# Final Project Report

*Comments about what should be in your report can be found in italics.* ***You must remove the comments and replace them with your own words****.*

## Method

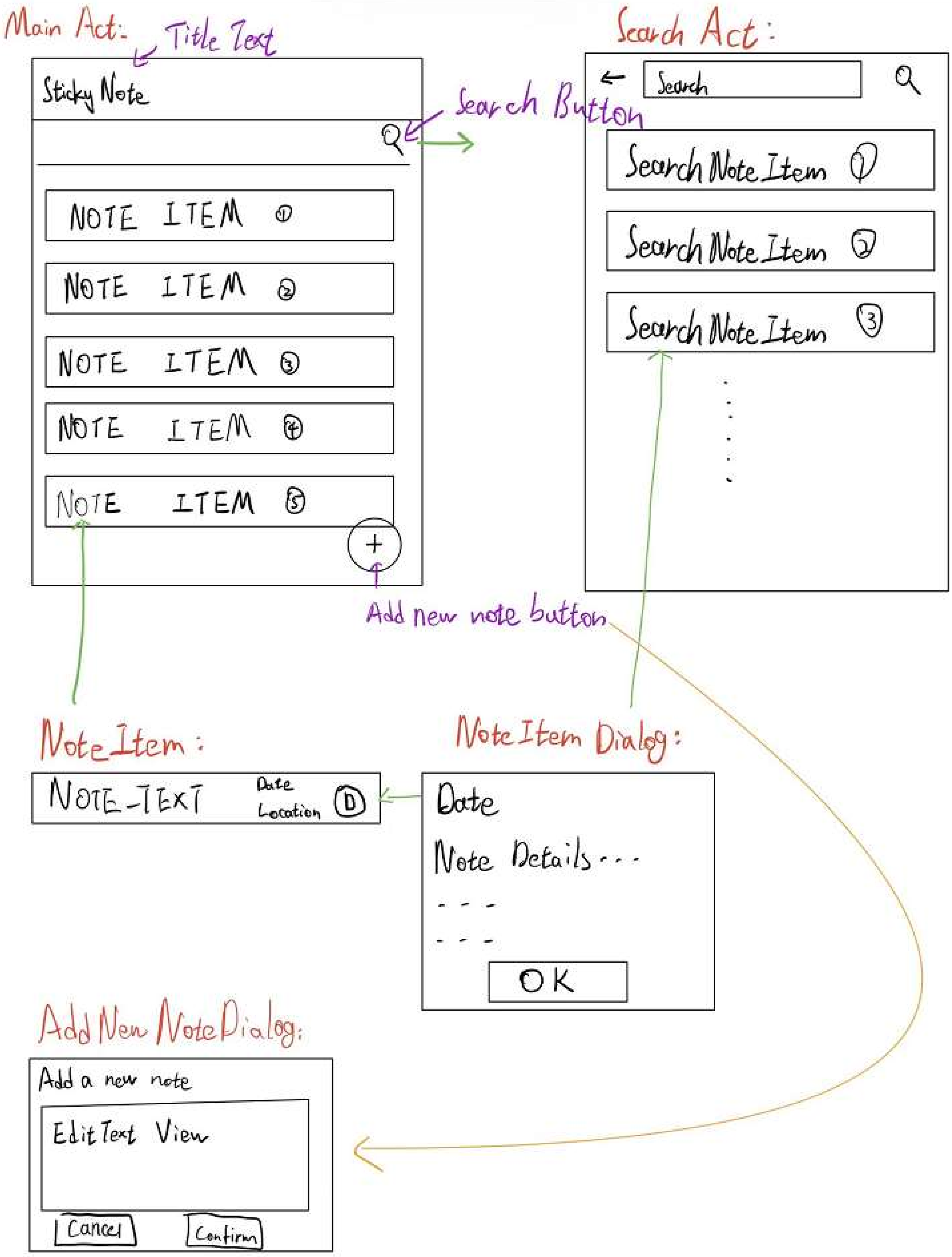
**Application Description**

*Write two sentences describing the purpose of your app. This can be the same text you use in your [design document](#) under* `Functional Overview`

Writing memos on a sticky note using pen and paper is a great way to memorize important things. However, it is inconvenient for visually impaired people to complete this task. Current note-taking apps don't have talkback functionalities. Therefore, my Sticky Note App can solve this problem by providing visually impaired people with an accessible and location-based note-taking experience.

*Review the low fidelity prototype sketches, your layout wireframe, and interactor hierarchy from your [design document](#). What changed from your original concept to your final implementation? Why did you make those changes from your original design vision?*

Here's my prototype sketches:



Changes:

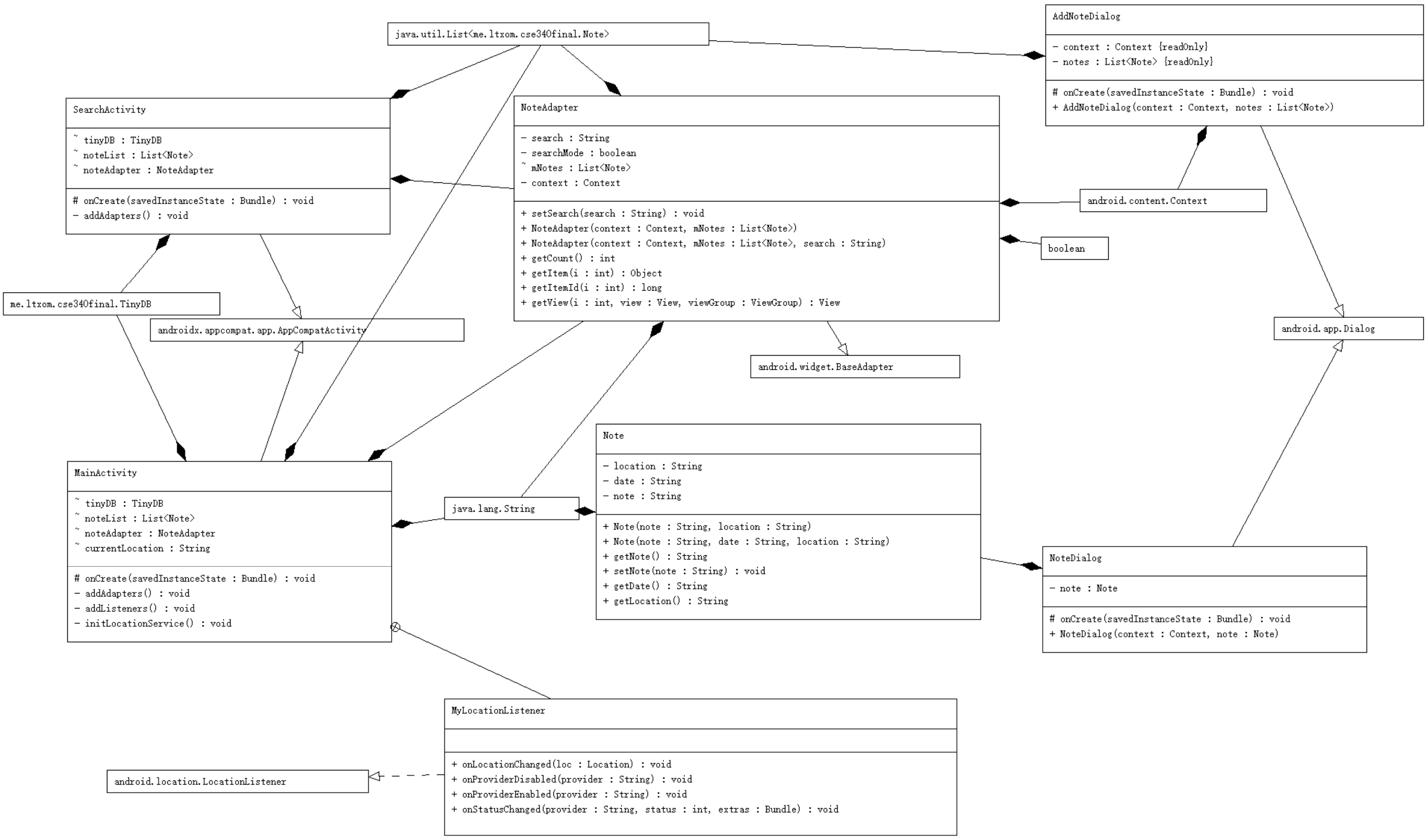
1. Remove Delete Button for Note Item in Search Activity

In my original sketch, I planned to make Main Activity and Search Activity have identical lists of Note Items. However, after implementation, when I used the talkback to navigate through my search results, I found the deletion buttons distracted attention from the user (talkback in delete button read all information of note content, date, location again). Users using the searching mode care more about the items of results. Therefore, I removed it for better readability.

2. Add TextView of Location Note Item Dialog

In my sketch, it only shows information of location in the Note Item, which only appears in the main activity or search activity. When the user clicks in each Note Item for the detailed information of the note, it pops up the Note Item Dialog. I found Note Item Dialog doesn’t contain this important location information, which can cause inconvenience for users to be aware of where they took the notes. Therefore, I added a TextView of the location under the Date TextView.

*Finally, include a simplified UML diagram that described the overall structure of the classes and interfaces you created. You do not need to include all of your methods and variables, but do include ones that are key to demonstrating your understanding of the work you did for this project. Your UML diagram may be hand drawn, or you can use various on-line tools (including the tool we use to generate our Mermaid UML diagrams), or you could find a plug in to add to Android Studio.*


java.util.List<me.ltxom.cse340final.Note>
AddNoteDialog
- context : Context {readOnly}
- notes : List<Note> {readOnly}
# onCreate(savedInstanceState : Bundle) : void
+ AddNoteDialog(context : Context, notes : List<Note>)
SearchActivity
~ tinyDB : TinyDB
~ noteList : List<Note>
~ noteAdapter : NoteAdapter
# onCreate(savedInstanceState : Bundle) : void
- addAdapters() : void
NoteAdapter
- search : String
- searchMode : boolean
~ mNotes : List<Note>
- context : Context
+ setSearch(search : String) : void
+ NoteAdapter(context : Context, mNotes : List<Note>)
+ NoteAdapter(context : Context, mNotes : List<Note>, search : String)
+ getCount() : int
+ getItem(i : int) : Object
+ getItemId(i : int) : long
+ getView(i : int, view : View, viewGroup : ViewGroup) : View
android.content.Context
boolean
me.ltxom.cse340final.TinyDB
androidx.appcompat.app.AppCompatActivity
android.app.Dialog
android.widget.BaseAdapter
Note
- location : String
- date : String
- note : String
+ Note(note : String, location : String)
+ Note(note : String, date : String, location : String)
+ getNote() : String
+ setNote(note : String) : void
+ getDate() : String
+ getLocation() : String
MainActivity
~ tinyDB : TinyDB
~ noteList : List<Note>
~ noteAdapter : NoteAdapter
~ currentLocation : String
# onCreate(savedInstanceState : Bundle) : void
- addAdapters() : void
- addListeners() : void
- initLocationService() : void
java.lang.String
NoteDialog
- note : Note
# onCreate(savedInstanceState : Bundle) : void
+ NoteDialog(context : Context, note : Note)
MyLocationListener
+ onLocationChanged(loc : Location) : void
+ onProviderDisabled(provider : String) : void
+ onProviderEnabled(provider : String) : void
+ onStatusChanged(provider : String, status : int, extras : Bundle) : void
android.location.LocationListener

## Tasks

*List out the task scenarios you asked your users to perform in your SUS test.*

1. Test Location with Emulator or Real Device
   In this task, please test the location sensing of StickyNote App. When you open the StickyNote App as first time, it will pop up a dialog to request you location permissions. Please try to accept it or deny it. If you deny it, please try to add a new note by clicking the plus button in the right bottom corner of the main screen. The note item should show you “No GPS” as the location information.

If you accept the location permissions, please try to move around (if using a real device) or adjust the location (if using an emulator), then add a new note and see if the location information is correct.
If you are using an emulator, you can click the hamburger menu in the sidebar. In the location section, you can drag the blue dot to a different location and click set location to simulate moving around.

2. Test Data Storage
   On the main screen, once you create a note, there is a delete button. Before deleting it, try to close (even reboot the device) and re-open the app and see if the notes you created are still there. Then, please try to delete some notes, and re-open the app to see if the modifications are still effective.

3. Test Searching Functionality
   Please add some notes before performing this task. On the main screen, in the top right corner, there is a search button. You can click it to navigate to the search screen. Please try to enter some keywords based on note content or note location. You can also try to use lower or upper cases. Please see if you can find expected results.

4. Accessibility/Talkback Test
   Please turn on the Talkback of Android. If you need assistance to do it, please contact me. Please also make sure to turn on the volume. Then, go back to the App and try to perform adding/deleting/searching notes without seeing the screen. The TalkBack will give you precise and accurate information about each

action/focus you are playing. It only requires you to do three types of interactions: swiping up, swiping down, or double-clicking.

**Setting**

*What device was used? Was it an emulator? Did they use a mouse or a finger? Where did the experiment take place?*

Two testers used an emulator, one with a mouse and another with a touchpad. The experiment with them took place via Zoom.

One tester used a real device (Oneplus 9 Pro) in-person with me. He used fingers to test. While testing, we moved from Seattle to Bellevue to test the location sensing.

**Participants:**

*Describe your participants (without identifying them). How were they recruited? How many were there? How were they they consented? You should also add some optional information such as: What was there average age? What genders were present? How experienced were they with android?*

One participant is one of my close friends. I recruited him via direct messages. I found another participant, a classmate in 340, via Ed. The last participant is one of the TA in 340, recruited via email. They have consented via emails and SUS reporting form in Google form. The genders are two males and one female. All of them are experienced with Android but only two are experienced with TalkBack tasks. Their average age is 22.

**Data Collected:**

*In your own words, describe the SUS testing questionnaire.*

The SUS testing questionnaire contains 10 questions, each question has a range from 1 to 5, where 1 means least agreed and 5 means most agreed. The higher score (more agreeable) for the odd-numbered questions and lower score (less agreeable) for the even-numbered questions indicate a better user experience.

The average rate of odd-numbered questions collected from three testers is 4.4 (agree to strongly agree). The average rate of even-numbered questions 1.8 (disagree to strongly disagree).

## Results

### SUS Scores

*List the overall SUS score for each of your testers, as well as the average from all tests. How does your average SUS score compare to the US average of 68? Using this chart, what grade would you give your app's interface?*

Tester 1: ((3+4+4+4+4-5)+(25-2-2-3-2-2))*2.5=70

Tester 2: ((5+5+4+5+5-5)+(25-1-2-2-1-2))*2.5=90

Tester 3: ((5+4+4+5+5-5)+(25-2-1-2-2-1))*2.5=87.5

Average: 82.5

The average of my SUS score compared to the US average of 68 ranks in about 90% percentile. The score fits in the A or A- grade.

*Recall that items 4 and 10 provide the learnability dimension and the other 8 items provide the usability dimension. What were the values on these subscores?*

The average subscore of learnability dimension is (1+2+1+2+2+2)/6=1.5 (disagree to strongly disagree), which indicates users do not require difficulty in learning using StickyNote App.

# Conclusions and Future Work

- *Draw and describe your overall conclusions from this user testing. How did your app fare relative to the average US score of 68.*

  The average of my SUS score compared to the US average of 68 ranks in about 90% percentile, which represents my App is potentially ranked higher than 90% of other results in US average of 68**.**

- *What aspects of your app do you feel contributed to this score?* The convenience of my app contributes a lot to this score. In SUS testing responses, two testers responded strongly agree (5) for "I think that I would like to use this app frequently." In talking with one tester, he commented my implementation of data storage is great since he can really install this App in his cell phone to take daily notes without worrying about losing data.

Also, the potential this App can help many people also contributed to this score. In responses, two testers said strongly agree (5) and one said agree (4) for the question “I would imagine that most people would learn to use this app very quickly.” Since I designed this App with full consideration of accessibility, different people can use it easily even if they use screen readers.

- *How did your app fare on the learnability vs usability dimensions? What do you think contributed to those scores?*

The average subscore of the learnability dimension is 1.5 (disagree to strongly disagree), which indicates that it is not difficult for users to learn using StickyNote App. I used icons that are large enough and with contrastive colors. Also, these icons are common shapes (plus icon for adding note action, magnifier icon for searching action) which are easy for users to learn. All three testers responded Agree (4) for “I found the various functions in this app were well integrated.” The design of the adapter with the MVC framework allows everything integrated together smoothly. Plus the above discussion of its convenience and accessibility of it, these factors prove my App has strong usability to contribute to these scores.

- *Describe two areas of future work for your app, including how you could increase the accessibility and usability of this app*

One feature that I want to make in the future is a location-based or time-based notification system. In each note item, the user can set a further time or a location, so in the set times or when the user enters the specified locations, the device will send notifications to the user with the content of the notes. In this case, the sticky note app will become more usable since it can remind users of the things to do at a certain time or location. Additionally, I wonder if I can make the notification read around when the user turns on TalkBack. In this way, the app can be more accessible for visually impaired people, because they don't have to open the phone to swipe to the detail of notification.

Another feature that I want to add is the editing feature. Currently, once the note is added, the user can only view/search/delete it. However, it is easy to make mistakes when adding a note. If the user added a very long note and found a typo, it would be frustrating for them to delete it and start over. Therefore, by allowing note editing, the using experience of StickyNote App will be greatly improved.

# Acknowledgements

*Cite anything (website or other resource) or anyone that assisted you in creating your solution to this assignment. Remember to include all online resources (other than information learned in lecture or section and android documentation) such as Stack Overflow, other blogs, students in this class, or TAs and instructors who helped you during OH*

LocationListener: http://rdcworld-android.blogspot.com/2012/01/get-current-location-coordinates-city.html
Permission Dialog:
https://developer.android.com/training/location/permissions

Adapter: https://gitee.com/happyanimee/simple-bookkeeping/blob/master/app/src/main/java/com/hui/tally/adapter/AccountAdapter.java

TinyDB: https://github.com/kcochibili/TinyDB--Android-Shared-Preferences-Turbo

# Final reflection

*Thinking about CSE 340 as a whole:*

- *What do you feel was the most valuable thing you learned in CSE 340 that will help you beyond this class, and why?*

I think one of most important and valuable things that I learned in 340 is the way to make accessible applications. Before learning this course, I was a front-end developer that familiar with HTML/CSS/JS, but I feel my front-end works have flaws because they just look weird compared to experts' works. Before taking 340, I didn't expect this class can help me with that. However, while learning the accessible section, I learned making a good application is not only writing good code. The most crucial aspect is to consider how a user can interact well with the application. Especially when I implemented the Accessibility assignment, I realized I need to improve the App from the user's perspectives. I realized what I missed in my front-end works is accessibility or consideration for the users. Therefore, 340 really helped me more than writing codes for Android, but also provided me an important way of thinking with consideration and accessibility, which can help me in an even broader aspect in my front-end developing career.

- *If you could go back and give yourself 2-3 pieces of advice at the beginning of the class, what would you say and why? (Alternatively what 2-3 pieces of advice would you give to future students who take CSE 340 and why?)*

1. Plan Ahead

   This course is content-rich to prepare you to grasp interactive programming. Make sure to preview the content of each week and study ahead, so that you can prepare questions in lectures or OHs.

2. Manage Time Wisely

   Although there are five late days through out the quarter to use, it is better to plan to complete every assignment without using it. There are assignments or checkpoints and practice quizzes due each week. Please make sure to start doing them earlier, especially because there are some tricky bugs that require a great deal of time dedication.

3. Check Canvas/Gradescope/Ed frequently

   There are many platforms (Canvas/Gradescope/Ed) that used in this class. Make sure to check all of them frequently to avoid missing any submission.