**Autumn Semester 2022/23**

**Mobile Device Programming (COMP3040)**

**Coursework 2**

|  |  |
| --- | --- |
| Name | Tan Zhun Xian |
| Student ID | 20313854 |
| Lecturer | Doreen Sim Ying Ying |
| Date | 12/12/2022 |

Contents

[Experience in Developing App 3](#_Toc121694640)

[Changes from Coursework One 5](#_Toc121694641)

[Minimum API Level 5](#_Toc121694642)

[App Permissions 5](#_Toc121694643)

[Manifest Permissions 5](#_Toc121694644)

[Post Notification Permission 5](#_Toc121694645)

[Read External Storage Permission 5](#_Toc121694646)

[Read Media Audio Permission 5](#_Toc121694647)

[Write External Storage Permission 6](#_Toc121694648)

[Sleep Reminder 6](#_Toc121694649)

[Broadcast Receiver 6](#_Toc121694650)

[Sleep Notification 6](#_Toc121694651)

[Edit Text Errors 6](#_Toc121694652)

[Empty field error 6](#_Toc121694653)

[Name taken error 6](#_Toc121694654)

[Password mismatch error 7](#_Toc121694655)

[Same password error 7](#_Toc121694656)

[Different password error 7](#_Toc121694657)

[Non-positive number error 7](#_Toc121694658)

[Spinner Button Removal 7](#_Toc121694659)

[List Spinners 7](#_Toc121694660)

[Chart Spinner 7](#_Toc121694661)

[Hidden Buttons 8](#_Toc121694662)

[Music List Items Hidden 8](#_Toc121694663)

[Music List Items Shown 8](#_Toc121694664)

[Sport Type Items Hidden 8](#_Toc121694665)

[Sport Type Items Shown 8](#_Toc121694666)

[Sleep Data Recycler List Items Hidden 8](#_Toc121694667)

[Sleep Data Recycler List Items Shown 8](#_Toc121694668)

[Music Playlist Expandable List Item Hidden 9](#_Toc121694669)

[Music Playlist Expandable List Item Shown 9](#_Toc121694670)

[Sport Data Expandable List Items Hidden 9](#_Toc121694671)

[Sport Data Expandable List Items Shown 9](#_Toc121694672)

[Sport Type Statistics Recycler List Items 9](#_Toc121694673)

[Sport Data Activity Hidden Menu 9](#_Toc121694674)

[Sport Type Dialog to Activity 10](#_Toc121694675)

[Save and Load Functionality Replaced 10](#_Toc121694676)

# Experience in Developing App

In the initial stages of app development, I had to decide on a suitable design for my app which matches the topic of Health and/or Mental Health and Well-Being during the post-pandemic era. My initial choices were Sleep, Sport, Diet and Mental Health. In the end, I decided to focus on Sleep and Sport. To make my app more interesting, I also decided to add a music player to the app so that my app can stand out.

Early on, I focused on making a database for the app because I wanted each user to have their own personal data and settings instead of sharing the same data for the entire app. In the end, I decided to use Room database as a local database as my app was no designed to have an online database. Tables were created for users, sleep data, song data, playlist data sport data and sport types. An issue I faced here was how to accurately represent playlist data and sport data. A playlist can have many different songs while a song can belong to many different playlists. The same can be said for sport data and sport type. To resolve the many-to-many relationship, I created intermediate tables to link the two tables together. This brought the total number of database tables to 8.

With my app now being able to accommodate different user data, I now had to give a way for the app to switch between users. So, a login page was created for users to login their accounts or as guests. An account page was also created for users to create new account, change username, change password, and delete account. Text fields were also added with text watchers and focus listeners to dynamically show errors if input is incorrect.

For displaying data in the app, three types of lists were used which are list view, recycler list view and expandable list view. Song and sport type data uses list view as their data is simple and can be shown in a single line. Sleep data and sport statistics uses recycler list view as their data is more complex and require more lines to display. Finally, playlist and sport data use expandable list view due to their variable child count which can change dynamically.

To make the app seem more responsive, live data was used to observe the changes in the data so that the lists can update themselves if any change in data occurs. The hardest part about implementing live data is to add null checks. As an example, deleting a song causes all instances of the song in all playlists to also be deleted. Sometimes, the song is deleted yet is not refreshed properly in the playlist and exceptions can occur when the playlist tries to display a null song. So, null checks are added to filter out the null data before data is displayed.

Sorting functionality was also added to the app. Spinners were used to hold the sort options. User can choose the data to be sorted and then the sort order. The confirmation button was removed to make the app feel more responsive as data is immediately sorted when either spinner was used. A search bar was planned but ultimately scrapped due to time constraints.

To make the editing and deleting data more seamless and user friendly, hidden buttons were incorporated into the list items. Users only need to long-click on the list item and buttons should appear. The buttons are the edit and delete button. Clicking them allows the user to edit and delete the data much more easily.

A sleep and sport bar chart were also implemented in the app. Users can choose the data they want displayed on the charts. The charts are sorted ascending by date by default. Users can use the chart to discern any trends in their daily habits like sleep and wake times or sleep and sport durations. The chart used was created by PhilJay and was imported from Github. This is because Android Studio does not have native bar chart functionality.

A sleep and sport calendar were also added to the app. A pop-up screen to display sleep/sport data was planned on date change. However, this idea was also scrapped due to time constraints. In the end two buttons were used to check if the selected data has sleep/sport data. If data is absent, the add button is enabled else the more info button is enabled.

A simple music player was also created for the app. Users can play the entire song list by clicking on a song in the song list fragment. If users desire to play songs in a specific playlist, they can click on the song under its playlist. Songs will loop through their playlist forever. The music player has simple functionalities such as play/pause, next song, and previous song.

Statistics for music, sleep and sport were also calculated. For music data, the playlist and song statistics were calculated while sleep statistics were calculated for the sleep data. The calculations for sport data were a little different. Instead of using the date to calculate the statistics, the sport type was used. Each sport type has its own statistics if it is performed at least once by the user. If no instances of the sport were recorded, then it is not shown.

Save and load functionality was also originally planned for the app. It was supposed to allow users to rollback any changes to their data if they were unsatisfied with the changes. However, due to the complexity and time constraints, it was ultimately removed from the app. Instead, it was replaced by a logs list where every notable action in the app was recorded and displayed. Only the actions in the current session were shown while the history of all actions are saved into a log file. Users can download the log file by clicking the download button provided.

A sleep reminder notification was also added to the app. The app will set a notification to notify the user to sleep at 10pm when the login page loads. If the user does not use the app for an entire day, then no notification is set for that day.

A notably challenging aspect of developing the app was choosing the correct permissions to include in the app. Different API levels have different permissions and coding all the different cases was quite interesting if not a bit frustrating.

In conclusion, developing the app is not a hard task. The main difficulty comes from the fact that I must learn how to code all the different functionalities planned. By promising too much in coursework one, I inadvertently made developing the app more tedious and time consuming. The main constraint in app development was time as I had to also balance my time between the other different coursework. So, some functionalities had to be reduced in complexity or removed from the app entirely to meet the deadline.

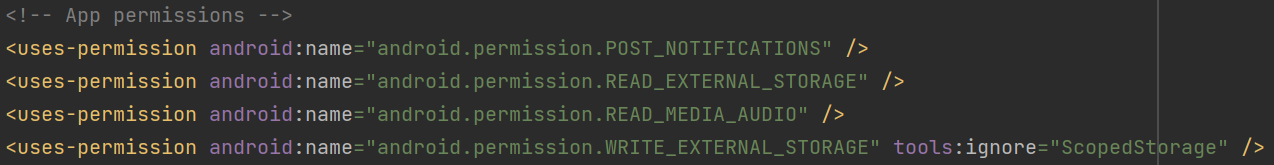
# Changes from Coursework One

## Minimum API Level

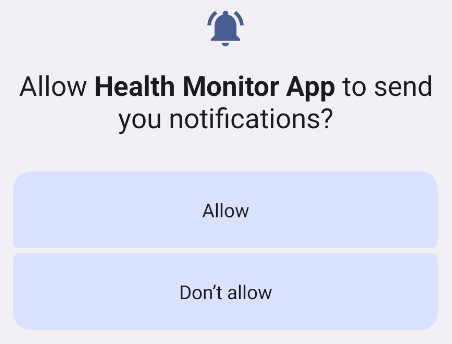
The minimum API level of the app had to be increased from API 21 to API 26. This is because some functions in the app cannot function on lower API levels. Even with API desugaring, some packages were still not able to be used such as java.nio.file as they are not supported.

## App Permissions

### Manifest Permissions

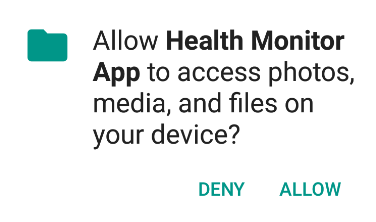


### Post Notification Permission



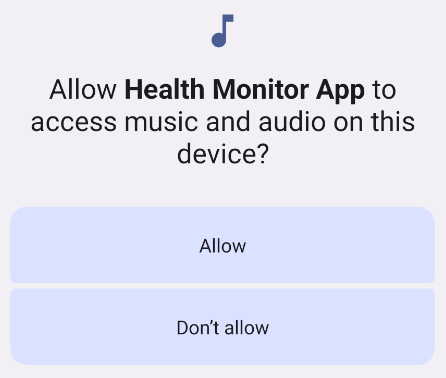
Post Notification Permission is needed for API level greater than or equal to 33. It is triggered on the login page if user does not have post notification permission.

### Read External Storage Permission



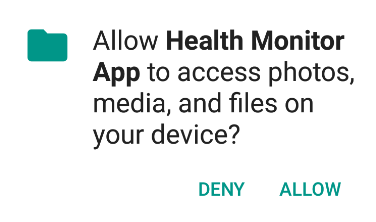
Read External Storage Permission is needed for API level less than 33. It is triggered if user wants to copy song to music folder yet does not have read external storage permission.

### Read Media Audio Permission



Read Media Audio Permission is needed for API level greater than or equal to 33. It is triggered if user wants to copy song to music folder yet does not have read media audio permission.

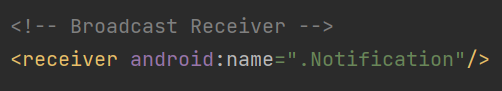
### Write External Storage Permission



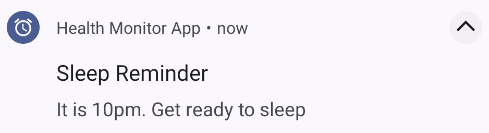
Write External Storage Permission is needed for API level less than 30. It is triggered if user wants to download logs file yet does not have write external storage permission.

## Sleep Reminder

### Broadcast Receiver



### Sleep Notification



This is the sleep notification set every time user reaches the login page. It is triggered every 10pm and reminds user to sleep early every night. If users do not use the app for an entire day, then the notification for that day is not set and users will not receive a sleep reminder.

## Edit Text Errors

Edit texts across the app were added with text watchers and on focus listeners to display errors to make the app more interactive. By reworking edit text, check buttons are not needed to check name availability as it is now done automatically. Some examples are given below.

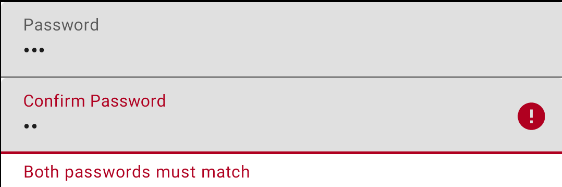
### Empty field error



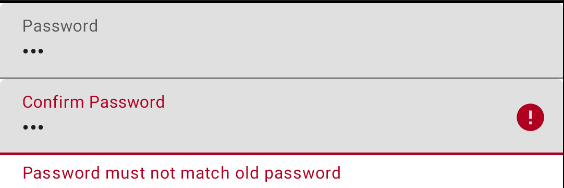
### Name taken error



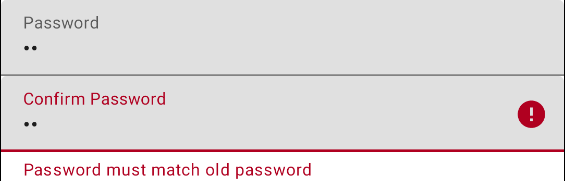
### Password mismatch error



### Same password error



### Different password error



### Non-positive number error



## Spinner Button Removal

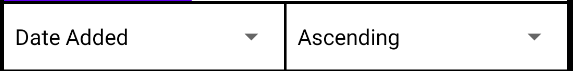
Spinners in the app were originally paired with a button to lock in the choices. However, it was decided that it was too bothersome, and it was removed. Instead, the spinner choice is now processed every time there is a change in the spinner data.

### List Spinners

List spinners are used to choose how the list data is to be sorted. The first spinner chooses the data to be sorted. The second spinner chooses the sort order of the data chosen.



Old spinners design



New spinners design

### Chart Spinner

Chart spinner is used to choose the data to be displayed in the chart.



Old spinner design

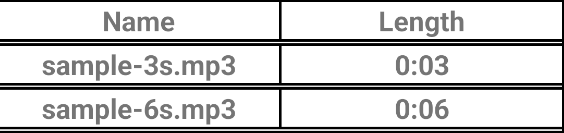


New spinner design

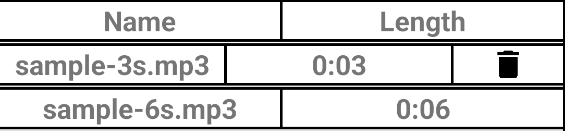
## Hidden Buttons

The buttons in all the list views are now hidden by default to give the app a cleaner look. To display the hidden buttons, users must long click on the list items. To hide the displayed buttons, users must long click on the list items again. Most list items have 2 hidden buttons, the edit and delete button. However, music list items only have the delete button while sport type statistics recycler list items have no hidden buttons.

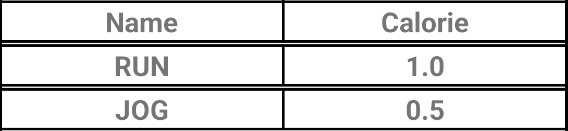
### Music List Items Hidden



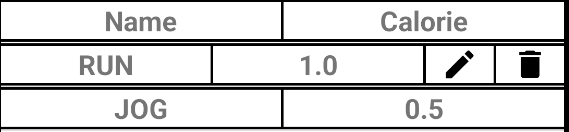
### Music List Items Shown



### Sport Type Items Hidden



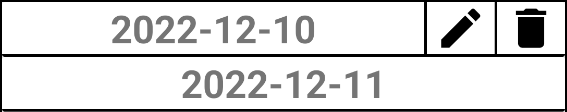
### Sport Type Items Shown



### Sleep Data Recycler List Items Hidden



### Sleep Data Recycler List Items Shown



### Music Playlist Expandable List Item Hidden



### Music Playlist Expandable List Item Shown



### Sport Data Expandable List Items Hidden



### Sport Data Expandable List Items Shown

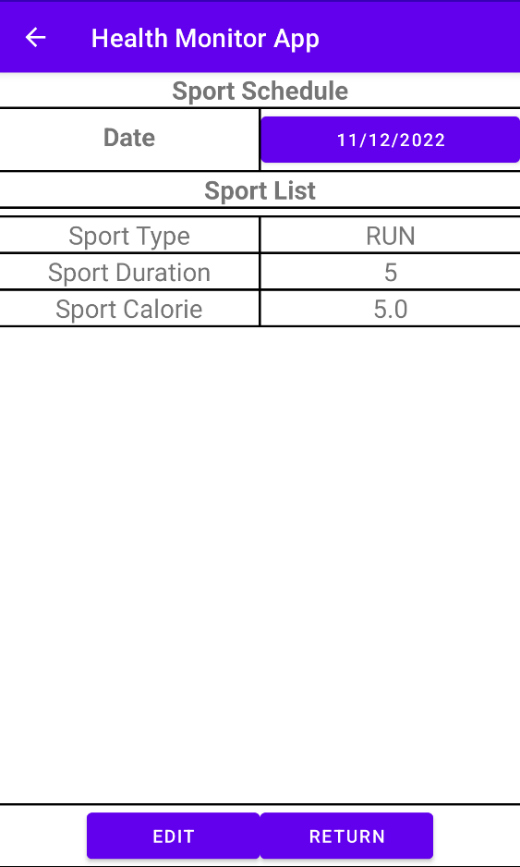


### Sport Type Statistics Recycler List Items



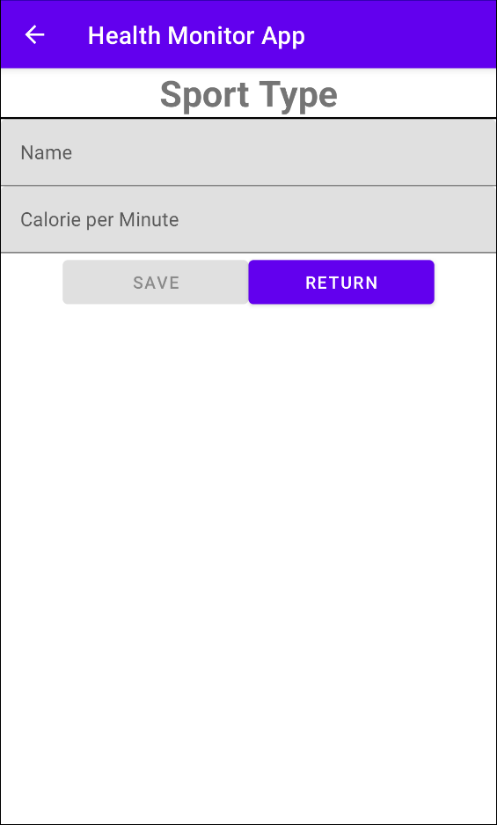
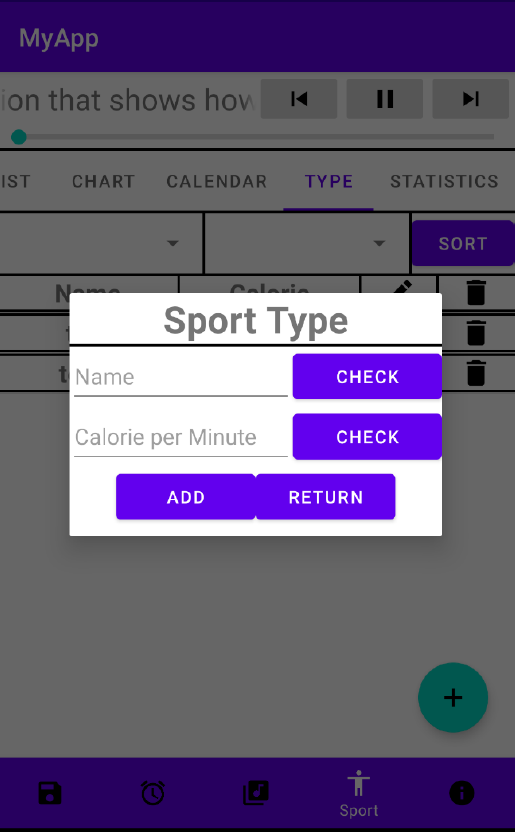
## Sport Data Activity Hidden Menu

The edit sport list menu of sport data activity is now initially hidden to allow users to view the sport list more easily. To start editing the sport list, users can click the edit button below.



## Sport Type Dialog to Activity

Sport Type dialog has also been changed to Sport Type Activity to maintain consistency in the app and to allow easier processing of data in an activity instead of a dialog.



## Save and Load Functionality Replaced

Save and load functionality was originally planned for users to rollback to an earlier save if they were unsatisfied with the changes or if they made a mistake. However, due to the complexity of the feature and time constraints, this feature was eventually removed from the app. It was replaced with a button to download and view user logs since account creation.

