Interview and Design Documentation

- You must interview at least 1 person to gather their input;
- You must writeup the results of the interview (not a raw transcript)

Determine the following:

- What is the expected functionality of the app?

This app is meant to keep track of roleplay scenes for certain characters.
You will be able to create/delete characters, edit their names, and search for them on the main screen.
You can also assign events to them, which can be viewed on an individual character and as a list under their month. Events can be edited by navigating to the event from the character screen or the events screen

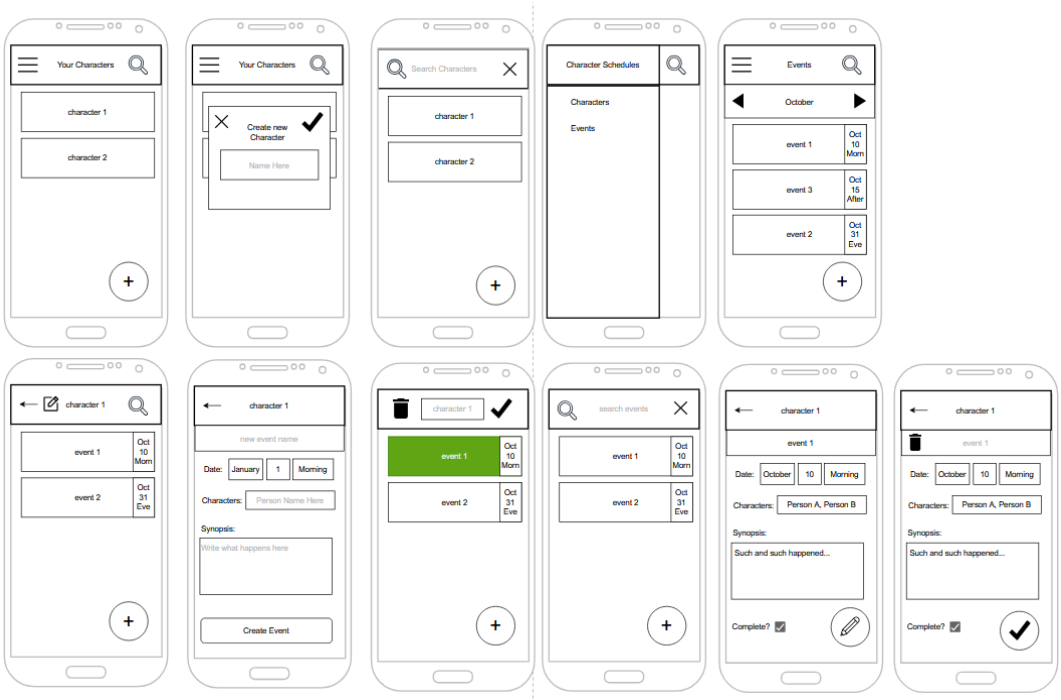
- What are the required screens of the app?

Screen: Character List Screen
Fragments: Character List Fragment, Month List Fragment, Event View Fragment
Screen: Character Screen
Fragments: Event List Fragment, Event View Fragment

- What data should be kept in the database?

Characters: Name, Event (List)
Event: Character, Name, Date (Month, Day, Time), Characters, Synopsis, Completed (T/F)
(Note: Character refers to the character the event is tied to. Characters refer to the other characters that are associated with the event. Does not need to be a created Character)

- What should the screens look like?



(Note: Event screen will show "character name" or "month name" at the top depending on where it was navigated from)

Interview:

1) *The app is well-designed and intuitive to use*
2) *The “Complete?” field on the event is a little unclear. Could be replaced by “Scene Complete” or something similar*
3) *The “Morning/Afternoon/Evening” field could be replaced with an image for a more streamlined appearance*
    a) *This may not be possible, as there are many possible inputs for this field (Early Morning/Morning/Early Afternon/etc)*

CRUD Operations and App Functionality

- The theme of the project is up to you! Choose anything that interests you.
- The general look and feel of the app should reflect the design diagrams.
- The project must use the Room library to create an on-device database.
- The data must exhibit at least 1 one-to-many relationship.
- Your app must facilitate all CRUD Operations: Create, Read, Update, Delete. In other words, your app must allow the user to add new data, edit or delete existing data, and see the data.
- You must construct your app and deliver it (as normal) to WebCourses. In addition, submit a brief description of the steps to be used to completely test your app.

Creating a new character:

1) *Press the (+) button. Type in Character name and click yes*
2) *Character has been created*

Retrieve a character:

1) *Press the magnifying glass in the toolbar. Type in character name (OPTIONAL)*
2) *Click on character to view their events*

Update:

1) *Click on pencil floating action button to change character name*
2) *Click the checkmark when complete*

Delete:

1) *Click on pencil*
2) *Click on the trash bin icon*
3) *Click "Yes" on the prompt*

Creating a new event:

1) *Click on character that you want to tie your event to*
2) *Click the plus button*
3) *Edit fields*
4) *Click checkmark when done*

Retrieve an event:

1) *METHOD 1:*
    a) *Hit the navigation drawer, go to "Events"*
    b) *Click on the arrow keys to navigate to the desired month*
    c) *Press the magnifying class on the toolbar to search per month (OPTIONAL)*
    d) *Click on the event to view its details (character name will be displayed on the top)*
2) *METHOD 2:*
    a) *Click on a character name*
    b) *Scroll through their events*
    c) *Press the magnifying glass on the toolbar to search the characters' events (OPTIONAL)*
    d) *Click on an event to view it*

Update:

1) *Click to view the event*

2) *Click on the pencil floating action button to change its details*
3) *Edit details*
4) *Click the checkmark when complete*

Delete:

1) *On the event detail screen, click the pencil floating action button*
2) *Click on the trash bin icon that appears next to the event name*
3) *Click yes*

Testing and Reflection

- You must test your app with at least 3 individuals and get their feedback using metrics (you can use the SUS document from the UX lecture in class). You will deliver the findings from your testing with the final app.
- Let the 3 individuals try your app on the emulator.
- Collect their feedback using the SUS document from class.

- Submit a writeup about their feedback in addition to the raw data.
    - What did the users like about the app?

  Users were split on their opinions about the UI and design, but most found it well-made (one found it confusing, but I also didn't give them instructions on how to use it… oops). UI and event creation were the aspects users seemed to talk about positively.

  Most seem confused about what the app was actually for- but I attribute that to the function being rather niche. Without a detailed list of how to do each function, users were left confused as well (one suggested I add a tutorial, which seems like a good idea actually! Just a little popup around the floating ation button that tells the user what it does)

    - What can be improved about the app, if you had more time?

  I actually have a list!
    - *Run database operations on a different thread*
    - *Use animations*
    - *Fix a bug where users can't search when the activity first starts*
        - *(I know the reason, I just don't know how to fix it just yet. I think it's because everything runs on the main thread, and I can't set the search listeners to be set after the menu is drawn. But once you navigate through different fragments then the menu will be drawn and everything works properly)*
    - *Learn how to use xml formatting better (the edit screen looks ROUGH)*
    - *Learn how to use themes (for a more cohesive colour scheme)*

    - *Replace the edit icon with a triple button popup, which will let the user delete or edit in one place (this should be an easy fix though, and probably a lot easier to code too lol)*
    - *Implement a proper calendar setup (to support different years)*
    - *Add a notes area for future / scenes to consider*
    - *Support for logins and cloud storage*

- Submit this document as a PDF on WebCourses.

Submitting Your Assignment

1. There are 3 deliverables for this assignment (see above). Make sure to submit on time for each.
    a. The Interview and Design Documentation is due before the app.
    b. The app and the testing & reflection documentation is due at the same time. You can place the reflection documentation in the .zip with the app project folder.
2. In WebCourses, navigate to the appropriate assignment and use File Upload to submit your files.
3. Your app must use Java (not Kotlin).