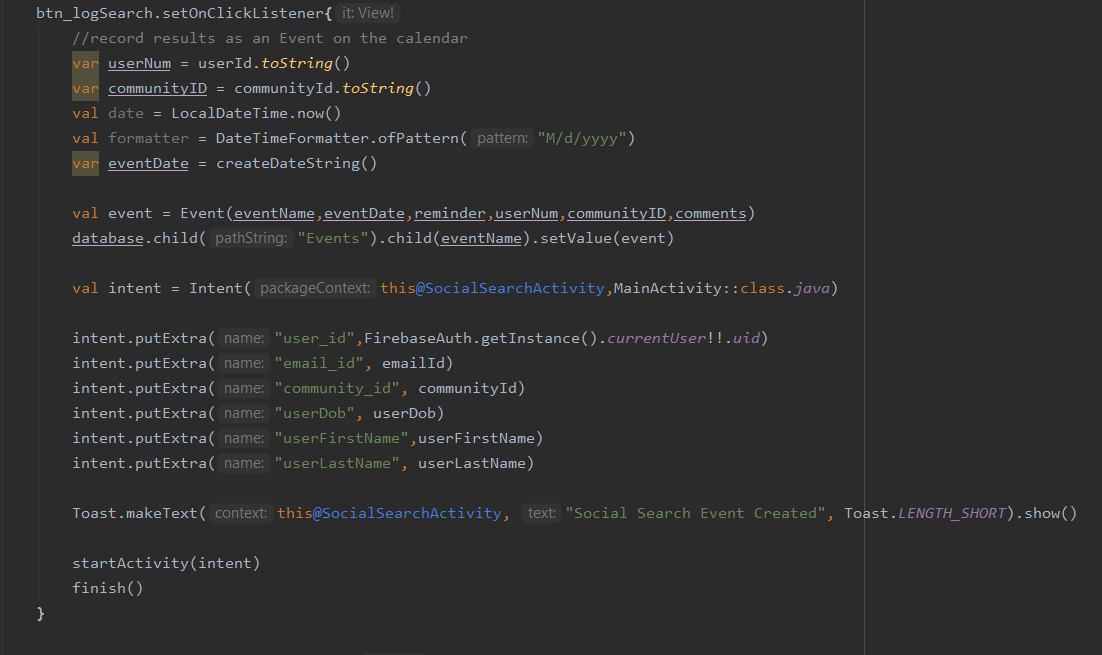


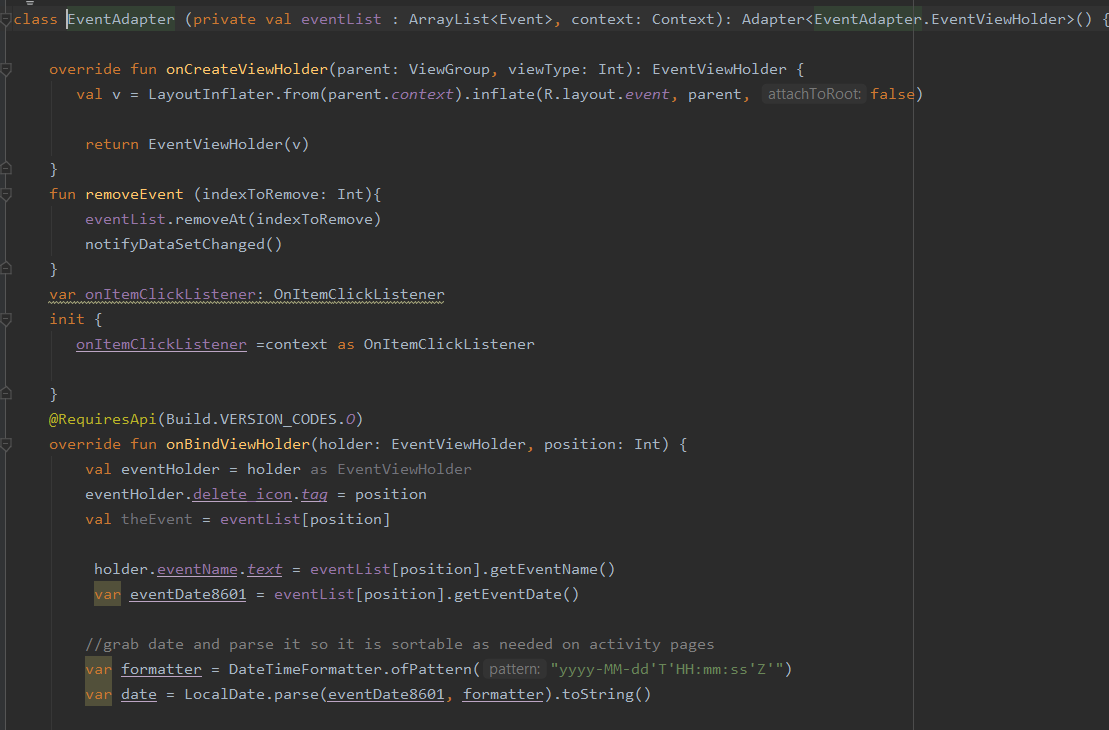


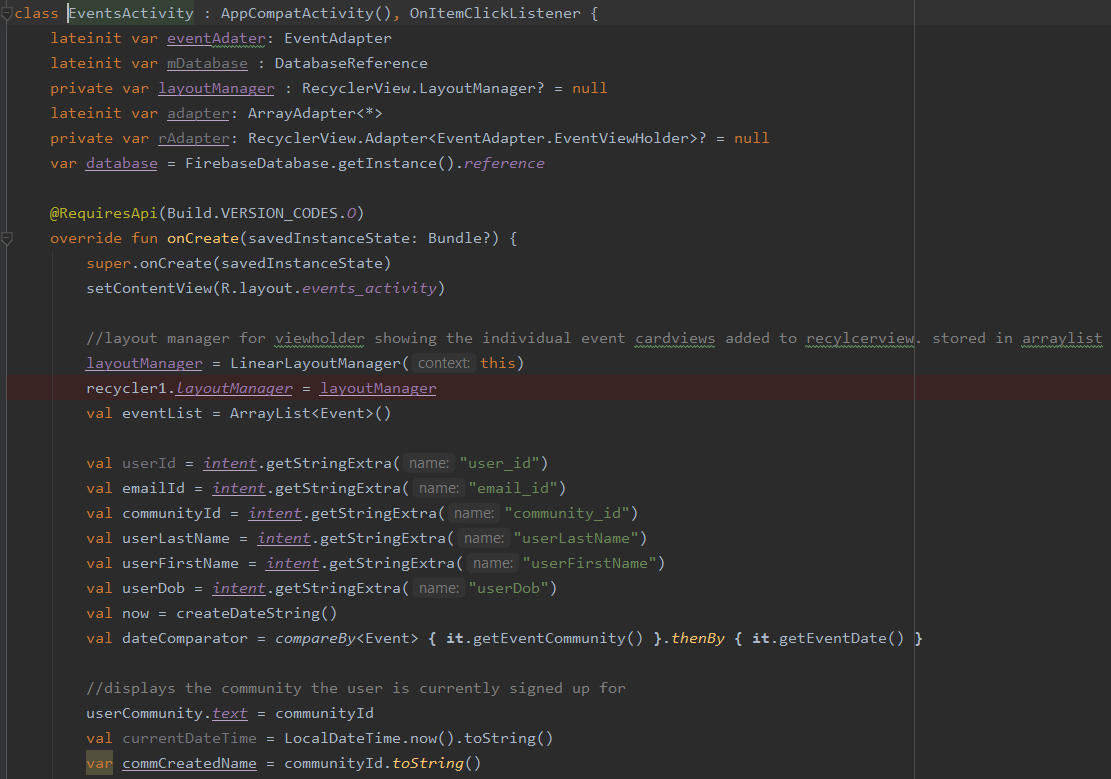


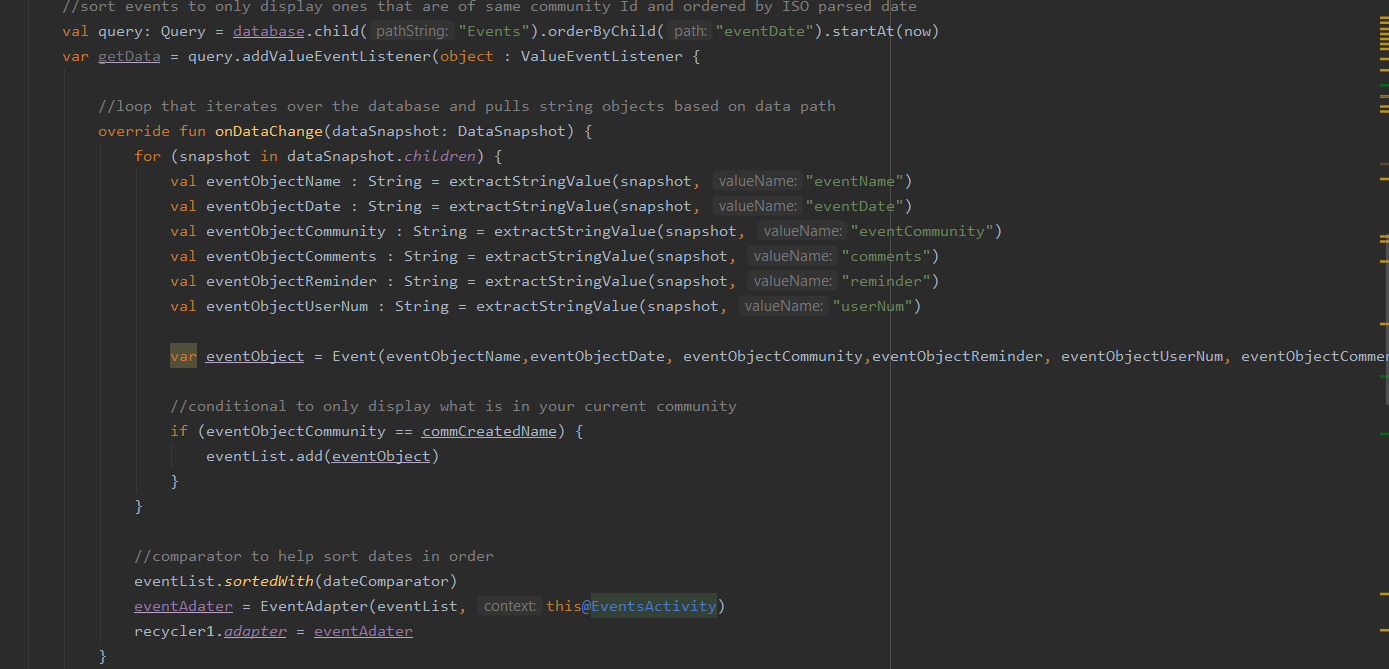


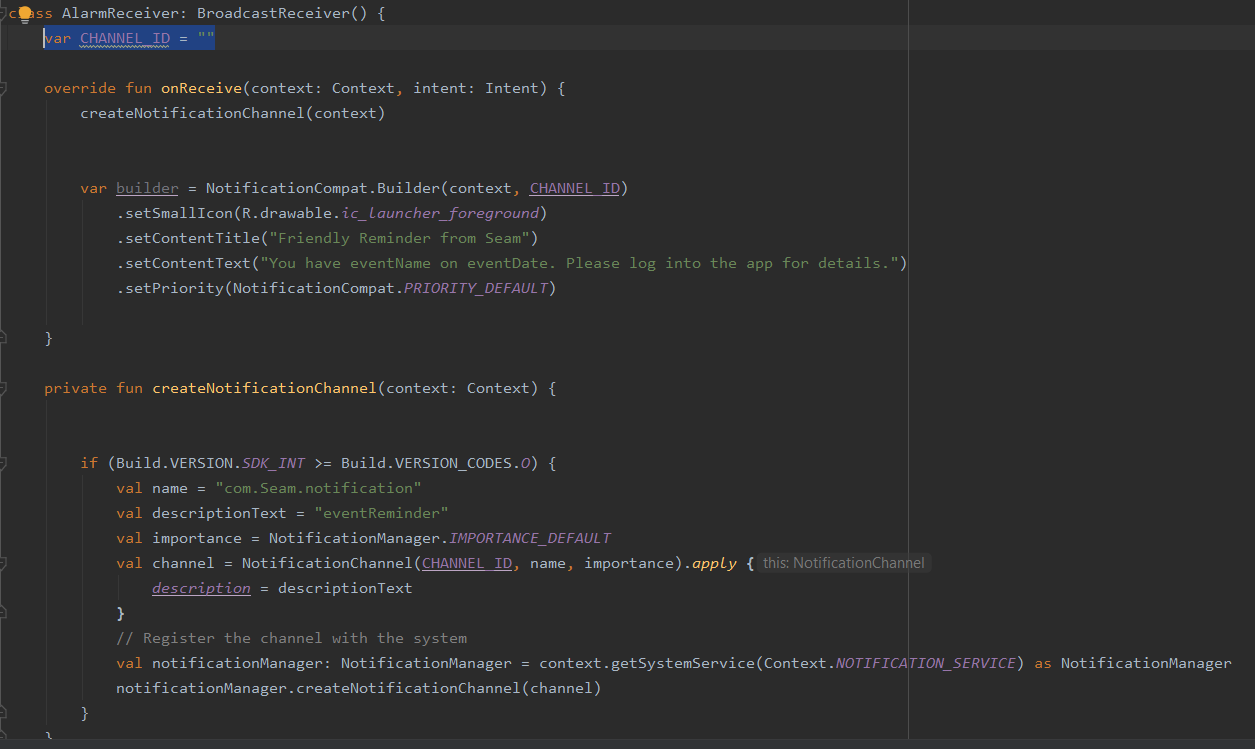






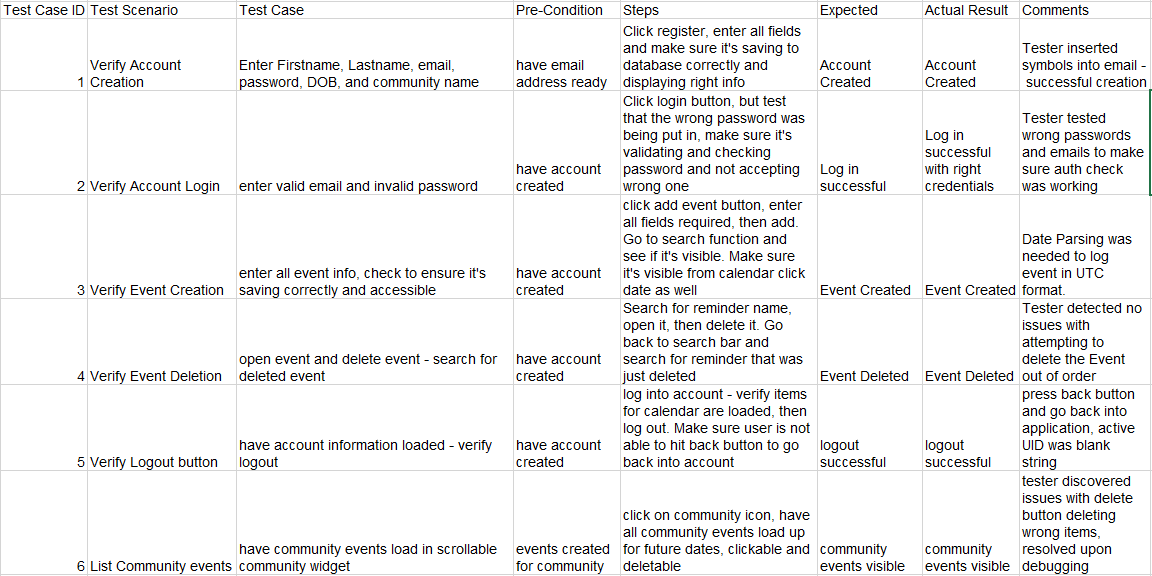








# Testing and Risk Mitigation

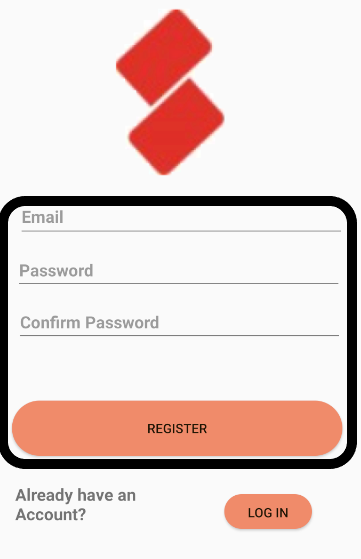
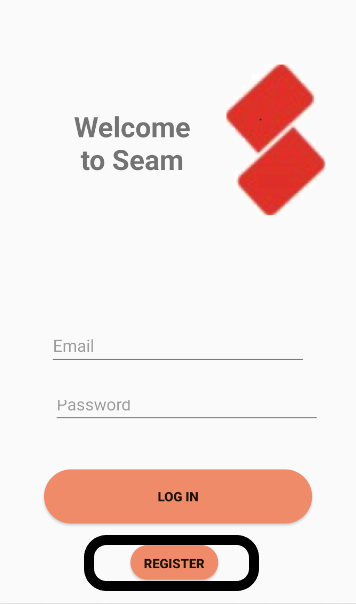


**Risk testing commentary and conclusions**- The above tables showed the full Spring and Fall semester testing plan. In terms of risk mitigation during the Spring semester, the project is moving forward from the previous iterations and has already taken advantage of solid practices such as pseudocode and functional and non-functional requirements from earlier development cycles. As a result, the project demonstrated a lower risk profile and did not suffer from catastrophic failures. The development plan going into the Spring was still built in a bit of mitigation of risk though the context of delivering a working and well-thought-out product. As the project began to demonstrate working features, the risk was further mitigated by ensuring that each layer built in works before starting the next one. Each unit was tested to localize errors and to make sure larger searches for bugs would be unnecessary once other functionality had been rolled in.

For example, making sure that all portions of adding items was working and debugged before working on calendar and community apparatus. By doing so, it kept errors and bugs contained into sections that were easier to locate and fix. An error that would have been catastrophic at later stages such as a null entry during profile editing was caught and it prevented a total deletion of the tables and database. Testing will also help to discover unforeseen bugs that might show up and could impact the final project deliverable. One other method of testing was to get a fresh set of eyes on the project to find holes and issues with database and program design. I enlisted several other users to download and use the program in many ways that might be typical of a user, as well as many ways that would be typical of attempting to "break" the program to find weaknesses or errors. They tried the program with different data types and tried to enter values that might cause an error. The feedback and testing they performed caught a few other issues such as dates not sorting correctly and notifications being assigned to the wrong month, since in date parsing, January is considered month 0, and needed to have a 1 added to each date calculation. There were also date parsing bugs in pop-up windows that were caught such as the startBy and endBy such as the current date acting as a greater than instead of a greater or equal than. These were caught during testing and fixed to ensure that final implementations did not have compounding errors that made project deliverables unattainable.

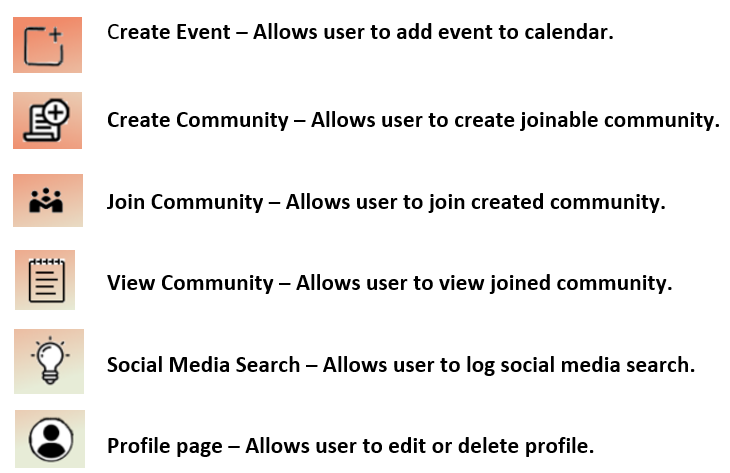
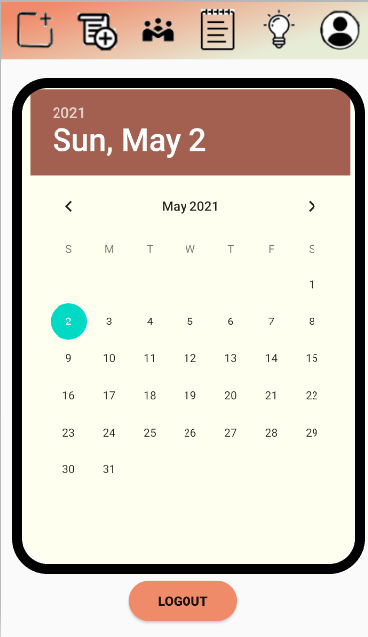
# Users' Manual

Included is a step-by-step instruction for users to use the Seam Application. The following steps will show the user how to create an account (1), editing and delete the user profile (2), navigate the landing page (3), create a community, or join a community to begin posting and reading events (4), and how to use the social search functionality to log specialized events on users' social media accounts. (5)

**Step 1 – Create an account and log in.**

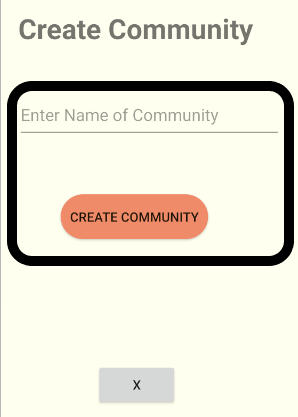
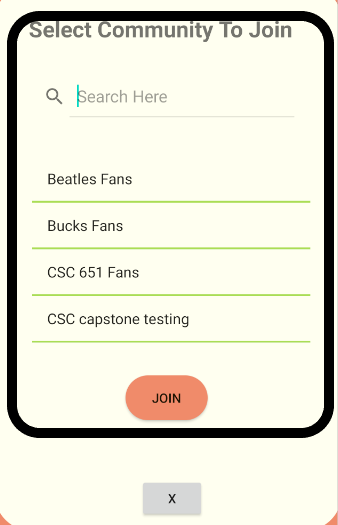
The user will go ahead and click the register button on the initial log in screen if no account exists. If user has already completed this step, they may skip it and proceed to step 2. Upon clicking of the register button, the user will be taken to a registration form screen. Complete all fields, click register, and complete profile screen, and user is registered with Seam. They may then click log in and proceed to step 2.

**Step 2 – Landing page navigation.**



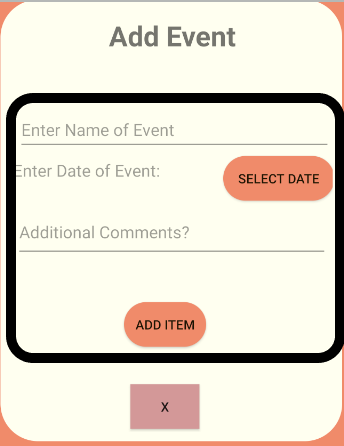
The user can click the specific day on the calendar and view that day's events, or they have several widget buttons on top that perform central functionality for the application. The user also has a log out button on the bottom of the landing page that will allow the user to exit the application and close their logged in session.

**Step 3 – Find a community or create a community.**

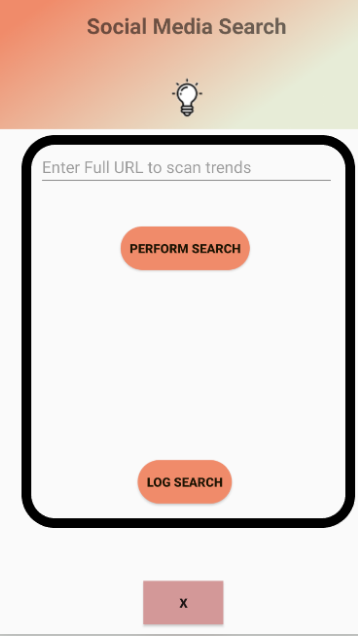
The user may now click on the create community button and enter the desired community name into the text field. You know belong to this community! Your community events will be blank, and other users may now join your community and enter events on it. If you choose to join an already existing community, press the join community button, search for the desired community name on the scroll list, and click join. This communities' events are now visible on your event and calendar widgets. You may only belong to one community at a time, but there is no limit to the number of times you may leave or rejoin a community.

**Step 4 – Start logging events!**



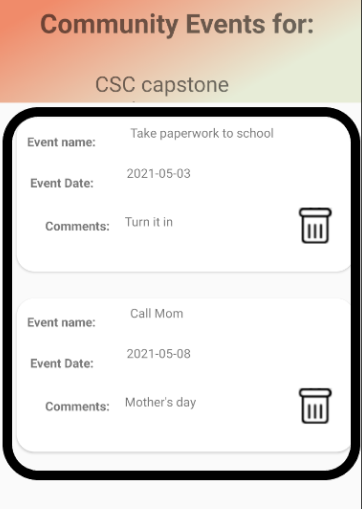
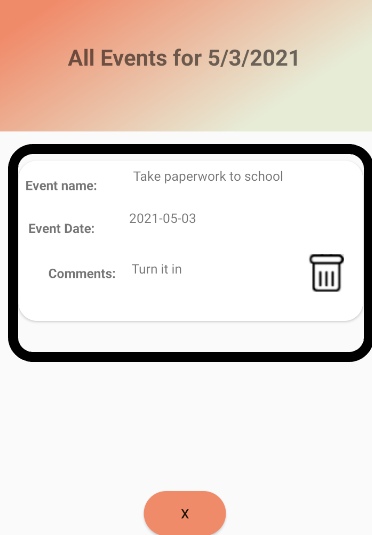
Here the user can begin logging events to the community that they have joined. Each event you log will have a timed alarm for the day and time of your event and will notify you as a reminder that you have the event that day. Users can not create null events and must enter at least one character into the event and comment fields. When this is complete the user will click add item and a prompt will appear to confirm that your reminder was successfully added to the database.

**Step 5 – Log a social media search event via HTML parser.**



If the user wishes to add social media search data as an event to the community, they may log an event through the social media parser. This parser uses the specific Twitter handle or URL that the user inputs and will scan the HTML entries of that page. It will display the most common information mentioned on that page and give the user the option to log that as an event to the community or not.

**Step 6 – View your events on daily and community calendar.**



Here is where the events that you've logged will appear. Both in the calendar and the community widget on top of the landing page. If you have created the specific event, you also have the option to select the trash can logo in the event and remove it from the calendar. These events are sorted by date, for the specific calendar they show that day's events, for the community, it will show all the future events past the current day.

# CSC651 Final Reflections

As the project ended during the Spring semester, I took a step back to view from the project from a distance to assess the successes and failures from the project design and implementation. The main overriding sentiment I'll be taking forward from this project is a sense of confidence in my ability to learn a new language and apply new principles as I'm developing skills. There were many sleepless nights during this project when I felt lost and that the goal was unattainable. This was mostly due to the nature of Kotlin and Android projects and the lack of solid documentation and examples from which to learn. The language and the Firebase platforms have changed so often over the years, that the developers' documentation and assorted tutorials often had out of date methods and ways to implement basic functionality. My prior background and experience were JavaScript, HTML, CSS and so forth, so I had the benefit of learning how to do design and programming with frameworks and a robust community out there to help learn and develop. Where I would get frustrated would be the amount of time spent simply to learn something easy such as page navigation. Whereas in HTML and website applications this is easy with just a link, Kotlin and Android required activities and intents and it wasn't always clear what and how to do certain things. In the end I figured it out but choosing a capstone project to learn a new language and complicated platform was probably not the best for my mental health. But and this is an important point in the whole endeavor, I preserved. I stuck with my schedule and my principles. I was honest with myself when something wasn't working. I started over on the project twice once I found better frameworks and approaches to accomplishing what I needed to do. I wasn't afraid to admit I didn't know how to do something and sought out resources and prior examples so that I could grow and build my skills.

Looking at the project moving forward, I'm going to continue working on it, refining it, and seeing if I can build something useful out of it. I really love the idea of social media platforms being an indicator of useful social predictors. This is a simple usage of the idea, just scraping and counting words on an HTML page, but I'd like to further develop the algorithm and research into something deeper to see if we can help strengthen the "seams" in our communities. In terms of future project maintenance, this will into the issue that I did when developing, it will be important to continually update the code and keep it current as it could fall into obsolescence quickly**.**