



DeepSee Launcher

Standup Documentation, Sprint 3.
Deliverable 4.
CSCC01 Group 6, Tutorial 2.

Day 1: Task distribution

Mohammed

- Designing the Messages application
- Designing the Message widget

Muhannad

- Weather widget

Akshath

- Implement additional settings
- Manage Translation/Screenreader features with Daniyal

Alex

- Algorithm/data structures for app suggestion algorithm

Daniyal

- Home screen redesign (needs to accommodate app suggestions)
- Storage/Memoization Management for App Drawer & AI
- Manage Translation/Screenreader features with Akshath

Angel

- Emergency contacts creation

Fahim

- App drawer needs to be optimized and redesigned to be more user friendly

Misc:

Decided on standup every other day for productivity purposes

Day 2:

Task progress: App shortcuts

- Created a uniform app shortcut class that can be used to display apps within many other features such as:
 - App drawer
 - Persistent app bar
 - Auto-suggest bar

Task progress: Auto-suggest algorithm

- Suppose we monitor the top n most used apps
- $n \times n$ matrix for likelihood of going from one app to another
- $n \times 24$ matrix for likelihood of using an app at a certain hour of day
- Multiply the relevant indices from both matrices to obtain likelihood of each app being used → then, display the top 3 most likely to be required at current time
- Also need an array for storing the total number of times ANY app is used
 - This will be useful for determining which apps to put in the top n most used, and potentially useful at other steps in the algorithm

Issue: Managing storage

- Need storage for the algorithm data
- Need storage for launcher settings
- Need storage for emergency contacts

Individual Progress:

Daniyal: Managed to make storageManager class, needs debugging & things must be Serializable. Will test and work on shortcuts before next standup

Mohammed: Designed SMS Message data structure and frontend ideas

Muhannad: Setup classes to store and retrieve weather information

Angel: Design Emergency Contact view page

Fahim: Fixing bugs in shortcuts branch where it's not visible

Akshath: Designed layout for SettingsActivity and made ListView clickable

Alex: Planned how to implement the Application Prediction algorithm

Day 3:

Task progress: Managing storage

- Storage manager classes created

Task Completed: Weather widget

- Implement the weather widget
 - API to receive data
 - UI to display data

Task Progress: App-Suggestion Algorithm

- Discuss how we can collect data

Task Progress: Settings

- Turn the setting selection pop-up from an alert into a fragment

Individual Progress:

Daniyal: Tested StorageManager class outside application. Planned out StorageContainer and other relevant classes to allow for Shortcuts. Will implement Adapters by next standup.

Mohammed: Managed to fetch all SMS Messages and store them using the data structure created and got frontend to partially display some Messages

Muhannad: Connected the weather classes I made to an online weather API to retrieve real time weather updates.

Angel: Design Emergency Contact adding contact page and set up recyclerviews and adapter

Fahim: Implemented app uninstall-mode

Akshath: Worked on changing alertDialogs in settings to Fragments

Alex: Collecting and managing data for algorithm

Day 4:

Task Completed: Managing storage

- Errors fixed, merged into Dev branch

Task Progress: Settings

Individual Progress:

Daniyal: Finished Handlers and made FragmentContainer for Shortcuts, working on Context Menu additions for Android. Handing off Shortcuts to Fahim, who has a better understanding of RecyclerView holder classes. Beginning work on TranslationActivity and ScreenReaderActivity using Alex's TTS engine.

Mohammed: Created a Minor Filtering system for SMS Messages and finished frontend to display all Messages

Muhannad: Fixed bugs with retrieving weather data and cleaned up the UI of the weather widget.

Angel: Set up fetching contacts from account

Fahim: Bug fixing app-uninstall mode (Apps Adapter isn't reacting to app removal correctly)

Akshath: Abandoned Fragmentation for settings and focused on persistent notification for ease of access, since setting has to be an activity to be accessed through a notification

Day 5:

Task progress: App-Suggestion Algorithm

- Custom data structure SortedIncrementList implemented for the purpose of keeping track of app usage
 - $O(1)$ app counter incrementation
 - $O(1)$ new app addition
 - $O(n)$ returning the top n most used apps

Individual Progress:

Daniyal: Completed and Tested Context Menu visibility. Implemented Persistent Shortcut for Settings, since Overlays have security issues. Will debug PersistentNotification shortcut and write a temporary Overlay generator service for QuickSettings and TranslationActivity.

Muhannad: Helped Mohammed setup the backend of the messaging widget which involved retrieving the parsing the sms messages stored on the phone.

Mohammed: Fixed Messages backend bugs and minor frontend edits

Angel: display all contacts in recyclerview of contacts to add to emergency contacts

Fahim: Integrating StorageManager, App Recommendation algorithm, and App Drawer

Akshath: Started design on translation activity, and created QuickSettings to be accessed through the persistent notification.

Alex: Implemented data structure for keeping track of app usage

Day 6:

Task redesigned: Storage manager

- Changed to an implementation using the serializable interface

Task completed: App-Suggestion Algorithm

- AlgoStruct uses multiple SortedIncrementList objects to store all the required data
 - App usages at certain hour of day
 - App usage in sequence after any particular other app
- $O(n)$ calculating the top n most relevant apps for the current situation

Individual Progress:

Daniyal: Completed TranslationActivity with Akshath. Found a new approach for Shortcuts since ShortcutsContainerFragment has been very annoying. Beginning work on a new approach for homescreen internal activities shortcuts. Discarding ShortcutsContainerFragment after discussion with Fahim.

Mohammed: Started working on Messages widgets and implemented frontend completely

Muhannad: Added message filtering and fixed bugs related to retrieving messages.

Angel: Populate recyclerview for added contacts with contacts that are added from to add list

Fahim: Bug fixing storage-manager integration, and implemented background synchronizer and broadcast-receiver for package changes

Akshath: Finished majority of work on TranslationActivity with Daniyal, as well as all the setting options for both QuickSettingsActivity and SettingsActivity. Also, I merged TranslationActivity and PersistentNotification into feat/settings branch, since all three branches were dependent on each other to some capacity.

Alex: Finishing up the Application Prediction algorithm

Day 7:

Task completed: UI overhaul

- Redesigned our UI to accommodate all the new features that have been added throughout the sprint
 - Determining which features should be included on the home screen
 - Creating a common theme among the features

Individual Progress:

Daniyal: Completed Homescreen activities shortcuts. Merges broke Context menus, investigating cause (Likely Package visibility or build version). Working on fixing this for ScreenReaderActivity and TranslationActivity.

Mohammed: Finished the widgets and its backend and contributed to frontend visual overhaul when it comes to the shortcuts tab

Muhannad: Worked with fahim to merge all branches together and fixed any bugs that happened as a result of that. Also worked on cleaning up the overall UI of the app.

Angel: display the added contacts on viewing page with green call button(button does not call in this prototype), added default emergency contact for police

Fahim: Implemented recommended app section and merged several feature branches into dev

Akshath: Implemented an extra feature (user input) on Translation activity, as well as merging feat/settings branch into dev. Also fixed any bugs in MainActivity, TranslationActivity, SettingsActivity, and QuickSettingsActivity.

Alex: Integrated the algorithm into the app