



**UNIVERSITY
OF MALAYA**



GROUP PROJECT REPORT

COURSE CODE

: WIF3005

COURSE NAME

: SOFTWARE MAINTENANCE AND EVOLUTION

LECTURER

: DR. HASAN KAHTAN KHALAF

TUTORIAL NUMBER

: GROUP 3.4

SEMESTER / SESSION

: SEMESTER 1, SESSION 2020/2021

TITLE

: LOOP HABIT TRACKER MAINTENANCE AND EVOLUTION

NO.	NAME	MATRIC NUMBER
1.	Ng Jih Yann	17060539/1
2.	Dylan Salim	17123480/1
3.	Fang Wai Nam	17183792/1
4.	Lew Hooy Ye	17181202/1
5.	Tan Seok See	17080720/1
6.	Tan Qing Lin	17074063/1
7.	Yeap Eon Sein	17101551/1

Add Headings (Format > Paragraph styles) and they will appear in your table of contents.

Group Report (This is the format from question):

a) Content:

1. Cover page with details of all group members
 2. Table of contents
 3. Checklist (refer to Appendix A)
 4. Project's tasks
 5. Screen captures of the tool set-up and source code modification.
 6. Reference list
 7. Appendix containing complete source code of the system
- b) Marks will be deducted if citations and/or references are not properly done.
- c) Format: line spacing 1.5, font size 12. Illegible text and diagrams will be given zero mark.
- d) Page limit: 80 pages (excluding appendix that contains the source code). Exceeded pages will not be marked.

Project's Tasks:

- a) Create new project with repository. Could be cloned from existing project or students could develop new one. (Show screenshot)
- b) Working as a team with different roles. (Table of users roles)
- c) Analysis your selected code. (Write a bit why we need to analyse the code before merge, eg check whether there is any bugs/incorrect practice code involved, more understanding on the code base, identify components for maintenance and evolution + static code analysis/dynamic code analysis tool used, project size [according to number of lines or other criterias], is the project a legacy application? Is it using the latest technology? Software architecture used?)
- d) Identify the components that require maintenance and evolution (just like section 3 and 4 from melvin report).
- e) Carry on impact analysis before any changes. (this one need 2 ppl at least)
- f) Team leader assign tasks and responsibilities. (Table of tasks allocation)

g) Team leader review code changes and approved it. (Show the process of reviewing code change, eg. add comment and screenshot of success before accepting pull request, and close issue)

h) Continuously build and test code on preferred branch. (Refer to section 8 in melvin report, Provide mechanism how we continuously build and test, eg create issue, self assign issue, make a pull request, bind pull request to the issue, test the code, review before merging it to the master branch) - ½ - Build process ½ Test

i) Continuously deploy application to preferred environment. (Continuously Deployment using CodeMagic and put the release apk in the release section of the Github)

j) Rollback to preferred version if system experience any failure. (Screenshot failure of a application and rollback the code to previous version) [Reverting a commit - GitHub Docs](#)

k) Baseline and dashboard are essential for any code changes (Explain in theory and how it helps us in implementation)

25 marks here

Melvin Report :

<https://365umedumy.sharepoint.com/:w/s/SEFinalYearAssignmentGroup/Eb4Poh6FuTVKpIR1JZwoMc8BwlJE45W-nj5MmxMgs9McAw?e=346T5r>

Tasks Allocation for report: (Finish early go help others) - Deadline 23 Dec

Each member will do their own

d. Component for maintenance or Component for evolution +

5. Source code modification (include UI before after) +

h. test cases

- 1. Introduction (What the system do) + 3. Maintain and keep track checklist (1 ppl) - Darren

*Big Impact

- g. Review code + h. CI + Help team members to do administrative stuff and screenshots (1 ppl)

- FANG

- e. Impact Analysis (2 ppl) nicole, yann

- i. Continuous Deployment with CodeMagic and release apk + Help on Impact Analysis (1 ppl)

- Dylan

- Writing part for A, C + 5. VCS Tool Setup, references, report structuring [Table of content, font formatting, alignment, finding missing pieces, appendices] (1 ppl) hooyye
- Writing part for B, F, J, K + 5. Development Setup and environment, report structuring [Table of content, font formatting, alignment, finding missing pieces, appendices] (1 ppl) sushi

1.0 Introduction

The objective of this project is to maintain, evolve, collaborate, test build and deploy a targeted system. The utilization of tools such as Github is needed to achieve the project objective.

In this project, the team has selected a mobile application, Loop Habit Tracker, to do the maintenance and evolution. Github is used to improve the collaboration between the team members in doing the system maintenance and evolution. The application is used to keep track of the habits of the users and notifies the user when a scheduled habit meets the timer set by the user. This application helps users maintain good habits and allows users to achieve long-term goals. It will display a list of habits created by users and users can tick at the calendar when users complete the tasks set for the day.

In addition, the application has an advanced formula for calculating the strength of user habits. Every repetition makes the user's habit stronger and every missed day makes it weaker. A few missed days after a long streak, however, will not completely destroy user progress, unlike other don't-break-the-chain-applications. The result of the calculation is the Habit Score which is displayed as a line graph to the user when the user looks for the result overview of the habit. The application can track as many habits as users wish. There are no limitations. Besides, the application works offline and respects users' privacy. It does not require internet connection or online account registration. The confidential data is never sent to anyone. Neither the developers nor any third-parties have access to it.

After that, each habit can generate a result overview with the collected information. The result overview consists of detailed graphs and statistics that show how users' habits improved over time. The application is completely ad-free and open source.

2.0 Checklists

2.1 Intention-Based Classification Software Maintenance

Maintenance Type	Implementation Status	Person in Charge	Justification
Corrective	Completed	Tan Qing Lin	CR01 is completed
Adaptive	N/A	Tan Qing Lin	There are no adaptive maintenance change requests issued because the languages and dependencies used in the system are up-to-date. The system is an android application and does not plan to integrate to a new platform or operating system.
Perfective	Completed	Tan Qing Lin	CR04 is completed
Preventive	N/A	Ng Jih Yann	Not applicable because the chosen system is not categorized as a safety-critical system or high available software system. The programming style of the chosen system is good as well.

2.2 Activity-Based Classification Software Maintenance

Maintenance Type	Implementation Status	Person in Charge	Justification
Corrective	N/A	Ng Jih Yann	Not applicable to the

			system as no other error or bugs found other than error has been fixed in Intension-based corrective maintenance
Enhancive	Completed	Ng Jih Yann	CR02, CR03, CR05 are completed

2.3 Evidence-Based Classification Software Maintenance

Maintenance Type	Implementation Status	Person in Charge	Justification
Business Rule	Enhancive	N/A	Dylan Salim Not applicable to the system as there are no changes to the business rules
	Corrective	N/A	Dylan Salim Not applicable to the system as there are no changes to the business rules
	Reductive	N/A	Dylan Salim Not applicable to the system as there are no changes to the business rules
	Preventive	N/A	Tan Seok See Not applicable to the system as there are no changes to the business rules
Software Properties	Adaptive	N/A	Tan Seok See Not applicable to the system because it is still using the same technology and

				resources which are Android and SQLite
Performance	N/A	Tan Seok See		Not applicable to the system because no algorithms and components are replaced
Preventive	N/A	Lew Hooy Ye		Not applicable to the system because the maintenance activities carried out are based on modules and do not change the code structure and programming style of the system. No modification made upon the system structure that affects the maintainability of the system in any way
Groomative	N/A	Lew Hooy Ye		Not applicable as current algorithm and naming conventions have met the optimum performance requirement.
Documentation	Adaptive	N/A	Lew Hooy Ye	Not applicable because the system chosen from

				open-source does not disclose its documentation
	Reformative	N/A	Yeap Eon Sein	Not applicable because the system chosen from open-source does not disclose its documentation
Support Interface	Evaluative	Completed	Fang Wai Nam	Continuously review the code changes through Github
	Consultive	N/A	Yeap Eon Sein	Not applicable due to this project does not have a necessity to run a help desk since there are no customer
	Training	N/A	Fang Wai Nam	Not applicable as this project does not involve any stakeholders

2.4 Github Operation

Task	Implementation Status	Person in Charge	Justification
Create repository	✓	Fang Wai Nam	A repository name loop_habit_tracker is created for this project.
Add collaborator (team member)	✓	Fang Wai Nam	All group members have been invited as

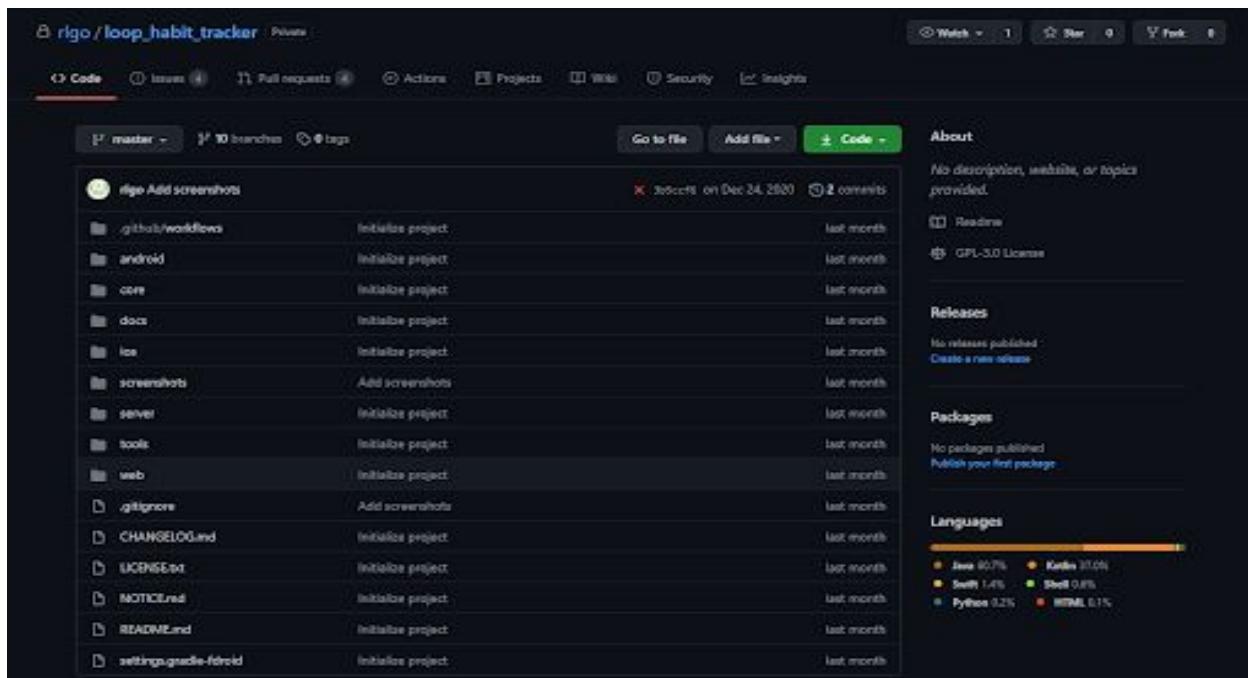
			collaborator for the repository so everyone has write access to the repository
Delegate function accordingly	✓	Fang Wai Nam	The project leader assign project's tasks evenly to every member so it is fair to all members
Create branch	✓	All members	A branch for each coding task. Each member has assigned at least a coding task. So there are a total of 10 branches including the master branch.
Make changes to code	✓	All members	Two members are assigned to do the evolution of the system and the other five members are assigned to do maintenance of the system. The five people are responsible to do five maintenance tasks to the system. More people are assigned to do evolution because there is a need to discover the libraries needed and implement the API to the system.
Proposed change request	✓	All members	All members have to test the system and determine the areas of improvement. Each member will need to list down potential

			improvements to the system and these improvements are made into change requests.
Review change and approve/reject change	✓	Ng Jih Yann & Yeap Eon Sein	Two members are assigned to review the change request and they are to approve or reject the change request. They would determine the validity of the change request through re-testing the system and impact analysis. Unfeasible change request is rejected.
Create Workflow (CI File)	✓	Fang Wai Nam & Dylan Salim	Workflow is created in Github action. Workflow is triggered to ensure the system can work smoothly and correctly after changes are done.
Merge file	✓	Fang Wai Nam	Project leader will review every committed request to ensure the code changes will have no impact on other components and avoid conflicts. Final version committed will be merged after reviewing the committed code.
Baseline (code, branch, etc)	✓	All members	Baseline is set for every change request by the previous properly working

			code
Update dashboard	√	All members	Simple Kanban board is created to help to keep track of the progress of the task of the project. Team members can review To-Do card for task need to be done and Completed card for task completed.

3.0 Project's Tasks

A) Create new project with repository. Could be cloned from existing project or students could develop new one.



The team leader has created a repository named `loop_habit_tracker` that is cloned from the existing repository for the system that we have chosen for maintenance and evolution. All the group members are added to the repository and have been given access to make changes on the repository. Every member will then upload their own maintenance or evolution part on the program to the branches created for each member. After that, the team leader will merge all the updates and modifications made. The link for the Github repository is https://github.com/rigo/loop_habit_tracker.

B) Working as a team with different roles.

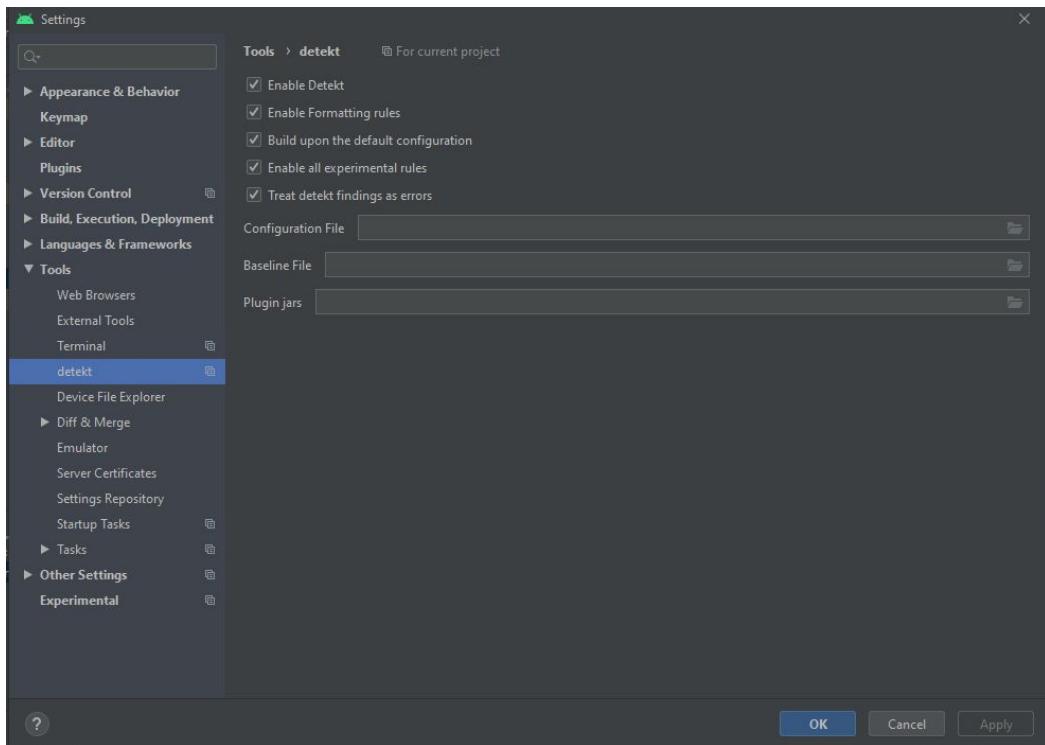
No	Name	Role
1	Ng Jih Yann	<ul style="list-style-type: none"> ● Evolution Personnel ● Tester
2	Dylan Salim	<ul style="list-style-type: none"> ● Evolution Personnel ● Tester
3	Fang Wai Nam	<ul style="list-style-type: none"> ● Project Leader ● Maintainer ● Tester
4	Lew Hooy Ye	<ul style="list-style-type: none"> ● Maintainer ● Tester
5	Tan Seok See	<ul style="list-style-type: none"> ● Maintainer ● Tester
6	Tan Qing Lin	<ul style="list-style-type: none"> ● Maintainer ● Tester
7	Yeap Eon Sein	<ul style="list-style-type: none"> ● Maintainer ● Tester

C) Analysis your selected code.

It is important for us to analyse the code before modification to detect and check if there is any error or bug involved in the code. It also helps to promote better understanding of the overall

program code so that we could identify the components that are needed to have maintenance and evolution.

We have carried out static code analysis by using Detekt which is a static analysis plugin used for Android Studio and IntelliJ, which is mainly for detecting "code smells", like those code patterns that might lead to trouble in the future. The Detekt plugin is installed and enabled in the Android Studio to use for static analysis of the code.



After running the detekt, it will help to identify the code style and code smell whether there is any presence of bad programming practice or strange code pattern. The complexity of the program could also be identified.

After analyzing the code, it can be said that the program is considered as a complex application system according to the program line and the relationship and dependency between the modules. The program has high interdependency and tight relationships between different modules, making the flow and connection of the program code hard to understand. Other than that, we also found out that although the source code does not consist of any visible mistakes or potential

errors, the code in overall is not that easy to understand through it. It might take some time to understand the flow of the program code and the dependency and relationship between different modules.

D) Identify the components that require maintenance and evolution(whichever its applicable).

After the team members test out the application and analyze the code, we have determined the components that are required for maintenance. This section describes the components and their respective types of maintenance activities. Each change request has been analyzed for its impact before approving the request. The column state explained the approval state of the request and if the request is rejected, the reasons are provided.

D.1 Components for Maintenance

Component	Change Request ID	Description	Type of Maintenance	Justification	Module Affected	State	Reasons	Remarks
Application Settings	CR01	Notifications are notified without sound, vibration and do not display on the mobile lockscreen.	Corrective (Intention-based)	Notifications notifying users through phone vibrations, notify sound and display on lock screen are considered a requirement for the system so this defect must be corrected	Notification Module	Accepted -		Done
	CR05	Add shortcut to application and link to specific	Enhancive (Activity-based)	Having shortcut allows the users to have quick access to	Shortcut Module	Accepted -		Done

		functionality of the application	particular functionality.			
Show Habits List	CR02	Habit can only be deleted on long press or click into the habit details	To achieve easy to use, users shall be allowed to easily delete a selected habit by left swiping the habit without changing the requirements.	Habit Module	Accepted -	Done
	CR03	Add favourite habit functionality	Feature that allow users to favourite one or more habits to enhance the system's functionality	Habit Module	Accepted -	Done
	CR04	Habits can only be viewed 4 days only and users need to scroll to the other	In order to monitor the habits every week easily, the day view is modified to 7	Habit Module	Accepted -	Done

	page to view the other days.	days without scrolling to another page.		
Create and Edit Habit Component	CR06	Analog time picker for setting the reminder time is changed to a digital time picker Perfective (Intention-based)	User can set the time more easier with digital time picker as compare to analog time picker Time Module	Rejected Too many modules will be affected by the change. Besides, the change does not bring benefits to the system. Users preference to type of time picker is different

D.2 Components for Evolution

D.2.1 Integration with Google Fit API

Our teams have brainstormed ideas for evolution of the system. The accepted idea for evolution is to integrate the Google Fit API into the mobile application with the purpose to analyze the impact of maintaining good habits on the fitness level of users. The data can be transferred seamlessly throughout all the users' Google Fit compatible devices, such as smartwatches. The application will collect 3 additional habits information from the user, the calorie burned, hydration requirements after the exercise and the activity duration of each activity. The data captured will be displayed in the form of a bar chart in the mobile app and be transferred to the Google Fit app to be analyzed.

Protecting user privacy is the utmost priority when it comes to data sharing between third party services. Therefore, the sharing of data between the application and the Google Fit API is not mandatory in the application. Users have the option to opt out from sharing their data to the Google Fit service during the habit creation.

The steps in creating this module is as follows:

1. Identifying appropriate APIs to be integrated in the mobile application
2. Register the application under Google Developer Console to gain access to the Google Fit API
3. Determine the data that need to be collected
4. Code and integrate the function to the system and its database
5. Design the layout of the data visualizations

Evolution Task ID	Task Descriptions
ET-001	Identify, create and design the input mechanisms for habits creation and edit.
ET-002	Establish communication between the application and the Google Fit API.
ET-003	Visualize the additional habit data that is transferred to Google Fit app in a bar chart.
ET-004	Create app linking with Google Fit app for users to open the Google Fit app to view the transferred data.

Component	Change Request ID	Description	Evolution Task ID	Module Affected	State	Reasons	Remarks
Habit Creation	CR07	Add custom input fields to capture additional habit information that can be transferred to the Google Fit app.	ET-001	Habit Module	Accepted	-	Done
Habit Detail	CR08	Transfer the captured data to Google Fit app using its API.	ET-002	Habit Module	Accepted	-	Done
Habit Detail	CR09	Retrieving the additional habit data and displaying it in a bar chart. Add function to open the Google Fit app.	ET-003	Habit Module	Accepted	-	Done

E) Carry on impact analysis before any changes.

Before the maintenance and evolution process is carried out, impact analysis must first be performed. Impact analysis means identifying the components that are impacted by the **Change Request** (CR). The impact analysis process involves identifying the **Starting Impact Set**(SIS), **Candidate Impact Set**(CIS), **Actual Impact Set**(AIS), **Discovered Impact Set**(DIS), and the **False Positive Impact Set**(FPIS).

Impact analysis begins with identifying the **SIS**. SIS is the initial set of objects presumed to be impacted. The change request(CR) specification as well as the source code are analyzed to find the SIS. We utilize the Android Studio Pattern Search Tool to find the SIS, the search tool will look for files that match the String search pattern that we entered.

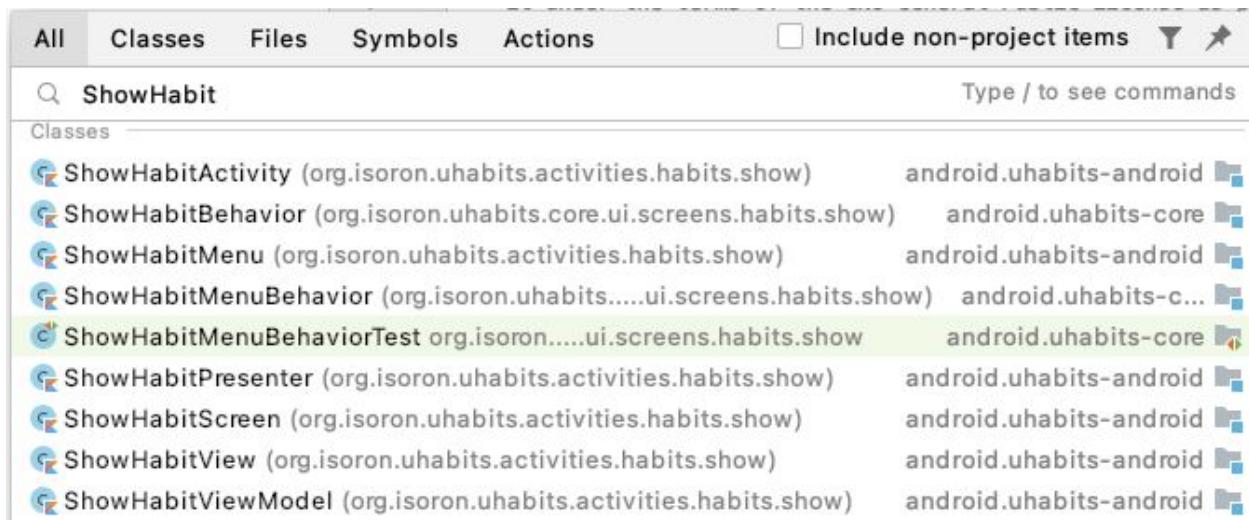


Figure 3.E.1

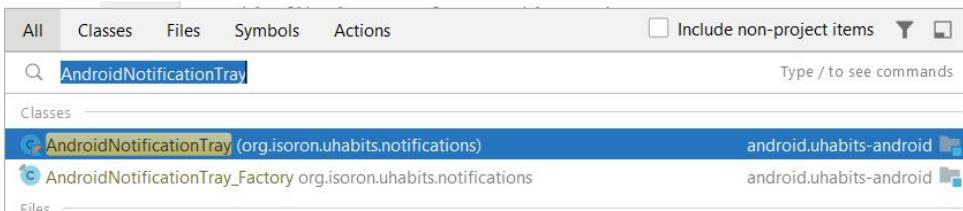
Figure E.1 shows the **Android Studio Pattern Search Tool**, for example, when we enter the ‘ShowHabit’ string as the search pattern, the files that matches the pattern will be listed.

After identifying the SIS, **CIS** is then identified by analyzing the code manually with the information provided from SIS. Then, we can start implementing the change requests, the **AIS**, **DIS** and **FPIS** are identified during and after implementation.

AIS is the set of objects actually changed as a result of performing CR. DIS is the set of objects not contained in the CIS but discovered during implementing CR. FPIS is the set of objects estimated to be impacted but not actually impacted.

E.1 Maintenance

E.1.1 CR1

Analysis Tool	Android Studio Pattern Search Tool
Analysis	 <p>The screenshot shows the Android Studio Pattern Search Tool interface. The search bar at the top contains the text "AndroidNotificationTray". Below the search bar, there are tabs for "All", "Classes", "Files", "Symbols", and "Actions". A checkbox labeled "Include non-project items" is checked. To the right of the search bar, there is a "Type / to see commands" button. The main results area shows two entries under the "Classes" tab: "AndroidNotificationTray (org.isoron.uhabits.notifications)" and "AndroidNotificationTray_Factory org.isoron.uhabits.notifications". Both entries have "android.uhabits-android" next to them. There is also a "File" tab below the classes tab.</p>
SIS	AndroidNotificationTray.kt AndroidNotificationTray_Factory.java
CIS	AndroidNotificationTray.kt AndroidNotificationTray_Factory.java AndroidManifest.xml RingtoneManager.kt
AIS	AndroidNotificationTray.kt AndroidManifest.xml RingtoneManager.kt
DIS	-
FPIS	AndroidNotificationTray_Factory.java

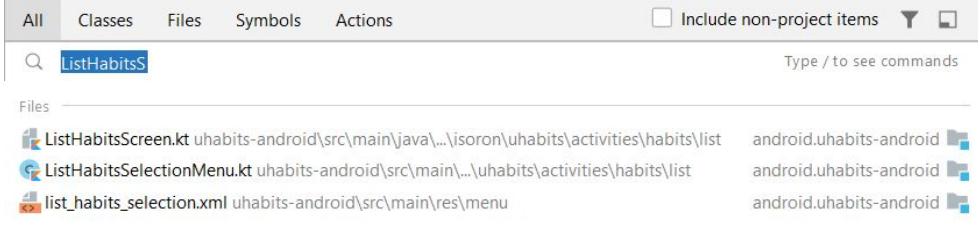
E.1.2 CR2

Analysis Tool	Android Studio Pattern Search Tool
----------------------	------------------------------------

Analysis	
SIS	HabitCardListController.kt HabitCardListView.kt ListHabitsBehavior.java ListHabitsBehaviorTest.java ListHabitsMenuBehavior.java ListHabitsMenuBehaviorTest.java ListHabitsSelectionMenuBehavior.java ListHabitsSelectionMenuBehaviorTest.java
CIS	HabitCardListController.kt HabitCardListView.kt ListHabitsBehavior.java ListHabitsBehaviorTest.java ListHabitsMenuBehavior.java ListHabitsMenuBehaviorTest.java ListHabitsSelectionMenu.kt ListHabitsSelectionMenuBehavior.java ShowHabitMenu.kt ShowHabitMenuBehavior.kt SwipeDeleteHabitTouchListener.java SwipeToDeleteCallback.kt ic_delete_white_24.xml
AIS	HabitCardListController.kt HabitCardListView.kt ListHabitsBehavior.java ListHabitsBehaviorTest.java ListHabitsSelectionMenu.kt ListHabitsSelectionMenuBehavior.java ShowHabitMenu.kt ShowHabitMenuBehavior.kt

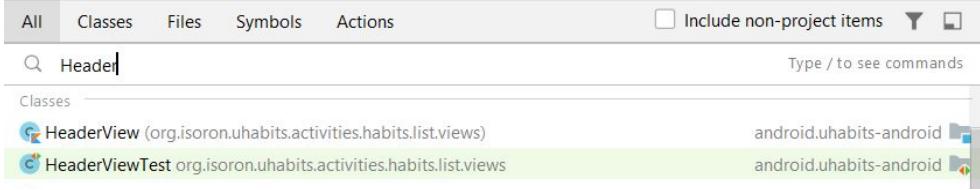
	SwipeDeleteHabitTouchListener.java SwipeToDeleteCallback.kt ic_delete_white_24.xml
DIS	OnConfirmedCallback.java OnCancelledCallback.java
FPIS	ListHabitsMenuBehavior.java ListHabitsMenuBehaviorTest.java

E.1.3 CR3

Analysis Tool	Android Studio Pattern Search Tool
Analysis	 <p>The screenshot shows the Android Studio Pattern Search Tool interface. The search bar at the top contains the text "ListHabits". Below the search bar, there is a toolbar with tabs for "All", "Classes", "Files", "Symbols", and "Actions". To the right of the toolbar is a checkbox labeled "Include non-project items" and a "Type / to see commands" input field. The main area displays search results under the heading "Files". The results list includes three items: "ListHabitsScreen.kt", "ListHabitsSelectionMenu.kt", and "list_habits_selection.xml", each with its file path and a small preview icon.</p>
SIS	ListHabitsScreen.kt ListHabitsSelectionMenu.kt list_habits_selection.xml
CIS	ListHabitsScreen.kt ListHabitsSelectionMenu.kt list_habits_selection.xml strings.xml FavouriteHabitsCommand.java UnfavouriteHabitsCommand.java Habit.java HabitRecord.java ListHabitsSelectionMenuBehavior.java 25.sql HabitCardView.kt
AIS	ListHabitsScreen.kt ListHabitsSelectionMenu.kt list_habits_selection.xml strings.xml FavouriteHabitsCommand.java UnfavouriteHabitsCommand.java

	Habit.java HabitRecord.java ListHabitsSelectionMenuBehavior.java 25.sql HabitCardView.kt
DIS	Config.java
FPIS	-

E.1.4 CR4

Analysis Tool	Android Studio Pattern Search Tool
Analysis	 <p>The screenshot shows the Android Studio Pattern Search Tool interface. The search bar at the top contains the text "Header". Below the search bar, there are tabs for "All", "Classes", "Files", "Symbols", and "Actions". A checkbox labeled "Include non-project items" is checked. The results list shows two entries under the "Classes" tab: "HeaderView (org.isoron.uhabits.activities.habits.list.views)" and "HeaderViewTest org.isoron.uhabits.activities.habits.list.views". Both entries have "android.uhabits-android" next to them.</p>
SIS	HeaderView.kt HeaderViewTest.kt
CIS	HeaderView.kt HeaderViewTest.kt dimens.xml
AIS	HeaderView.kt dimens.xml
DIS	-
FPIS	HeaderViewTest.kt

E.1.5 CR5

Analysis Tool	Android Studio Pattern Search Tool
----------------------	------------------------------------

Analysis	<p>All Classes Files Symbols Actions <input type="checkbox"/> Include non-project items </p> <p> AndroidManifest Type / to see commands</p> <p>Files AndroidManifest.xml uhabits-android\src\main android.uhabits-android </p> <p> strings Type / to see commands</p> <p>Files strings.xml android-base\src\main\res\values android.android-base </p>
SIS	AndroidManifest.xml strings.xml
CIS	AndroidManifest.xml strings.xml ActivityOne.kt ActivityTwo.kt activity_one.xml activity_two.xml shortcut.xml
AIS	AndroidManifest.xml strings.xml ActivityOne.kt ActivityTwo.kt activity_one.xml activity_two.xml shortcut.xml
DIS	-
FPIS	-

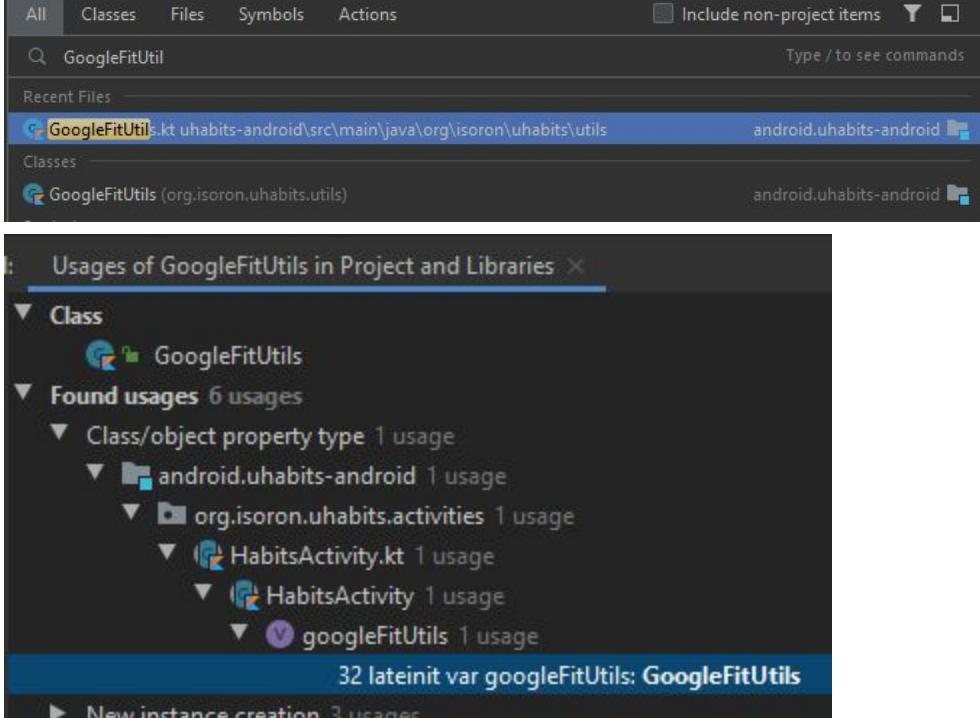
E.2 Evolution

E.2.1 CR07

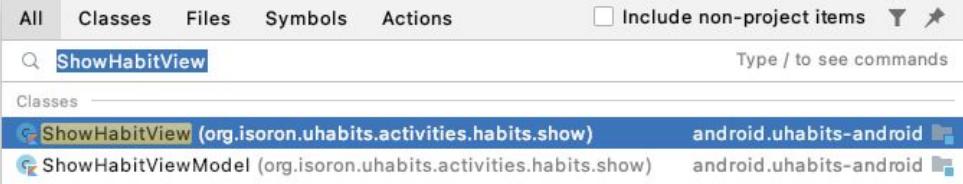
Analysis Tool	Android Studio Pattern Search and Find Usage Tool
----------------------	---

Analysis	<p>All Classes Files Symbols Actions <input type="checkbox"/> Include non-project items <input type="text"/> Type / to see commands</p> <p>Classes EditHabitActivity (org.isoron.uhabits.activities.habits.edit) android.uhabits-android</p> <p>Find: Usages of buildHabitListRepository() in Project ...</p> <ul style="list-style-type: none"> Method <ul style="list-style-type: none"> (m) buildHabitListRepository() Found usages 1 usage <ul style="list-style-type: none"> Unclassified usage 1 usage <ul style="list-style-type: none"> android.uhabits-core 1 usage <ul style="list-style-type: none"> org.isoron.uhabits.core.models.sqlite 1 usage <ul style="list-style-type: none"> SQLiteHabitList 1 usage m SQLiteHabitList(ModelFactory) 1 usage <pre>59 this.repository = modelFactory.buildHabitListRepository();</pre>
SIS	EditHabitActivity.kt
CIS	EditHabitActivity.kt show_habit.xml HabitTypeDialog.kt HabitsApplication.kt HabitRecord.java Habits.java
AIS	EditHabitActivity.kt show_habit.xml HabitTypeDialog.kt HabitsApplication.kt HabitRecord.java Habits.java
DIS	SQLiteHabitList.java EditHabitActivity.kt show_habit.xml HabitTypeDialog.kt HabitsApplication.kt HabitRecord.java Habits.java
FPIS	-

E.2.2 CR08

Analysis Tool	Android Studio Pattern Search Tool
Analysis	 <p>The screenshot shows the 'Usages' search results for 'GoogleFitUtil'. The search bar at the top has 'GoogleFitUtil' typed in. Below it, the 'Recent Files' section shows 'GoogleFitUtils.kt' from 'uhabits-android' and 'GoogleFitUtils (org.isoron.uhabits.utils)' from 'android.uhabits-android'. The main panel displays the 'Usages of GoogleFitUtils in Project and Libraries' with a tree view. It starts with a 'Class' node for 'GoogleFitUtils', which then branches into 'Found usages' (6 usages). These usages are categorized by file type: 'Class/object property type' (1 usage), 'New instance creation' (3 usages), and 'lateinit var' (3 usages). The 'lateinit var' category is expanded, showing 'googleFitUtils' (1 usage) in 'HabitsActivity.kt'.</p>
SIS	GoogleFitUtils.kt
CIS	GoogleFitUtils.kt AndroidManifest.xml build.gradle HabitsActivity.kt HabitCardView.kt
AIS	GoogleFitUtils.kt AndroidManifest.xml build.gradle HabitsActivity.kt HabitCardView.kt
DIS	ListHabitsScreen.kt ListHabitsBehavior.java
FPIS	-

E.2.3 CR09

Analysis Tool	Android Studio Pattern Search Tool
Analysis	 <p>The screenshot shows the 'Pattern Search' dialog in Android Studio. The search term 'ShowHabitView' is entered in the search bar. The results list shows two items under 'Classes': 'ShowHabitView (org.isoron.uhabits.activities.habits.show)' and 'ShowHabitViewModel (org.isoron.uhabits.activities.habits.show)'. Both items are associated with the package 'android.uhabits-android'.</p>
SIS	ShowHabitView.kt ShowHabitViewModel.kt
CIS	ShowHabitView.kt ShowHabitViewModel.kt show_habit.xml ShowHabitActivity.kt BarCard.kt ShowHabitBehaviour.kt
AIS	ShowHabitView.kt ShowHabitViewModel.kt show_habit.xml ShowHabitActivity.kt BarCard.kt ShowHabitBehaviour.kt
DIS	Peferences.java CalorieBarCard.kt HydrationBarCard.kt ActivityDurationBarCard.kt show_habit_caloriebar.xml show_habit_hydrationbar.xml show_habit_hydrationbar.xml
FPIS	-

F) Team leader assign tasks and responsibilities.

No	Name	Tasks/Responsibilities
1	Ng Jih Yann	- Work on evolution aspect (CR06)

		<ul style="list-style-type: none"> - Impact analysis
2	Dylan Salim	<ul style="list-style-type: none"> - Work on evolution aspect (CR06) - Continuous Integration and Deployment (CI/CD) of the application
3	Fang Wai Nam	<ul style="list-style-type: none"> - Create a GitHub repository - Add all members as collaborators - Review code, merge pull requests - Work on maintenance aspect (CR03)
4	Lew Hooy Ye	<ul style="list-style-type: none"> - Work on maintenance aspect (CR05) - Code analysis
5	Tan Seok See	<ul style="list-style-type: none"> - Work on maintenance aspect (CR02) - Rollback to preferred version if system experiences failure
6	Tan Qing Lin	<ul style="list-style-type: none"> - Work on maintenance aspect (CR01) - Make sure that every planned tasks have been carried out
7	Yeap Eon Sein	<ul style="list-style-type: none"> - Work on maintenance aspect (CR04) - Impact analysis

G) Team leader review code changes and approved it.

When the collaborators have done their development on a branch and want a branch to be merged in master branch, a pull request will be created. The team leader will review the request and ensure no conflict or all conflict is resolved. If the pull request is approved, the changes will be merged into main branch.

H) Continuously build and test code on preferred branch.

H.1 Continuous Integration (CI)

To make a change request, a collaborator must create an issue to describe the current problem and possible improvements. After creating an issue, a collaborator must create a branch from the master branch and all code modification will go to this branch. When the code is ready, a change

request will be made with linked issue. Every time a change request is made, codemagic CI workflow will be triggered and run automated build and test. If the CI result is successful, the team leader will review all the changed files to ensure that it does not introduce any conflict, test by running the application and finally approve the change request if everything is working as expected. A new version of apk will then be created and added to the release section.

Figure 3.H.1.1 List of issues

The table below shows the created issues and their respective branches

Issue ID	Issue	Branch	Assignee
#2	Maintenance: Swipe to Delete	swipeToDelete	Tan Seok See
#3	Evolution: Integrate with the Google Fit API to analyze the impact of	inputFieldsGoogleFit	Dylan Salim

	maintaining good habits on the fitness level of user - Add input fields for calorie burned, hydration and activity duration		
#5	Maintenance: Notification sound, vibrates, and display on lock screen	configNotification	Tan Qing Lin
#7	Evolution: Integrate with the Google Fit API to analyze the impact of maintaining good habits on the fitness level of user - Visualize additional habit data in bar charts, App linking with Google Fit app for user redirection	customFieldsBarChart	Ng Jih Yann
#10	Maintenance: Habits Day Header Modified to 7 / 14 Days View	modifyDayView	Yeap Eon Sein
#14	Maintenance: Favorite Habit	Maintenance-CR03-favoriteHabits	Fang Wai Nam
#16	Evolution: Integrate with the Google Fit API to analyze the impact of maintaining good habits on the fitness level of user - Establish communication between the application and the Google Fit API.	Evolution-CR08-GoogleFitIntegration	Dylan Salim
#19	Maintenance: Add Shortcut	Maintenance-CR05-shortcut	Lew Hooy Yee

H.2 Test Cases

These test cases have been developed to test the functionality of each module added and verify whether the impacts of the added functionality. The test cases have been designed according to the components available and the Change Requests (CRs) made.

a. Show Habits List Component

Table: Verify the Impact of CR02 on the Show Habits List Component

Test Case ID	TC-01-001
Component	Show Habits List
Change Requests	CR02

Objective	To verify that CR02 has been implemented successfully.			
Input	Expected Result	Special Procedural Requirements	Intercase Dependency	Pass/Fail
Hold and left swipe a habit on the list.	The selected habit will be deleted permanently.	N/A	N/A	Pass
Left swipe a habit and swipe the habit back to its original position.	The selected habit will remain in the list and database.	N/A	N/A	Pass
Right swipe a habit.	Nothing will happen.	N/A	N/A	Pass

Table: Verify the Impact of CR03 on the Show Habits List Component

Test Case ID	TC-01-002
--------------	-----------

Component	Show Habits List			
Change Requests	CR03			
Objective	To verify that CR03 has been implemented properly without error.			
Input	Expected Result	Special Procedural Requirements	Intercase Dependency	Pass/Fail
Long press on a habit and select favourite from the top right corner of the menu bar.	The selected habit will be favourite and a star icon will be added next to the name of the habit.	N/A	N/A	Pass
Long press on favourite habit and select unfavourite from the top right corner of the menu bar.	The favourite attribute of the selected habit will be false and the star icon will be removed in the habits list.	N/A	N/A	Pass

Test Case ID	TC-01-003			
Component	Show Habits List			
Change Requests	CR04			
Objective	To verify that CR04 has been implemented successfully.			
Input	Expected Result	Special Procedural Requirements	Intercase Dependency	Pass/Fail
Enter the Habits List	The habits day header has a 7 days	N/A	N/A	Pass

Screen with phone portrait mode.	(1 week) view. Users can scroll to the following days to view previous weeks' habits.		
Enter the Habits List Screen with phone landscape mode.	The habits day header has a 14 days view (2 weeks). Users can scroll to the following days to view previous weeks' habits.	N/A	Pass

b. Show Habits Detail Component

Table: Verify the Impact of CR08 on the Show Habits Detail Component

Test Case ID	TC-02-001		
Component	Show Habits Detail Component		
Change Requests	CR08		
Objective	To verify that CR08 has been implemented successfully.		
Input	Expected Result	Special Procedural Requirements	Intercase Dependency Pass/Fail
Add habit frequency from the calendar section.	The addition of habit calorie, hydration and activity duration data is sent to the Google Fit API.	N/A	Pass
Test data 1: Add 1 frequency			

Delete habit frequency from the calendar section.	The reduction of habit calorie, hydration and activity duration data is sent to the Google Fit API.	N/A	N/A	Pass
Test data 2:				
Remove 1 frequency				

Table: Verify the Impact of CR09 on the Show Habits Detail Component

Test Case ID	TC-02-002	Component	Show Habits Detail Component	Change Requests	CR09	Objective	To verify that CR09 has been implemented successfully.	Input	Expected Result	Special Procedural Requirements	Intercase Dependency	Pass/Fail
Enter value for additional habit fields(calorie burned, activity duration, hydration level).	The data of additional habit fields are displayed in bar charts.	N/A	N/A	Test data: Calorie : 100kcal Activity duration: 100 seconds Hydration level: 3 litres								Pass

Press on the ‘Open Google Fit’ button.	User is redirected to Google Fit app if Google Fit has been installed. If Google Fit has not been installed, user will be redirected to Google Playstore to download the Google Fit app.	N/A	N/A	Pass
--	---	-----	-----	------

c. Edit Habits Component

Table: Verify the Impact of CR07 on the Edit Habits Component

Test Case ID	TC-03-001	Component	Edit Habits Component	Change Requests	CR07	Objective	To verify that CR07 has been implemented successfully.	Input	Expected Result	Special Procedural Requirements	Increase Dependency	Pass/Fail
Enable Google Fit API and edit habits Test data 1: Enable Google Fit: True Calorie: 80kcal	Calorie, hydration and activity duration data is updated in the database.	N/A	N/A	Pass								

Hydration: 1.8 litre Activity Duration: 110s			
Disable Google Fit API and edit habits Test data 2: Enable Google Fit False	Calorie, hydration and activity duration data is updated to 0 in the database.	N/A	Pass

d. Create Habits Component

Table: Verify the Impact of CR07 on the Create Habits Component

Test Case ID	TC-04-001	Component	Create Habits Component	Change Requests	CR07	Objective	To verify that CR07 has been implemented successfully.	Input	Expected Result	Special Procedural Requirements	Increase Dependency	Pass/Fail
Enable Google Fit API and create habits Test data 1:	Calorie, hydration and activity duration data is stored in the database. New habit was created.	N/A	N/A									Pass

Enable Google Fit: True		
Calorie: 60kcal		
Hydration: 1.5 litre		
Activity Duration: 120s		

e. Application Settings Component

Table: Verify the Impact of CR01 on the Application Settings Component

Test Case ID	TC-05-001	Component	Application Settings Component	Change Requests	CR01	Objective	To verify that CR01 has been implemented successfully.	Input	Expected Result	Special Procedural Requirements	Intercase Dependency	Pass/Fail
Schedule a reminder for a habit:	Reminder notification notify at 9:00 A.M everyday with sound, vibration and display on lock screen	N/A	N/A	Test data 1:								Pass
Time: 9:00 A.M Day: Everyday												
Change the notification sound. Schedule a reminder for a habit:	Reminder notification notify at 7:00 P.M on Monday with new sound set, vibration and display on lock screen	N/A	N/A									Pass

Test data 1:		
Time: 7:00 P.M		
Day: Monday		

Table: Verify the Impact of CR05 on the Application Settings Component

Test Case ID	TC-05-002			
Component	Application Shortcut Component			
Change Requests	CR05			
Objective	To verify that CR05 has been implemented successfully.			
Input	Expected Result	Special Procedural Requirements	Intercaese Dependency	Pass/Fail
Long Press on application icon	Pop out a shortcut list with setting, about and create habit.	N/A	N/A	Pass
Press on “Create Habit” shortcut label	Redirect to the create habit page in the application.	Shortcut list is displayed.	N/A	Pass
Long press on the “About” shortcut label and drag to homescreen	“About” shortcut is created at the homescreen.	Shortcut list is displayed.	N/A	Pass

i) Continuously deploy application to preferred environment.

Each time the project leader merges the pull request to the master branch, he will build the Android APK file and deploy it to the release section of Github.

[Screenshot of release section]

Our group also used an automated deployment tool we used in this project is CodeMagic. It is a build automation tool that specialize for mobile application development. It provides 500 free build minutes every month and it can integrate with Github easily. CodeMagic supports easy deployment of Android APK to Github releases section. We have set up an automated deployment pipeline that will listen to any push or pull request on the master branch, continuously build and publish the application.

After creating a CodeMagic project and linking it to the Github repository, we provide all the commands and configuration metadata using a yaml file. The configuration yaml file consists of the logic to build the Android installation APK file, send the built APK to the recipients by email and publish it to the release section in the Github repository.



The screenshot shows a GitHub code editor interface with a dark theme. At the top, there is a header bar with the text "16 lines (16 sloc) | 504 Bytes". On the right side of the header, there are several small icons: "Raw", "Blame", "Copy", and "Edit". The main content area contains a YAML configuration file named "codemagic.yaml". The code is color-coded, with keywords in green and values in white. The configuration includes sections for workflows, android-app, scripts, artifacts, publishing, email, recipients, and github_releases.

```
1 workflows:
2   android-app:
3     name: loop-habits
4     scripts:
5       - cd android && chmod +x gradlew && ./gradlew assembleDebug
6     artifacts:
7       - android/uhabits-android/build/outputs/**/*.apk
8     publishing:
9       email:
10         recipients:
11           - dylansalim015@gmail.com
12           - fangwainam7@gmail.com
13     github_releases:
14       prerelease: false
15       artifact_patterns:
16         - '*.apk'
```

Figure 3.I.1: codemagic.yaml configuration file

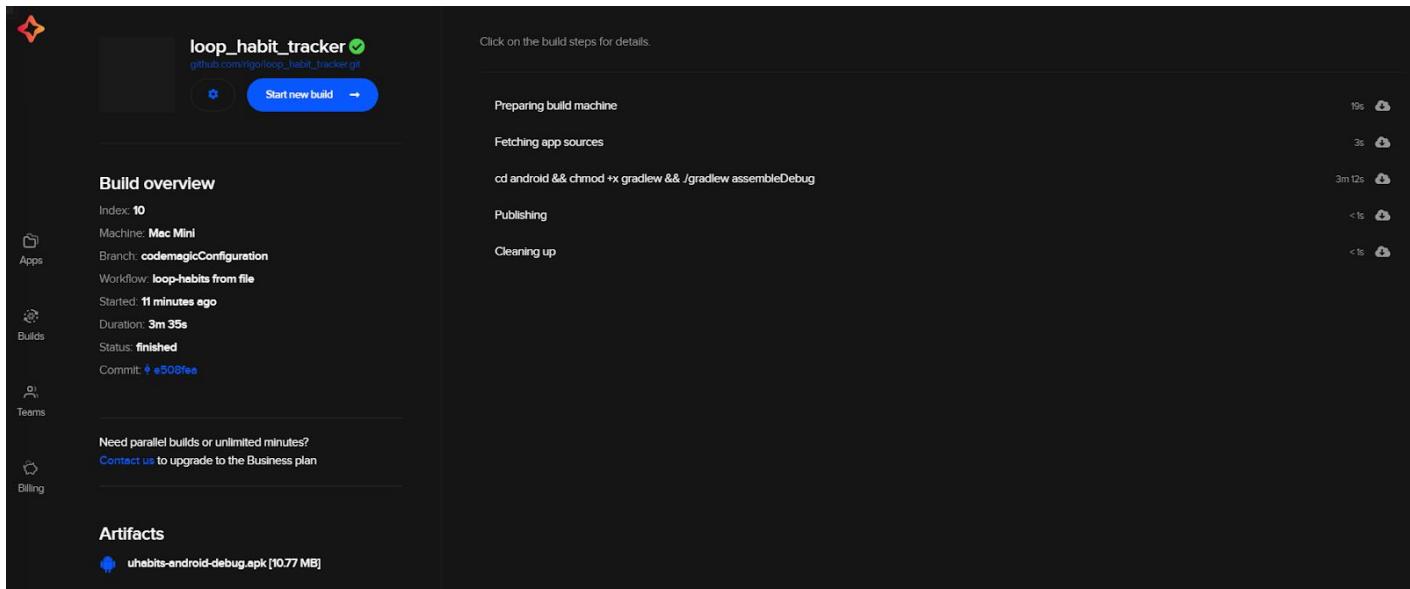


Figure 3.I.2: Pipeline Building Process of CodeMagic

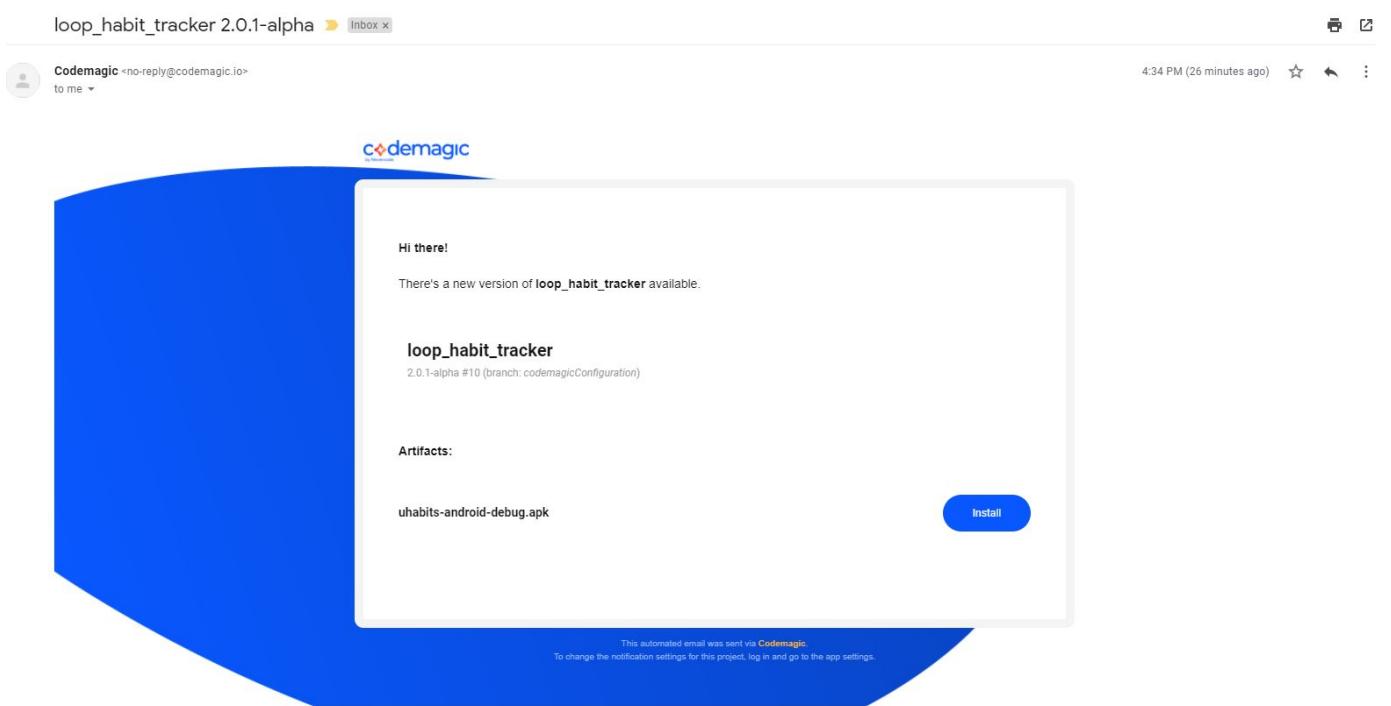
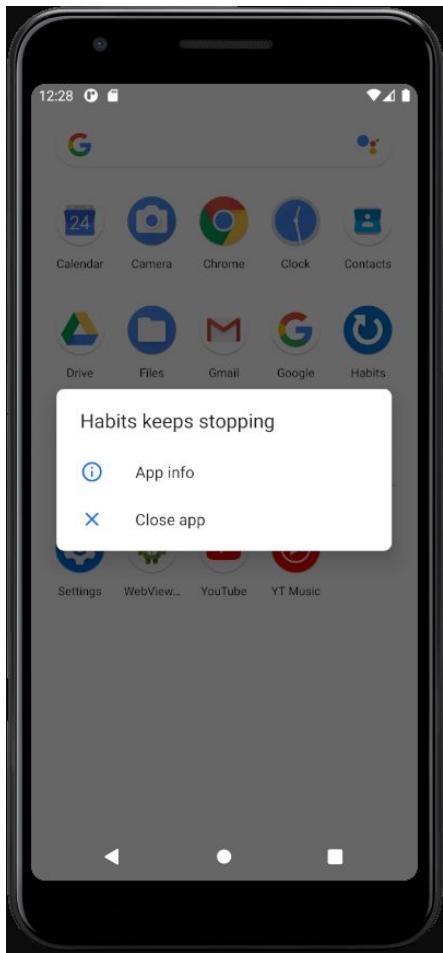


Figure 3.I.3: APK artifacts email sent by CodeMagic

j) Rollback to preferred version if system experiences any failure.



A catastrophic failure has occurred after 513c9d49de5634cdad645bc20a5940204eea60ef commit with the commit message “change database version”, therefore the system is rolled back to the previous version before this commit.

Figure 3.J.1 Application Failure

Commits on Jan 24, 2021

Commit	Author	Date	Message	Revert Commit	Revert Author	Revert Date
Revert "change database version"	seoksee	1 minute ago	...	b2c1528		
change database version	seoksee	9 minutes ago	X	0e1a9a0		

Figure 3.J.2 Revert the failed version

It is interesting to note that the revert itself is also a commit in git and the failed version of commit still remains in the repository.

k) Baseline and dashboard are essential for any code changes.

Branch name	configNotification
Change Request(s)	CR01
Collaborator(s)	Tan Qing Lin
Reviewed by	Fang Wai Nam
Approved by	Fang Wai Nam
Screenshot	<p>The screenshot shows a GitHub pull request interface. At the top, it indicates the pull request has been merged. Below this, a comment from user 'devildarren1997' states: 'Notification module is configured. Solve issue #5 . The notification should notify with sound, vibration and can display on lock screen. The notification sound is now default set to a notification sound found by myself online.' Below the comment, a list of commits is shown, each with a small profile icon, a commit message, and a unique identifier:</p> <ul style="list-style-type: none"> configured Notification (52c3a68) changes made (27c9079) update (2721828) changes made (e4d2366) <p>Further down, another comment from 'devildarren1997' adds the 'maintenance' label. Subsequent comments from 'dylansalim3' and 'devildarren1997' show the status moving from 'TODO - MAINTENANCE' to 'COMPLETED - MAINTENANCE'. At the bottom, a merge commit is shown, with 'View details' and 'Revert' buttons. A final review comment from 'rigo' states: 'Reviewed and tested . All good. Will merge and release to version 2.0.1'. A linked issue titled 'Maintenance: Notification sound, vibrates, and display on lock screen #5' is marked as closed.</p>

Branch name	swipeToDelete
Change Request(s)	CR02
Collaborator(s)	Tan Seok See
Reviewed by	Fang Wai Nam
Approved by	Fang Wai Nam
Screenshot	<p>The screenshot shows a GitHub pull request titled "Swipe to delete #4". The pull request has been merged by rigo. The commit history shows several commits from seoksee, including adding swipe-to-delete functionality, removing unused files, and adding a red background on swipe. The pull request also includes a link to an issue labeled "Maintenance: Swipe to Delete #2" which is now closed. DylanSalim3 added the pull request to the "TODO - MAINTENANCE" column in the Loop_Habits_Maintenance_&_Evolution board. rigo approved the changes, stating "Reviewed and tested. All good. Will add this to release v2.0.2". The pull request was merged into the master branch by rigo, and the merge was successful.</p>

Branch name	Maintenance-CR03-favouriteHabits
Change Request(s)	CR03

Collaborator(s)	Fang Wai Nam
Reviewed by	Fang Wai Nam
Approved by	Fang Wai Nam
Screenshot	<p>The screenshot shows a GitHub pull request interface. At the top, it says "Merged rigo merged 1 commit into master from Maintenance-CR03-favouriteHabits 6 hours ago". Below this, there are tabs for Conversation (0), Commits (1), Checks (4), and Files changed (12). The main area shows a comment from "rigo" that reads "Solve issue #14 Allow user to add habit as favorite". Below the comment, a commit titled "favorite habit" is shown, with a link to "Maintenance: Favorite Habit #14" which is marked as "Closed". Another commit from "rigo" adds the "maintenance" label. A merge commit follows, merging "44b6ac6" into "master" with "2 of 4 checks passed". The final message indicates the pull request was successfully merged and closed, with a button to "Delete branch".</p>

Branch Name	modifyDayView
Change Request(s)	CR04
Collaborator(s)	Yeap Eon Sein
Reviewed by	Fang Wai Nam
Approved by	Fang Wai Nam

Screenshot	<h3>Modify Day Header View #13</h3> <p>Merged rigo merged 2 commits into <code>master</code> from <code>modifyDayView</code> 6 hours ago</p> <p>Conversation 1 Commits 2 Checks 4 Files changed 0</p> <p>eonsein commented 19 hours ago No description provided.</p> <p>eonsein added 2 commits 19 hours ago</p> <ul style="list-style-type: none"> lastCommit 3a8f066 lastCommit 3db552a <p>eonsein added the <code>maintenance</code> label 19 hours ago</p> <p>eonsein linked an issue that may be closed by this pull request 19 hours ago Maintenance: Habits Day Header Modified to 7 / 14 Days View #10 (Closed)</p> <p>rigo approved these changes 6 hours ago View changes</p> <p>rigo left a comment Reviewed and tested. Will add to release v2.0.3</p> <p>rigo merged commit 355b27e into <code>master</code> 6 hours ago 2 of 4 checks passed View details Revert</p> <p>Pull request successfully merged and closed Delete branch You're all set—the <code>modifyDayView</code> branch can be safely deleted.</p>
------------	--

Branch Name	
Change Request(s)	CR05
Collaborator(s)	Lew Hooy Ye
Reviewed by	Fang Wai Nam
Approved by	Fang Wai Nam
Screenshot	

Branch Name	inputFieldsGoogleFit
Change Request(s)	CR07

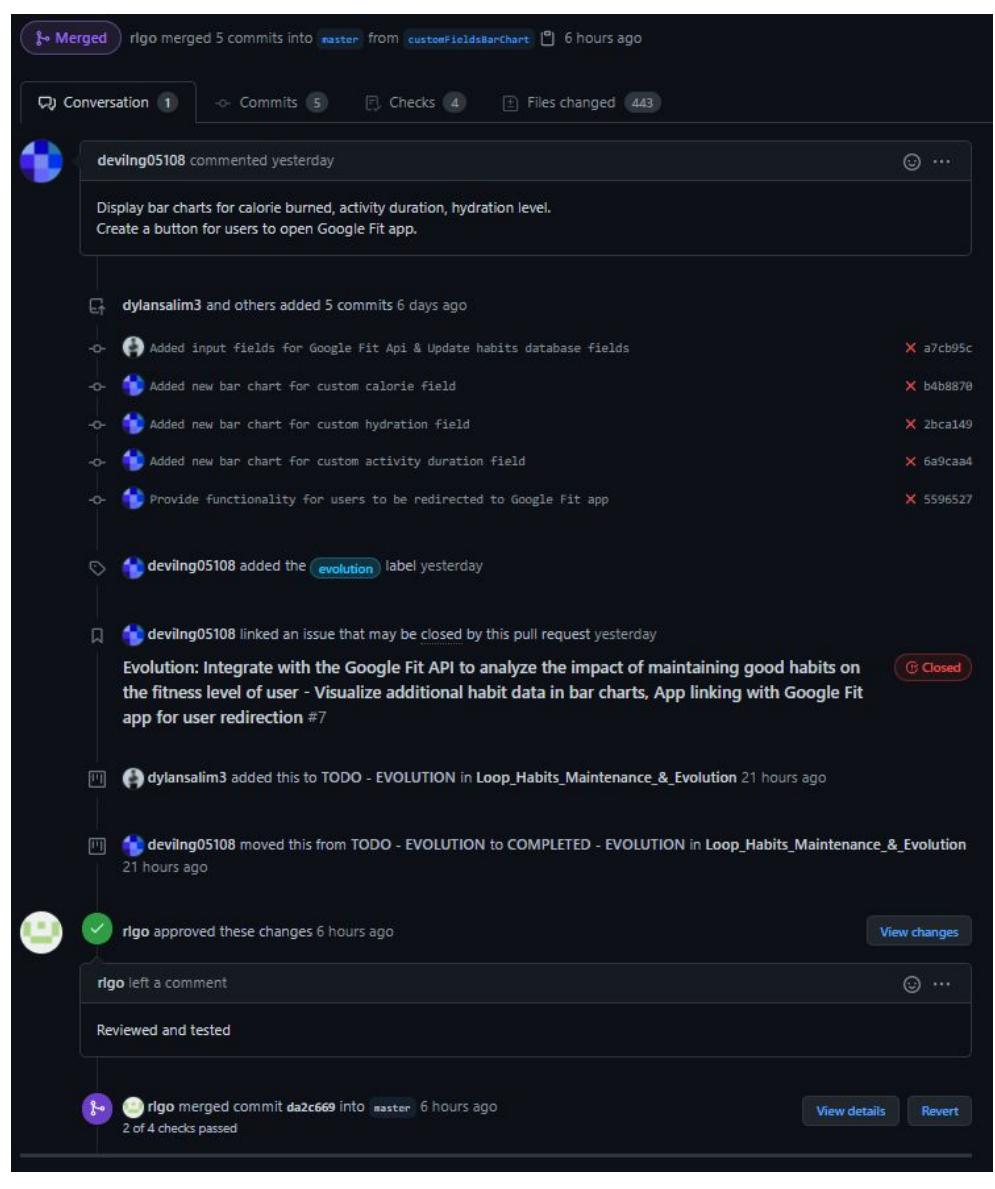
Collaborator(s)	Dylan Salim
Reviewed by	Fang Wai Nam
Approved by	Fang Wai Nam
Screenshot	<p>The screenshot shows a GitHub pull request merge history. At the top, it says "Added input fields for Google Fit Api & Update habits database fields #9" with a "Merged" button. Below this, Dylan Salim has added several comments and moved items between milestones: "Add input fields for calorie burned, hydration and activity duration", "evolution", "TODO - EVOLUTION", "Loop_Habits_Maintenance_&_Evolution", "COMPLETED - EVOLUTION", "MAINTENANCE", and "EVOLUTION". Fang Wai Nam approved the changes and merged the commit 85dd3f7 into the master branch 6 hours ago. A message at the bottom states "Pull request successfully merged and closed" and "Delete branch".</p>

Branch Name	
Change Request(s)	CR08
Collaborator(s)	Dylan Salim

Reviewed by	Fang Wai Nam
Approved by	Fang Wai Nam
Screenshot	

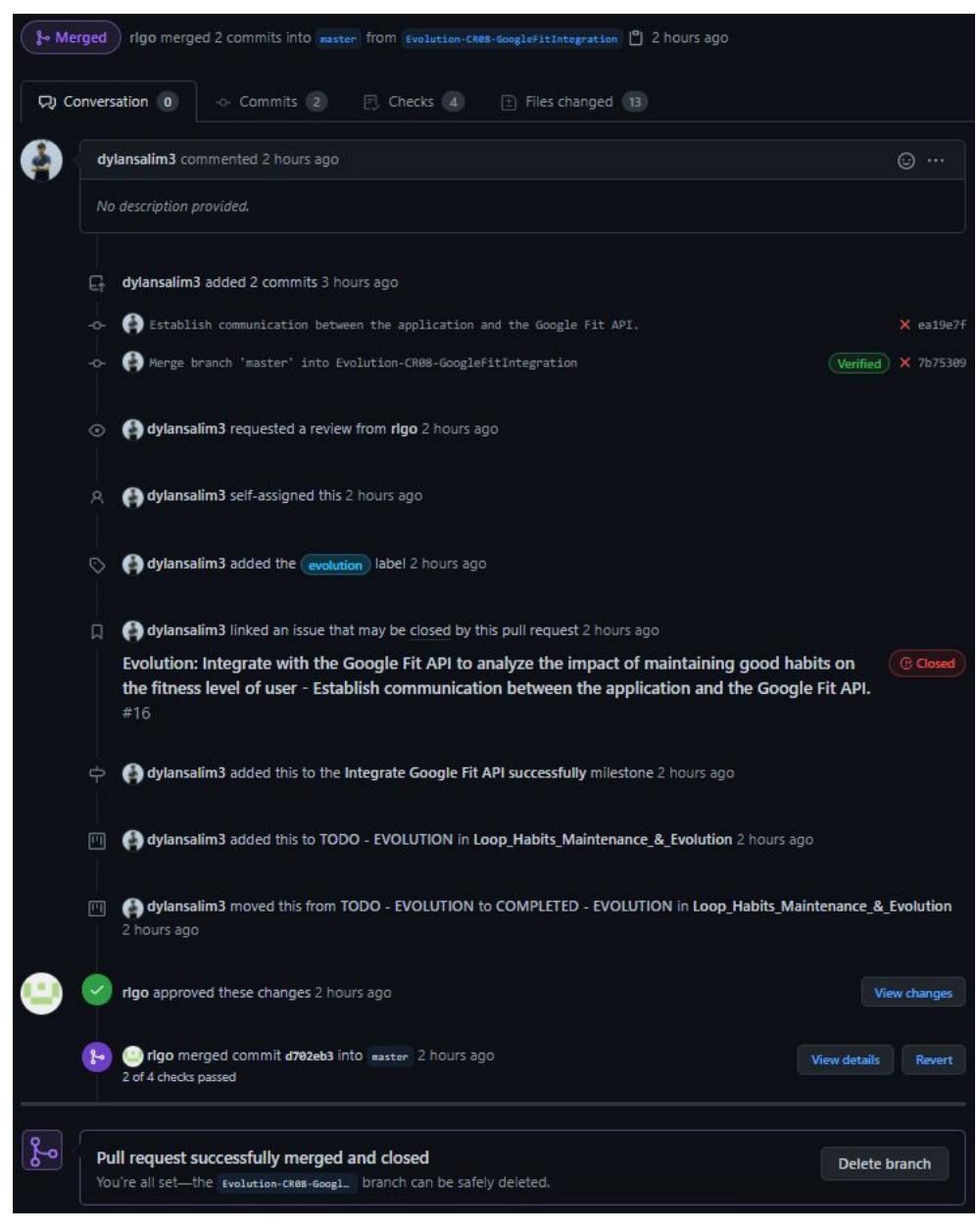
Branch Name	customFieldsBarChart
Change Request(s)	CR09
Collaborator(s)	Ng Jih Yann
Reviewed by	Fang Wai Nam
Approved by	Fang Wai Nam

Screenshot



Branch Name	Evolution-CR08-GoogleFitIntegration
Change Request(s)	CR06, CR07, CR08
Collaborator(s)	Dylan Salim
Reviewed by	Fang Wai Nam
Approved by	Fang Wai Nam

Screenshot



A kanban board is also used to monitor the progress of the project at all time:

[Screenshot Dashboard at project initialization]

[Screenshot Dashboard at project completion] *Both of this wait all ppl added their pull request

4.0 Screen captures of the tool set-up and source code modification.

4.1. Tools Set-Up And Deployment

4.1.1. Development

The application is built up from a combination of Java and Kotlin programming language, therefore, we need to either be using IntelliJ or Android Studio IDE to edit or run the code. There are 2 main packages that we will be using in the development phase, namely uhabits-android which contains all Android-specific code, and uhabits-core which contains the common core libraries resources that are reusable throughout the application.

After cloning the project onto our local machine, we must sync the project with gradle files to resolve the dependencies, it can be done by clicking on the gradle icon in Android studio top right corner as shown below.

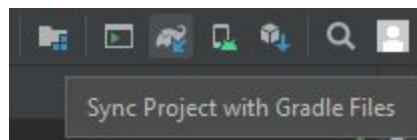


Figure 4.1.1 Sync Project with Gradle Files

This application is working on SQLite database, if we would like to see the changes or the data in the database, we can choose to use any SQLite database browser. If a new migration has to be done to the database, simply go to uhabits-core\src\main\resources\migration, create a new .sql file with the desired migration, and change the DATABASE_VERSION to the name of the sql file in uhabits-core\src\main\org\isoron\uhabits\core\Config.java file.

During the development phase, we are suggested to follow the ktlint coding style when we code for Kotlin. However, there are no specific guidelines in coding Java code. To visualise the change we implemented, we can simply run the code in Android Studio as illustrated below,

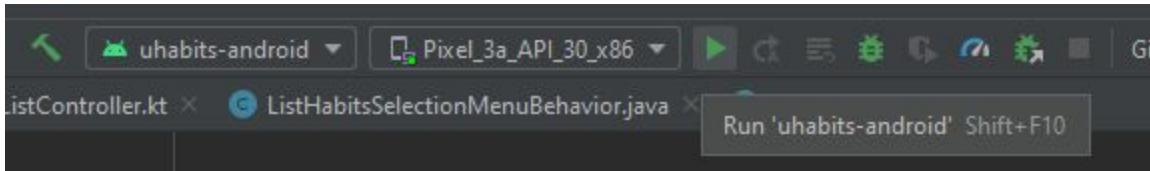


Figure 4.1.2 Run the application on Android Studio

and see it via the emulator provided by Android Studio.

4.1.2 VCS Tool Setup

Version control helps to manage every change we made to the application on whether the maintenance or evolution part. In this project, we are managing version control through Git. Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

A github repository is created with all the group members in the same organisation. The repository is used to manage all the changes made like the code modification and organize and merge every branch. Different branches are then created for each component that will be modified on. Every change made should be committed to their respective branches. The members are required to push and commit every change made together with a proper naming that is clearly explaining the changes for better and easier understanding by others. Then, the branches are used to track back the changes made along.

Below is the list of branches created in the Github repository.

rlgo / **loop_habit_tracker** Private

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Overview Yours Active Stale All branches Search branches...

All branches

Branch	Last updated	Commits	Pull Requests	Status	Actions
master	Updated 5 hours ago by rlgo	12 0	#21 Merged	Default	
Evolution-CR08-GoogleFitIntegration	Updated 5 hours ago by rlgo	11 0	#18 Merged		
Maintenance-CR05-shortcut	Updated 6 hours ago by rlgo	23 0	#9 Merged		
inputFieldsGoogleFit	Updated 15 hours ago by rlgo	27 0	#15 Merged		
modifyDayView	Updated yesterday by eonsein	44 0	#13 Merged		
CICD-codemagicConfiguration	Updated 2 days ago by dylansalim3	48 10	#20 Merged		
customFieldsBarChart	Updated 2 days ago by devilng05108	43 0	#8 Merged		
swipeToDelete	Updated 5 days ago by seoksee	43 0	#4 Merged		
configNotification	Updated 5 days ago by devildarren1997	44 0	#6 Merged		
cancelSwipeToDelete	Updated 6 days ago by seoksee	47 1	New pull request		

4.2. Source Code Modification

4.2.1 Maintenance

CR01

Change Request	CR01
Commit ID	52c3a68a49c5e05bd599cd9c2702eb43b4a8bb08, 27c90794c3639950a961d60e140574da94a83aa3, 27218286a96708a4a7790d09d4a5bbe93d3f05b2, e4d23663b0ce0199845e83f6b1bfe6c11815cdd9
Commit Title	configured Notification, changes made, update, changes made
Description	Solve CR01 to make notification notify the user with sound, vibration and display on lock screen.
Branch	configNotification

Code Modification



The screenshot shows a code editor with the file `AndroidManifest.xml` open. The code is a snippet of XML defining permissions and an application. A new line of code has been added at line 27:

```
1. @@ -24,6 +24,7 @@
24.     <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
25.     <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
26.     <uses-permission android:name="android.permission.INTERNET" />
27. +     <uses-permission android:name="android.permission.ACCESS_NOTIFICATION_POLICY"/>
28.
29. <application
30.     android:name=".HabitsApplication"
```

The line 27 is highlighted in green, indicating it is a new addition. The code editor interface includes a status bar at the top right with "Viewed" and three dots, and a toolbar at the bottom.

```
26 ...uhabits-android/src/main/java/org/isoron/uhabits/notifications/AndroidNotificationTray.kt
```

```
+ @@ -23,6 +23,8 @@ import android.app.*  
23 import android.content.*  
24 import android.graphics.*  
25 import android.graphics.BitmapFactory.*  
26  
27 import android.os.*  
28 import android.os.Build.VERSION.*  
29 import android.util.*  
30  
+ @@ -43,7 +45,7 @@ class AndroidNotificationTray  
31  
32     @AppContext private val context: Context,  
33     private val pendingIntents: PendingIntentFactory,  
34     private val preferences: Preferences,  
35     - private val ringtoneManager: RingtoneManager  
36     ) : NotificationTray.SystemTray {  
37         private var active = HashSet<Int>()  
38  
+ @@ -107,14 +109,18 @@ class AndroidNotificationTray  
39  
40     // WearableExtender.  
41     val wearableExtender = WearableExtender().setBackground(wearableBg)  
42  
43     val defaultText = context.getString(R.string.default_reminder_question)  
44     val builder = Builder(context, REMINDERS_CHANNEL_ID)  
45         .setSmallIcon(R.drawable.ic_notification)  
46         .setContentTitle(habit.name)  
47         .setContentText(if(habit.question.isBlank()) defaultText else habit.question)  
48         .setContentIntent(pendingIntents.showHabit(habit))  
49         .setDeleteIntent(pendingIntents.dismissNotification(habit))  
50  
51     import android.content.*  
52     import android.graphics.*  
53     import android.graphics.BitmapFactory.*  
54     + import android.media.AudioAttributes  
55     + import android.net.Uri  
56     import android.os.*  
57     import android.os.Build.VERSION.*  
58     import android.util.*  
59  
60     @AppContext private val context: Context,  
61     private val pendingIntents: PendingIntentFactory,  
62     private val preferences: Preferences,  
63     + private val ringtoneManager: RingtoneManager  
64     ) : NotificationTray.SystemTray {  
65         private var active = HashSet<Int>()  
66  
67         // WearableExtender.  
68         val wearableExtender = WearableExtender().setBackground(wearableBg)  
69  
70         +  
71             val defaultText = context.getString(R.string.default_reminder_question)  
72             val builder = Builder(context, REMINDERS_CHANNEL_ID)  
73                 .setSmallIcon(R.drawable.ic_notification)  
74                 .setContentTitle(habit.name)  
75                 .setContentText(if(habit.question.isBlank()) defaultText else habit.question)  
76                 .setContentIntent(pendingIntents.showHabit(habit))  
77                 .setDeleteIntent(pendingIntents.dismissNotification(habit))
```

```

v 26 ...uhabits-android/src/main/java/org/isoron/uhabits/notifications/AndroidNotificationTray.kt
174 +         .setContentType(AudioAttributes.CONTENT_TYPE_SONIFICATION)
175 +         .setUsage(AudioAttributes.USAGE_NOTIFICATION)
176 +         .build()
177 +
178 +     channel.setSound(changesRingtone, audioAttributes)
179     notificationManager.createNotificationChannel(channel)
180   }
181 }

v 4 android/uhabits-android/src/main/java/org/isoron/uhabits/notifications/RingtoneManager.kt
53
54     fun getURI(): Uri? {
55         var ringtoneUri: Uri? = null
56         -     val defaultRingtoneUri = Settings.System.DEFAULT_NOTIFICATION_URI
57
58         val prefRingtoneUri = prefs.getString("pref_ringtone_uri",
59             defaultRingtoneUri.toString()!!)
60
61         if (prefRingtoneUri.isNotEmpty())
62             ringtoneUri = Uri.parse(prefRingtoneUri)
63
64     }

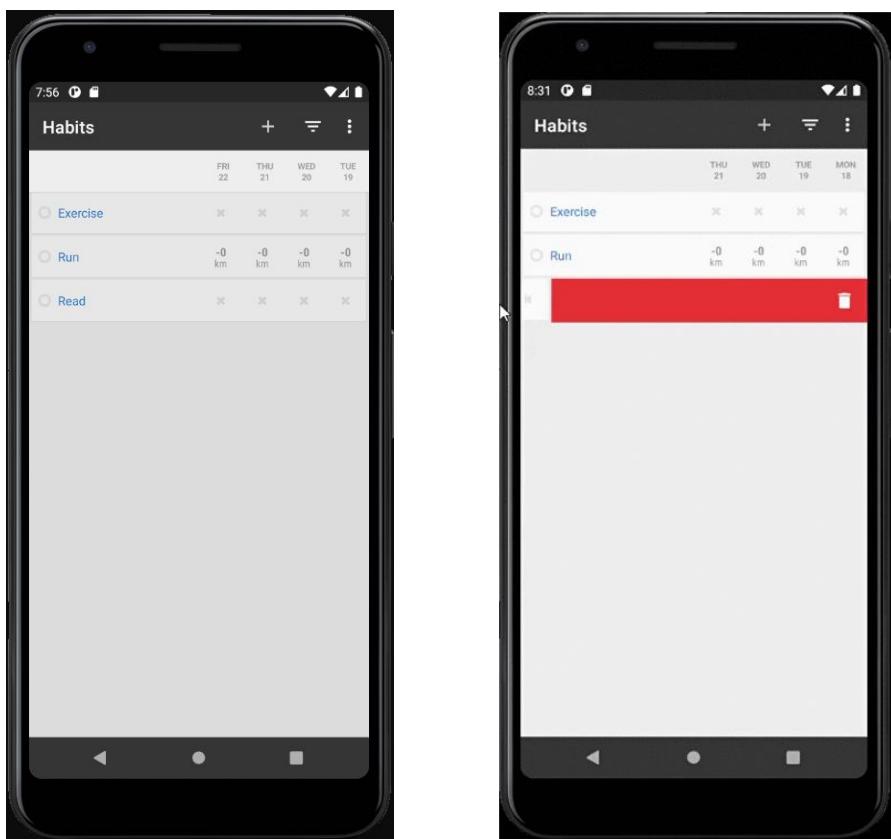
v BIN +102 KB android/uhabits-android/src/main/res/raw/customnotification.mp3
Binary file not shown.

```

CR02

Change Request	CR02
Commit ID	1f00bfd309b2378b1f384669c30c5b43f73cc213, bec0cadd62bb2eea3a875f0a373d89236d7e4a7c, 07d459ab4a18032b148a724e87386032aa4496a8, cbfd18efab5ec7fb8dfba1b386682c52e96fd32d, 4f4a39f3e6fb79217ee9a9fec44852354caf90ca
Commit Title	Added swipe to delete functionality, removed unused files, added red background on swipe, added delete icon upon swiping, clean code
Description	Solve CR02 to add swipe to delete functionality
Branch	swipeToDelete

UI Before and After Change



Before change

After change

Click this link to view the after change gif: <https://imgur.com/3pPMk8i>

Code modification

```
 4 @ -54,6 +56,16 @@ class HabitCardListView( 5   private val touchHelper = ItemTouchHelper(TouchHelperCallback()).apply { 6     attachToRecyclerView(this@habitCardListView) 7   } 8 9   private val touchHelper = ItemTouchHelper(TouchHelperCallback()).apply {10    attachToRecyclerView(this@habitCardListView)11  }12 + val swipeToDeleteCallBack = object : SwipeToDeleteCallBack(){13 +   override fun onSwipe(viewHolder: ViewHolder, direction: Int) {14 +     val position = viewHolder.adapterPosition15 +     controller.get().onItemDelete(position)16 +     adapter.notifyItemRemoved(position)17 +   }18 + }19 +20 + val itemTouchHelper = ItemTouchHelper(swipeToDeleteCallBack).apply {21 +   attachToRecyclerView(this@habitCardListView)22 + }
```

```

160 +
161 +     inner abstract class SwipeToDeleteCallback : ItemTouchHelper.Callback() {
162 +         private val deleteIcon = ContextCompat.getDrawable(context, R.drawable.ic_delete_white_24)
163 +
164 +         private val intrinsicWidth = deleteIcon.intrinsicWidth
165 +         private val intrinsicHeight = deleteIcon.intrinsicHeight
166 +         private val background = ColorDrawable()
167 +         private val backgroundColor = Color.parseColor("#F44336")
168 +
169 +         override fun getMovementFlags(recyclerView: RecyclerView, viewHolder: ViewHolder): Int {
170 +             val swipeFlag = itemTouchHelper.LEFT
171 +             return makeMovementFlags(0, swipeFlag)
172 +         }
173 +
174 +         override fun onMove(recyclerView: RecyclerView, viewHolder: ViewHolder, target: ViewHolder): Boolean {
175 +             return false
176 +         }
177 +
178 +         override fun onChildDraw(c: Canvas, recyclerView: RecyclerView, viewHolder: ViewHolder, dx: Float, dy: Float,
179 +             actionState: Int, isCurrentlyActive: Boolean) {
180 +             val itemView = viewHolder.itemView
181 +             val itemHeight = itemView.bottom - itemView.top
182 +
183 +             //Draw red delete background
184 +             background.color = backgroundColor
185 +             background.setBounds(
186 +                 itemView.right + dx.toInt(),
187 +                 itemView.top,
188 +                 itemView.right,
189 +                 itemView.bottom
190 +             )
191 +             background.draw(c)
192 +
193 +             //Calculate position of delete icon
194 +             val deleteIconTop = itemView.top + (itemHeight - intrinsicHeight)/2
195 +             val deleteIconMargin = (itemHeight - intrinsicHeight) / 2
196 +             val deleteIconLeft = itemView.right - deleteIconMargin - intrinsicWidth
197 +             val deleteIconRight = itemView.right - deleteIconMargin
198 +             val deleteIconBottom = deleteIconTop + intrinsicHeight
199 +
200 +
201 +
202 +
203 +
204 +
205 +
206 +
207 +
208 +
209 +
210 +
211 +
212 +
213 +
214 +
215 +
216 +
217 +
218 +
219 +
220 +
221 +
222 +
223 +
224 +
225 +
226 +
227 +
228 +
229 +
230 +
231 +

```

```

224 +         //Draw delete icon
225 +         deleteIcon.setBounds(deleteIconLeft, deleteIconTop, deleteIconRight, deleteIconBottom)
226 +         deleteIcon.draw(c)
227 +
228 +         super.onChildDraw(c, recyclerView, viewHolder, dx, dy, actionState, isCurrentlyActive)
229 +     }
230 +
231 +     override fun isItemViewSwipeEnabled() = true

```

```

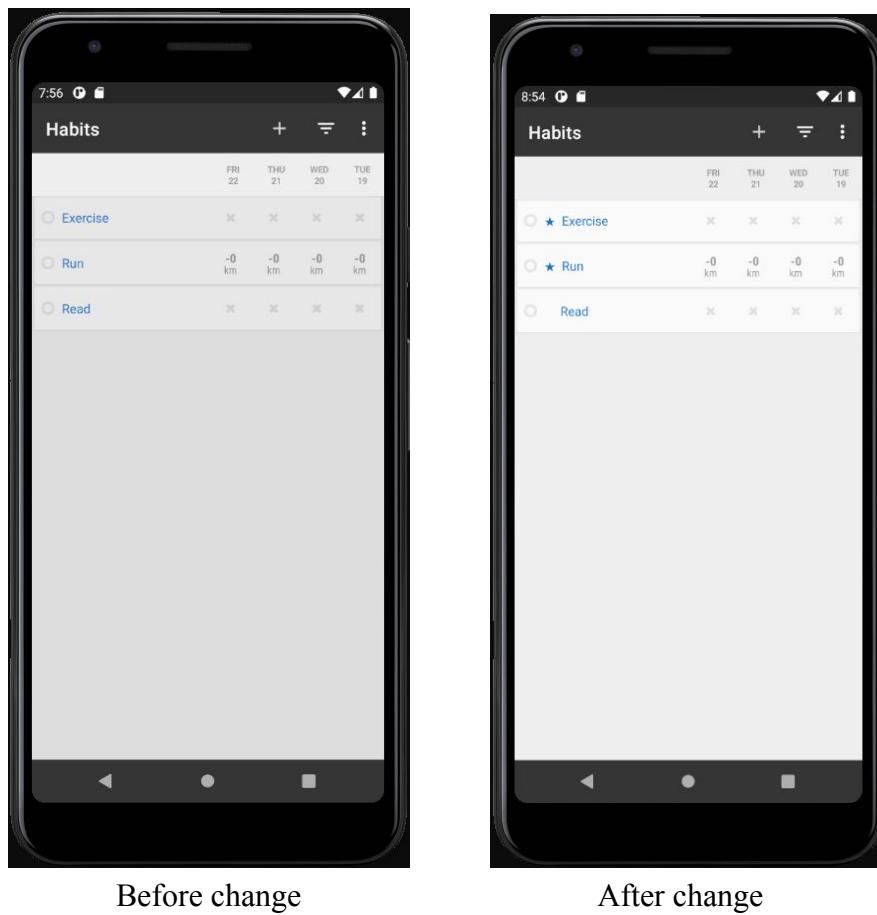
1 + <?xml version="1.0" encoding="utf-8"?>
2 + <vector xmlns:android="http://schemas.android.com/apk/res/android"
3 +     android:width="24dp"
4 +     android:height="24dp"
5 +     android:viewportWidth="24"
6 +     android:viewportHeight="24">
7 +
8 +     <path
9 +         android:fillColor="#FFFFFF"
10 +        android:pathData="M6 19c0 1.1 0.9 2.2 218 0c1.1 0 2 -0.9 2 -2L18 7 6 7 6 19ZM9 4L15.5 4 14.5 3 9.5 3 8.5 4 5 4 5 6 19
11 +        M19 4Z"/>

```

CR03

Change Request	CR03
Commit ID	
Commit Title	update habits table field & added favourite habit functionality, added unfavourite functionality, add star to fav habits
Description	Solve CR03 to allow users to favourite one or more habit
Branch	favouriteHabit

UI Before and After Change



Code Modification

```
v 14 ... android/src/main/java/org/isaron/uhabits/activities/habits/list/ListHabitsSelectionMenu.kt

+ @@ -64,6 +64,16 @@ class ListHabitsSelectionMenu @Inject constructor(
64     return true
65   }
66
67   +   R.id.action_favourite_habit -> {
68   +     behavior.onFavouriteHabits()
69   +     return true
70   +   }
71
72   +   R.id.action_unfavourite_habit -> {
73   +     behavior.onUnfavouriteHabits()
74   +     return true
75   +   }
76   +
```

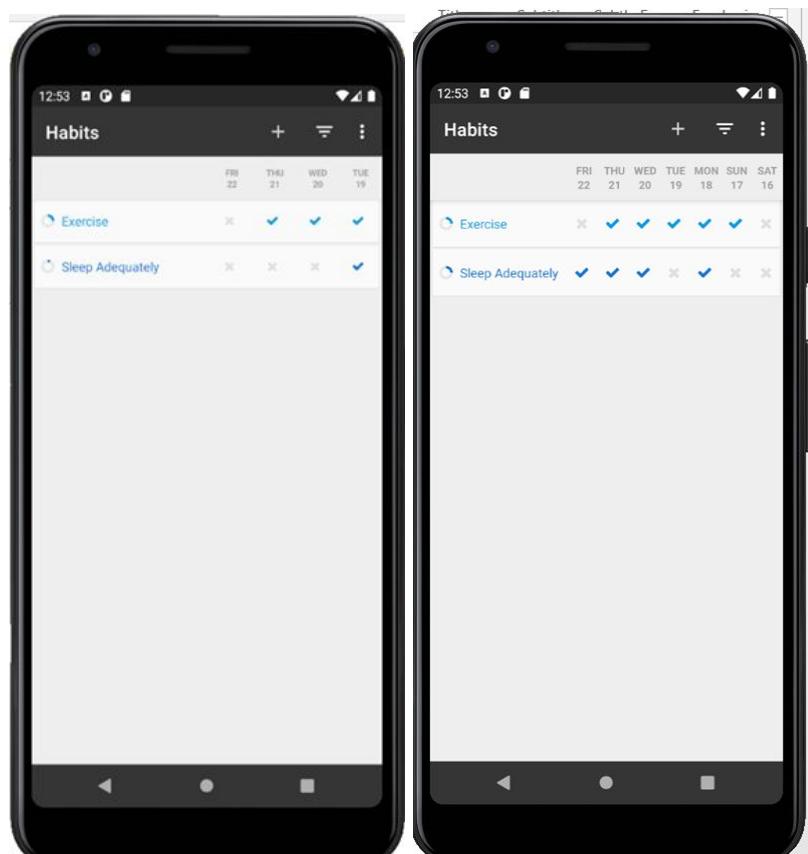


```
+ @@ -89,12 +99,16 @@ class ListHabitsSelectionMenu @Inject constructor(
89   val itemColor = menu.findItem(R.id.action_color)
90   val itemArchive = menu.findItem(R.id.action_archive_habit)
91   val itemUnarchive = menu.findItem(R.id.action_unarchive_habit)
92
93   val itemNotify = menu.findItem(R.id.action_notify)
94
95   itemColor.isVisible = true
96   itemEdit.isVisible = behavior.canEdit()
97   itemArchive.isVisible = behavior.canArchive()
98   itemUnarchive.isVisible = behavior.canUnarchive()
99
100  setTitle(Integer.toString(listAdapter.selected.size))
101  itemNotify.isVisible = prefs.isDeveloper
102
103  +   val itemColor = menu.findItem(R.id.action_color)
104  +   val itemArchive = menu.findItem(R.id.action_archive_habit)
105  +   val itemUnarchive = menu.findItem(R.id.action_unarchive_habit)
106  +   val itemFavourite = menu.findItem(R.id.action_favourite_habit)
107  +   val itemUnfavourite = menu.findItem(R.id.action_unfavourite_habit)
108  +   val itemNotify = menu.findItem(R.id.action_notify)
109
110  +   itemColor.isVisible = true
111  +   itemEdit.isVisible = behavior.canEdit()
112  +   itemArchive.isVisible = behavior.canArchive()
113  +   itemUnarchive.isVisible = behavior.canUnarchive()
114  +   itemFavourite.isVisible = behavior.canFavourite()
115  +   itemUnfavourite.isVisible = behavior.canUnfavourite()
116  +   setTitle(Integer.toString(listAdapter.selected.size))
117  +   itemNotify.isVisible = prefs.isDeveloper
118
```

CR04

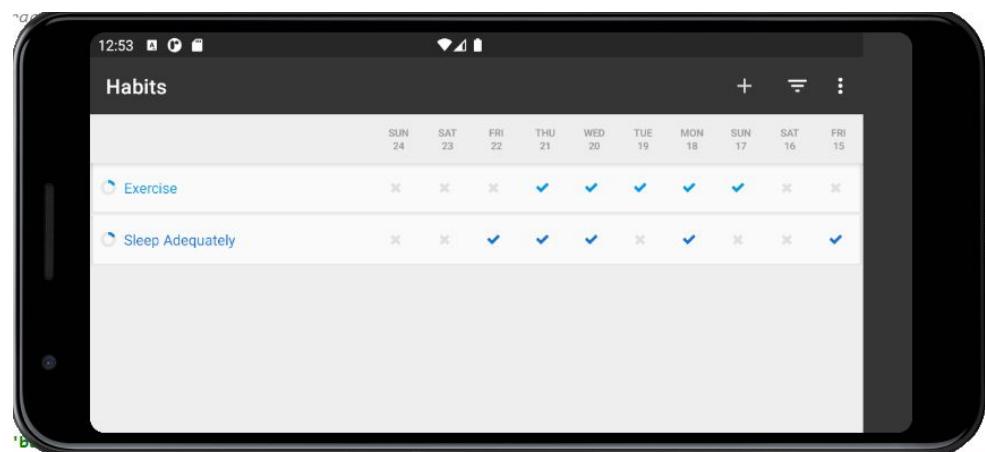
Change Request	CR04
Commit ID	9618408eb5d9f59259f3aeced9562a5a4e466a67, 2a72f97bd825cb524ea4686070fd563a30315628
Commit Title	changeDateview, minorChange
Description	Solve CR04 to allow users to monitor the habits every week easily by modifying the day view to 7 days or 14 days without scrolling to another page.
Branch	modifyDayView

UI Before and After Change

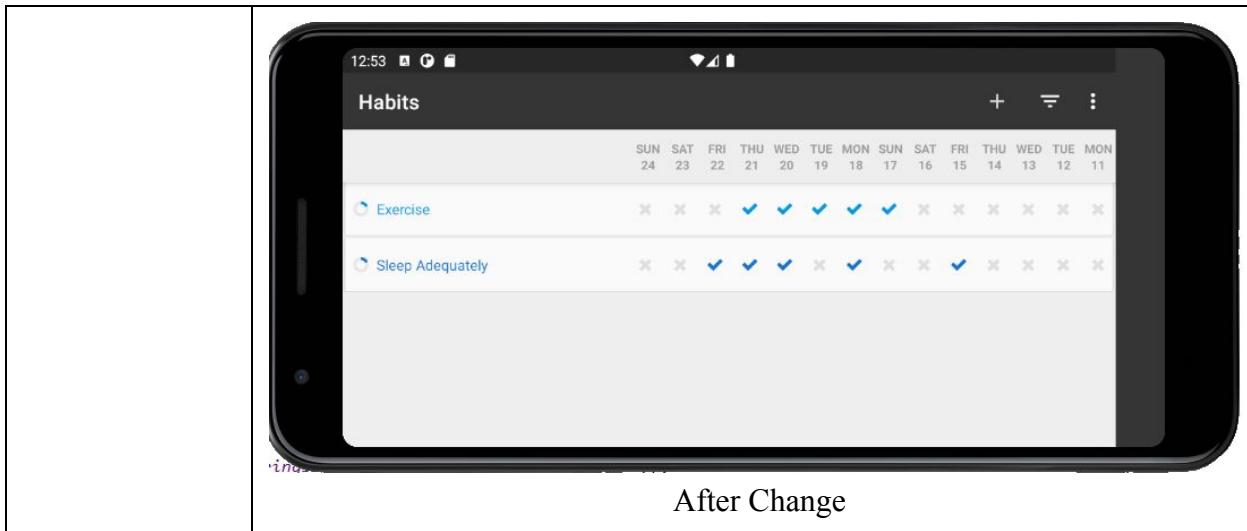


Before change

After change



Before Change



Code Modification

```

14  android/uhabits-android/src/main/res/values/dimens.xml
19
20      <resources>
21          + <dimen name="baseSize">20dp</dimen>
22          - <dimen name="checkmarkWidth">48dp</dimen>
23          - <dimen name="checkmarkHeight">48dp</dimen>
24          <dimen name="history_editor_max_height">450dp</dimen>
25          <dimen name="history_editor_padding">8dp</dimen>
26          - <dimen name="regularTextSize">16sp</dimen>
27          - <dimen name="smallTextSize">14sp</dimen>
28          - <dimen name="smallerTextSize">12sp</dimen>
29          - <dimen name="tinyTextSize">10sp</dimen>
30          <dimen name="habitNameWidth">160dp</dimen>
31          <dimen name="checkmarkWidget_heightBreakpoint">55dp</dimen>
32      </resources> ⊖
19
20      <resources>
21          <dimen name="baseSize">20dp</dimen>
22          + <dimen name="checkmarkWidth">34dp</dimen>
23          + <dimen name="checkmarkHeight">52dp</dimen>
24          <dimen name="history_editor_max_height">450dp</dimen>
25          <dimen name="history_editor_padding">8dp</dimen>
26          + <dimen name="regularTextSize">18sp</dimen>
27          + <dimen name="smallTextSize">18sp</dimen>
28          + <dimen name="smallerTextSize">14sp</dimen>
29          + <dimen name="tinyTextSize">12sp</dimen>
30          + <dimen name="habitNameWidth">120dp</dimen>
31          <dimen name="checkmarkWidget_heightBreakpoint">55dp</dimen>
32      </resources> ⊖

```

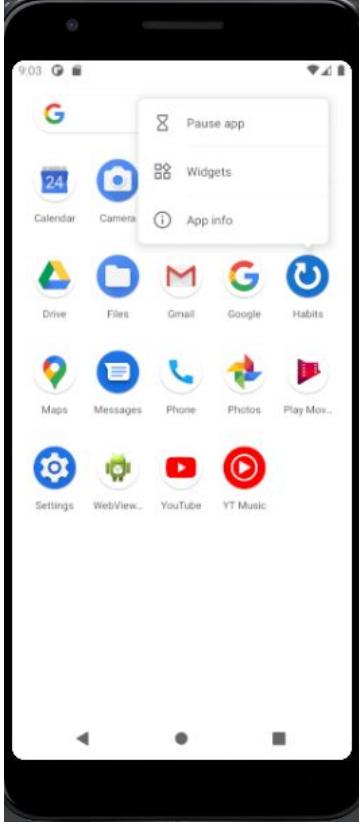
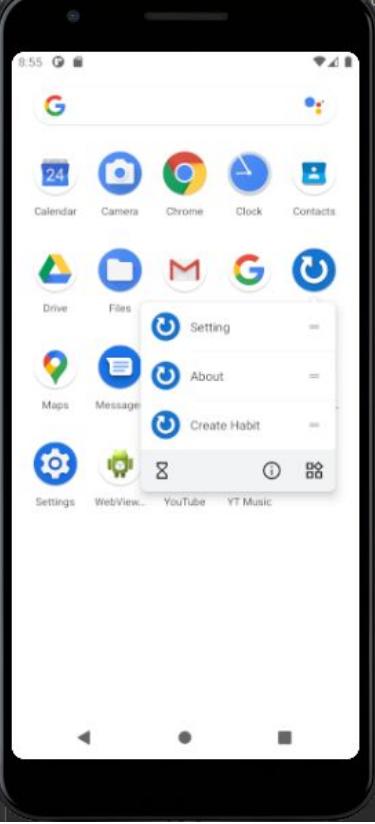
```

4  ...habits-android/src/main/java/org/isoron/uhabits/activities/habits/list/views/HeaderView.kt
50
51
52     init {
53         - setScrollerBucketSize(dim(R.dimen.checkmarkWidth).toInt())
54         setBackgroundColor(res.getColor(R.attr.headerBackgroundColor))
55         - if (SDK_INT >= LOLLIPOP) elevation = dp(2.0f)
56     }
57
58     override fun atMidnight() {
59
50         }
51
52     init {
53         + setScrollerBucketSize(dim(R.dimen.scrollerBucketSize).toInt())
54         setBackgroundColor(res.getColor(R.attr.headerBackgroundColor))
55         + if (SDK_INT >= LOLLIPOP) elevation = dp(5.0f)
56     }
57
58     override fun atMidnight() {

```

CR05

Change Request	CR05
Commit ID	31829557fa54690e823d50378a2d64504d640dfa
Commit Title	add shortcut

Description	Solve CR05 to add quick shortcut to the application
Branch	shortcut
UI Before Change	 <p>Before Change</p>  <p>After Change</p>



After Change

Code Modification

android/uhabits-android/src/main/res/xml/shortcut.xml

```

1 + <?xml version="1.0" encoding="utf-8"?>
2 + <shortcuts xmlns:android="http://schemas.android.com/apk/res/android">
3 +     <shortcut
4 +         android:shortcutId="ActivityOne"
5 +         android:enabled="true"
6 +         android:icon="@mipmap/ic_launcher"
7 +         android:shortcutShortLabel="@string/short_label"
8 +         android:shortcutLongLabel="@string/long_label"
9 +         android:shortcutDisabledMessage="@string/disabled_message"
10 +        >
11 +            <!-- Set Intent and Category -->
12 +            <intent
13 +                android:action="android.intent.action.VIEW"
14 +                android:targetPackage="org.isoron.uhabits"
15 +
16 +                android:targetClass="org.isoron.uhabits.activities.settings.SettingsActivity"
17 +            />
18 +            <category android:name="android.shortcut.conversation"/>
19 +
20 +        </shortcut>
21 +        <shortcut
22 +            android:shortcutId="ActivityThree"
23 +            android:enabled="true"
24 +            android:icon="@mipmap/ic_launcher"
25 +            android:shortcutShortLabel="@string/short_label_three"
26 +            android:shortcutLongLabel="@string/long_label_three"
27 +            android:shortcutDisabledMessage="@string/disabled_message_three"
28 +            >
29 +                <!-- Set Intent and Category -->
30 +                <intent
31 +                    android:action="android.intent.action.VIEW"
32 +                    android:targetPackage="org.isoron.uhabits"
33 +                    android:targetClass="org.isoron.uhabits.activities.about.AboutActivity"
34 +                />
35 +                <category android:name="android.shortcut.conversation"/>
36 +
37 +            </shortcut>
38 +            <shortcut
39 +                android:shortcutId="ActivityTwo"
40 +                android:enabled="true"
41 +                android:icon="@mipmap/ic_launcher"
42 +                android:shortcutShortLabel="@string/short_label_two"
43 +                android:shortcutLongLabel="@string/long_label_two"
44 +                android:shortcutDisabledMessage="@string/disabled_message_two"
45 +                >
46 +                    <!-- Set Intent and Category -->
47 +                    <intent
48 +                        android:action="android.intent.action.VIEW"
49 +                        android:targetPackage="org.isoron.uhabits"
50 +
51 +                        android:targetClass="org.isoron.uhabits.activities.habits.edit.EditHabitActivity"
52 +                    />
53 +                    <category android:name="android.shortcut.conversation"/>
54 +
55 +            </shortcut>
56 +
57 +        </shortcuts> ⚡

```

android/uhabits-android/src/main/AndroidManifest.xml

```

+ @@ -78,8 +64,11 @@
78     android:targetActivity=".activities.habits.list.ListHabitsActivity"
79     <intent-filter android:label="@string/main_activity_title">
80         <action android:name="android.intent.action.MAIN" />
81         <category android:name="android.intent.category.LAUNCHER" />
82     </intent-filter>
83     </activity-alias>
84
85     <activity
64             android:targetActivity=".activities.habits.list.ListHabitsActivity"
65             <intent-filter android:label="@string/main_activity_title">
66                 <action android:name="android.intent.action.MAIN" />
67             +
68             <category android:name="android.intent.category.LAUNCHER" />
69         </intent-filter>
70         +
71         <meta-data android:name="android.app.shortcuts"
72             android:resource="@xml/shortcut"/>
73     </activity-alias>
74     <activity

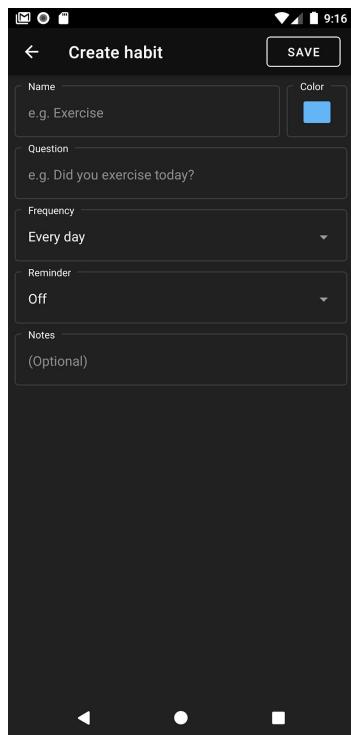
```

4.2.2 Evolution

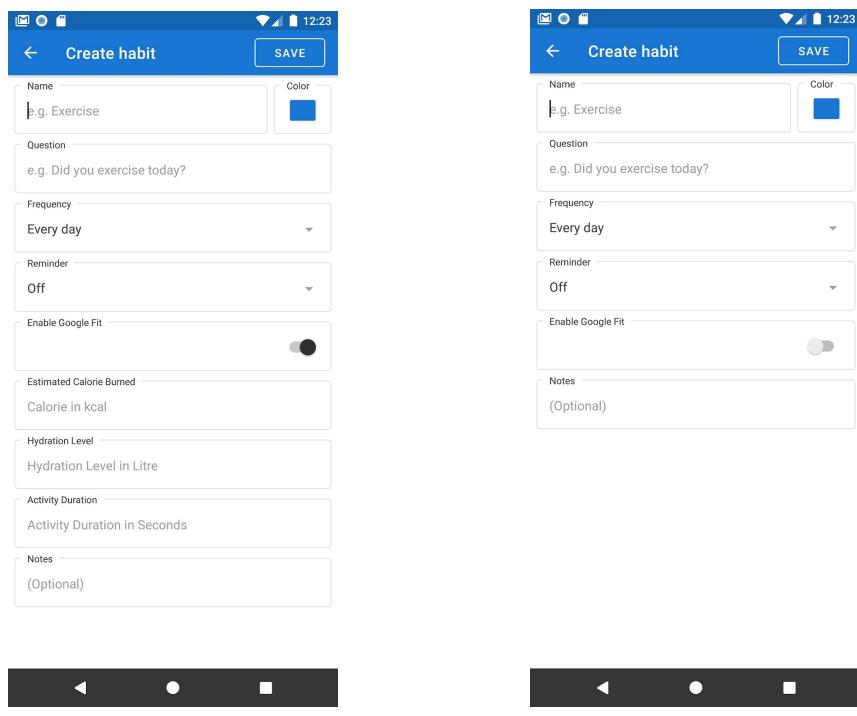
CR07

Change Request	CR07
Pull Request	Added input fields for Google Fit Api & Update habits database fields #9 Open dylansalim3 wants to merge 1 commit into master from inputFieldsGoogleFit
Commit ID	a7cb95c4e2d1540b37cb89d4ee2b560f4955d1c6
Commit Title	Added new input fields for custom calorie field Added new input fields for custom hydration field Added new input fields for custom activity duration field Add database field to store calorie, hydration and activity duration fields
Description	Create Google Fit API fields and store all the fields to local database
Branch	inputFieldsGoogleFit

UI Before Change



UI After Change



Code Modification

The screenshot shows a code editor with two tabs open. The top tab, 'EditHabitActivity.kt', displays Java code for an AppCompatActivity. The bottom tab, 'migrations/25.sql', displays SQL migration code for a database table.

EditHabitActivity.kt (Java Code)

```
149 181     else -> 1
150 182
151 183     @@ -187,8 +219,23 @@ class EditHabitActivity : AppCompatActivity() {
152 184         dialog.show(supportFragmentManager, "dayPicker")
153 185     }
154 186
155 187     binding.fitSwitchInput.setOnCheckedChangeListener { _, isChecked ->
156 188         run {
157 189             enableGoogleFit = isChecked
158 190             if (isChecked) {
159 191                 f1_calorie_burned.visibility = View.VISIBLE
160 192                 f1_hydration.visibility = View.VISIBLE
161 193                 f1_activity_duration.visibility = View.VISIBLE
162 194             } else {
163 195                 f1_calorie_burned.visibility = View.GONE
164 196                 f1_hydration.visibility = View.GONE
165 197                 f1_activity_duration.visibility = View.GONE
166 198             }
167 199         }
170 200     }
171 201
172 202     binding.buttonSave.setOnClickListener {
173 203         if(validate()) save()
174 204         if(validate()) save()
175 205     }
176 206
177 207     for (fragment in supportFragmentManager.fragments) {
178 208         @@ -223,6 +270,10 @@ class EditHabitActivity : AppCompatActivity() {
179 209             habit.unit = unitInput.text.trim().toString()
180 210         }
181 211         habit.type = habitType
182 212         habit.enableGoogleFit = enableGoogleFit
183 213
184 214         if (enableGoogleFit) {
185 215             habit.calorieBurned = calorieBurnedInput.text.toString().toDoubleOrNull()
186 216             ?: 0.0
187 217             habit.hydration = hydrationInput.text.toString().toDoubleOrNull()
188 218             ?: 0.0
189 219             habit.activityDuration = activityDurationInput.text.toString().toDoubleOrNull() ?: 0.0
190 220         } else {
191 221             habit.calorieBurned = 0.0
192 222             habit.hydration = 0.0
193 223             habit.activityDuration = 0.0
194 224         }
195 225     }
196 226 }
```

migrations/25.sql (SQL Migration)

```
1  ✓ alter table habits add column enable_google_fit INTEGER;
2  ✓ alter table habits add column calorie_burned TEXT;
3  ✓ alter table habits add column hydration TEXT;
4  ✓ alter table habits add column activity_duration REAL;
5  ✓ alter table habits add column favourite Integer default 0; ⏺
```

```

v 68 android/uhabits-android/src/main/res/layout/activity_edit_habit.xml [ ]
211 212             </LinearLayout>
212 213         </FrameLayout>
213 214
215 + <!--      Enable Google Fit-->
216 +
217 +     <FrameLayout style="@style/FormOuterBox">
218 +         <LinearLayout style="@style/FormInnerBox">
219 +             <TextView
220 +                 style="@style/FormLabel"
221 +                 android:text="@string/enable_google_fit" />
222 +             <Switch
223 +                 android:id="@+id/fitSwitchInput"
224 +                 style="@style/FormInput"
225 +                 android:layout_height="wrap_content"
226 +                 android:layout_width="match_parent"/>
227 +         </LinearLayout>
228 +     </FrameLayout>
229 +
230 + <!-- Calorie Burned -->
231 +
232 +     <FrameLayout
233 +         style="@style/FormOuterBox"
234 +         android:id="@+id/fl_calorie_burned"
235 +         android:visibility="gone">
236 +         <LinearLayout style="@style/FormInnerBox">
237 +             <TextView
238 +                 style="@style/FormLabel"
239 +                 android:text="@string/estimated_calorie_burned" />
240 +             <EditText
241 +                 style="@style/FormInput"
242 +                 android:id="@+id/calorieBurnedInput"
243 +                 android:maxLines="1"
244 +                 android:ems="10"
245 +                 android:hint="@string/calorie_in_kcal"/>
246 +         </LinearLayout>
247 +     </FrameLayout>
248 +
249 + <!--      Hydration -->
250 +
251 +     <FrameLayout
252 +         style="@style/FormOuterBox"
253 +         android:id="@+id/fl_hydration"
254 +         android:visibility="gone">
255 +         <LinearLayout style="@style/FormInnerBox">
256 +             <TextView
257 +                 style="@style/FormLabel"
258 +                 android:text="@string/hydration_level" />
259 +             <EditText
260 +                 style="@style/FormInput"
261 +                 android:id="@+id/hydrationInput"
262 +                 android:maxLines="1"
263 +                 android:ems="10"
264 +                 android:hint="@string/hydration_level_in_litre"/>
265 +         </LinearLayout>
266 +     </FrameLayout>

```

CR08

Change Request	CR08
Pull Request	<p>Establish communication between the application and the Google Fit API. #17</p> <p>Merged rigo merged 2 commits into master from Evolution-CR08-GoogleFitIntegration 11 hours ago</p>
Commit ID	ea19e7f8f0dde35c506a4b799a32e45b8e03fb5f
Commit Title	Establish communication between the application and the Google Fit API.
Description	Intergration with Google Fit Api

Branch

Evolution-CR08-GoogleFitIntegration

Code Modification

```
 180 └── android/uhabits-android/src/main/java/org/isoron/uhabits/utils/GoogleFitUtils.kt
 27 + class GoogleFitUtils(val context: Context) {
 28 +
 29 +     private val fitRequestCode = 3792
 30 +     private val fitRequestCode2 = 3793
 31 +     private val tag = "Google Fit Utils"
 32 +
 33 +     private val fitnessOptions = FitnessOptions.builder()
 34 +         .addDataType(DataType.TYPE_ACTIVITY_SEGMENT, FitnessOptions.ACCESS_WRITE)
 35 +         .addDataType(DataType.TYPE_CALORIES_EXPENDED, FitnessOptions.ACCESS_WRITE)
 36 +         .addDataType(DataType.TYPE_HYDRATION, FitnessOptions.ACCESS_WRITE)
 37 +         .addDataType(DataType.TYPE_WEIGHT, FitnessOptions.ACCESS_WRITE)
 38 +         .build()
 39 +
 40 +     private val account = GoogleSignIn.getAccountForExtension(context, fitnessOptions)
 41 +
 42 +     fun processHabit(habit: Habit, value: Int, timestamp: Timestamp) {
 43 +         if (!habit.enableGoogleFit) return
 44 +
 45 +         val duration = habit.activityDuration.toLong()
 46 +
 47 +         if (value == YES_MANUAL) {
 48 +             val hydration = habit.hydration.toFloat()
 49 +             val calorie = habit.calorieBurned.toFloat()
 50 +             insertSession(habit.name, hydration, calorie, duration, timestamp)
 51 +         } else if (value == NO)
 52 +             deleteSession(duration, timestamp)
 53 +     }
 54 +
 55 +     fun processNumericHabit(habit: Habit, value: Double, timestamp: Timestamp) {
 56 +         if (!habit.enableGoogleFit) return
 57 +
 58 +         val duration = habit.activityDuration.toLong()
 59 +
 60 +         if (value > 0) {
 61 +             val hydration = habit.hydration.toFloat()
 62 +             val calorie = habit.calorieBurned.toFloat()
 63 +             insertSession(habit.name, hydration, calorie, duration, timestamp)
 64 +         } else if (value == 0.0)
 65 +             deleteSession(duration, timestamp)
 66 +     }
 67 + }
```

```
68 +     // duration: seconds | calorie: kcal | volume: litre
69 +     private fun insertSession(name: String, volume: Float? = null, calorie: Float? = null,
70 +                               duration: Long, timestamp: Timestamp) {
71 +
72 +         val end = timestamp.toJavaDate().toInstant().atZone(ZoneId.systemDefault())
73 +         val start = end.minusSeconds(duration)
74 +
75 +         val session = Session.Builder()
76 +             .setName(name)
77 +             .setDescription("loop habits tracker")
78 +             .setIdentifier(UUID.randomUUID().toString())
79 +             .setActivity(FitnessActivities.RUNNING)
80 +             .setStartTime(start.toEpochSecond(), TimeUnit.SECONDS)
81 +             .setEndTime(end.toEpochSecond(), TimeUnit.SECONDS)
82 +             .build()
83 +
84 +         val insertRequest = SessionInsertRequest.Builder()
85 +             .setSession(session)
86 +
87 +         if (calorie != null) {
88 +             val calorieSource = DataSource.Builder()
89 +                 .setAppPackageName(context)
90 +                 .setDataType(DataType.TYPE_CALORIES_EXPENDED)
91 +                 .setType(DataSource.TYPE_RAW)
92 +                 .build()
93 +
94 +             val caloriePoint = DataPoint.builder(calorieSource)
95 +                 .setField(Field.FIELD_CALORIES, calorie)
96 +                 .setTimeInterval(start.toEpochSecond(), end.toEpochSecond(), TimeUnit.SECONDS)
97 +                 .build()
98 +
99 +             val calorieSet = DataSet.builder(calorieSource)
100 +                 .add(caloriePoint)
101 +                 .build()
102 +
103 +             insertRequest.addDataSet(calorieSet)
104 +         }
105 +     }
```

```

106 +         if (volume != null) {
107 +             val hydrationSource = DataSource.Builder()
108 +                 .setAppPackageName(context)
109 +                 .setDataType(DataType.TYPE_HYDRATION)
110 +                 .setType(DataSource.TYPE_RAW)
111 +                 .build()
112 +
113 +             val hydrationPoint = DataPoint.builder(hydrationSource)
114 +                 .setField(Field.FIELD_VOLUME, volume)
115 +                 .setTimestamp(end.toEpochSecond(), TimeUnit.SECONDS)
116 +                 .build()
117 +
118 +             val hydrationSet = DataSet.builder(hydrationSource)
119 +                 .add(hydrationPoint)
120 +                 .build()
121 +
122 +             insertRequest.addDataSet(hydrationSet)
123 +
124 +
125 +             Fitness.getSessionsClient(context, account)
126 +                 .insertSession(insertRequest.build())
127 +                 .addOnSuccessListener { Log.i(tag, "session inserted") }
128 +                 .addOnFailureListener { error ->
129 +                     Log.e(tag, "session insert error $error")
130 +                 }
131 +
132 +
133 +             fun deleteSession(duration: Long, timestamp: Timestamp) {
134 +                 val end = timestamp.toJavaDate().toInstant().atZone(ZoneId.systemDefault())
135 +                 val start = end.minusSeconds(duration)
136 +
137 +                 val deleteRequest = DataDeleteRequest.Builder()
138 +                     .setTimeInterval(start.toEpochSecond(), end.toEpochSecond(), TimeUnit.SECONDS)
139 +                     .deleteAllData()
140 +                     .deleteAllSessions()
141 +                     .build()

```

```

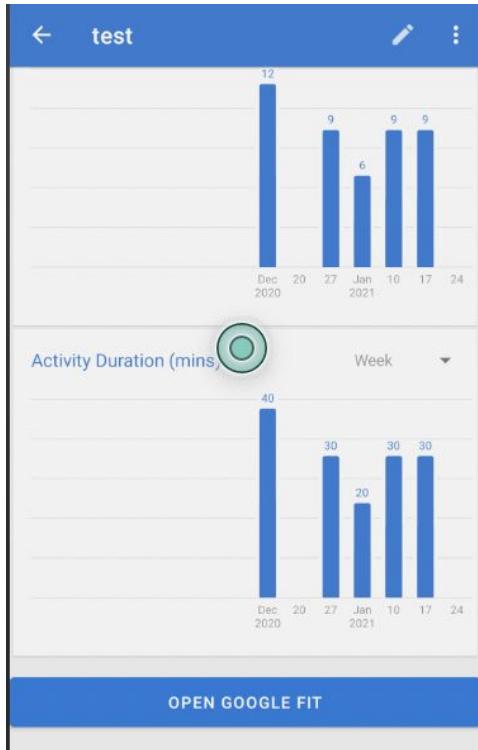
142 +
143 +     Fitness.getHistoryClient(context, account)
144 +         .deleteData(deleteRequest)
145 +             .addOnSuccessListener { Log.i(tag, "session deleted") }
146 +             .addOnFailureListener { error ->
147 +                 Log.e(tag, "delete session failed $error")
148 +             }
149 +     }
150 +
151 +     fun getPermission(activity: Activity) {
152 +         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.Q) {
153 +             if (ContextCompat.checkSelfPermission(activity, Manifest.permission.ACTIVITY_RECOGNITION)
154 +                 != PackageManager.PERMISSION_GRANTED) {
155 +                 ActivityCompat.requestPermissions(activity,
156 +                     arrayOf(Manifest.permission.ACTIVITY_RECOGNITION),
157 +                     fitrequestCode2)
158 +             }
159 +         }
160 +         if (!GoogleSignIn.hasPermissions(account, fitnessOptions)) {
161 +             GoogleSignIn.requestPermissions(
162 +                 activity,
163 +                 fitrequestCode,
164 +                 account,
165 +                 fitnessOptions)
166 +             Log.i(tag, "no permissions")
167 +         } else {
168 +             Log.i(tag, "permissions granted")
169 +         }
170 +     }
171 +
172 +     fun onPermissionsResult(activity: Activity, requestCode: Int, resultCode: Int, data: Intent?) {
173 +         if (requestCode == fitrequestCode && resultCode != Activity.RESULT_OK) {
174 +             Toast.makeText(activity, "Permission not granted", Toast.LENGTH_LONG).show()
175 +         }
176 +         if (requestCode == fitrequestCode2 && resultCode != Activity.RESULT_OK) {
177 +             Toast.makeText(activity, "Permission not granted", Toast.LENGTH_LONG).show()
178 +         }
179 +     }

```

CR09

Change Request	CR09
Pull Request	Additional habit data bar chart, app linking with Google Fit app  devling05108 wants to merge 5 commits into master from customFieldsBarChart
Commit ID	B4b887003899cb5993f40569cd47d3d1b78ec232,

	2bca149ad8d6b51043f49df4628621981efa55a0, 6a9caa421d89cf4814cb9982fd702684edb9dc87 55965270d7da821e9c9b92a615338c611bb54e33
Commit Title	Added new bar chart for custom calorie field Added new bar chart for custom hydration field Added new bar chart for custom activity duration field Provide functionality for users to be redirected to Google Fit app
Description	Visualize the additional habit data in bar charts and create a button for users to open Google Fit app
Branch	customFieldBarChart
UI Before and After Change	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Before Change</p> </div> <div style="text-align: center;"> <p>After Change</p> </div> </div>



After Change

Code Modification for CR-09

```
✓ 89 .../uhabits-android/src/main/java/org/isoron/uhabits/activities/habits/show>ShowHabitView.kt ...
...
... @@ -1,5 +1,5 @@
 1   1   /*
 2 -   * Copyright (C) 2016-2020 Alinson Santos Xavier <isoron@gmail.com>
 2 +   * Copyright (C) 2016-2020 Linson Santos Xavier <isoron@gmail.com>
 3   3   *
 4   4   * This file is part of Loop Habit Tracker.
 5   5   *
@@ -20,8 +20,11 @@
20  20 package org.isoron.uhabits.activities.habits.show
21  21
22  22 import android.content.*
23 + import android.util.Log
24  24 import android.view.*
25  25 import android.widget.*
26 + import kotlinx.android.synthetic.main.show_habit.view.*
27 + import org.isoron.uhabits.R
28  28 import org.isoron.uhabits.activities.habits.show.views.*
29  29 import org.isoron.uhabits.core.models.*
30  30 import org.isoron.uhabits.core.preferences.*
@@ -41,25 +44,40 @@
41  44     val frequency: FrequencyCardViewModel,
42  45     val history: HistoryCardViewModel,
43  46     val bar: BarCardViewModel,
47 +     val calorieBar: CalorieBarCardViewModel,
48 +     val hydrationBar: HydrationBarCardViewModel,
49 +     val activitydurationBar: ActivitydurationBarCardViewModel,
50  50 )
51  51
46 - class ShowHabitView(context: Context) : FrameLayout(context) {
52 + class ShowHabitView(context: Context, habits: Habit) : FrameLayout(context) {
47  53     private val binding = ShowHabitBinding.inflate(LayoutInflater.from(context))
48  54
49  55     var onScoreCardSpinnerPosition: (position: Int) -> Unit = {}
50  56     var onClickEditHistoryButton: () -> Unit = {}
51  57     var onBarCardBoolSpinnerPosition: (position: Int) -> Unit = {}
52  58     var onBarCardNumericalSpinnerPosition: (position: Int) -> Unit = {}
59 +     var onCalorieBoolSpinnerPosition: (position: Int) -> Unit = {}
60 +     var onCalorieNumericalSpinnerPosition: (position: Int) -> Unit = {}
61 +     var onHydrationBoolSpinnerPosition: (position: Int) -> Unit = {}
62 +     var onHydrationNumericalSpinnerPosition: (position: Int) -> Unit = {}
63 +     var onActivitydurationBoolSpinnerPosition: (position: Int) -> Unit = {}
64 +     var onActivitydurationNumericalSpinnerPosition: (position: Int) -> Unit = {}
53  65
54  66     init {
55  67         addView(binding.root)
56  68         binding.scoreCard.onSpinnerPosition = { onScoreCardSpinnerPosition(it) }
57  69         binding.historyCard.onClickEditButton = { onClickEditHistoryButton() }
58  70         binding.barCard.onBoolSpinnerPosition = { onBarCardBoolSpinnerPosition(it) }
59  71         binding.barCard.onNumericalSpinnerPosition = { onBarCardNumericalSpinnerPosition(it) }
72 +         binding.calorieBarCard.onCalorieBoolSpinnerPosition = { onCalorieBoolSpinnerPosition(it) }
73 +         binding.calorieBarCard.onCalorieNumericalSpinnerPosition = { onCalorieNumericalSpinnerPosition(it) }
74 +         binding.hydrationBarCard.onHydrationBoolSpinnerPosition = { onHydrationBoolSpinnerPosition(it) }
75 +         binding.hydrationBarCard.onHydrationNumericalSpinnerPosition = { onHydrationNumericalSpinnerPosition(it) }
76 +         binding.activitydurationBarCard.onActivitydurationBoolSpinnerPosition = { onActivitydurationBoolSpinnerPosition(it) }
```

```

77  +     binding.activitydurationBarCard.onActivitydurationNumericalSpinnerPosition = { onActivitydurationNumericalSpinnerPosition(it) }
60  78      }
61  79
62  -     fun update(data: ShowHabitViewModel) {
63  +     fun update(data: ShowHabitViewModel, habit: Habit) {
64  +         setupToolbar(binding.toolbar, title = data.title, color = data.color)
65  +         binding.subtitleCard.update(data.subtitle)
66  +         binding.overviewCard.update(data.overview)
67  +         @@ -70,6 +88,40 @@
68  +             class ShowHabitView(context: Context) : FrameLayout(context) {
69  +                 binding.frequencyCard.update(data.frequency)
70  +                 binding.historyCard.update(data.history)
71  +                 binding.barCard.update(data.bar)
72  +
73  +                 // checkmarks for calories
74  +                 var newCalorieCheckmarksList = mutableListOf<Checkmark>()
75  +                 for (checkmark in data.calorieBar.checkmarks){
76  +                     if(newCalorieCheckmarksList.size < data.calorieBar.checkmarks.size) {
77  +                         var newCalorie = Checkmark(checkmark.timestamp, checkmark.value * habit.calorieBurned.toInt() );
78  +                         newCalorieCheckmarksList.add(newCalorie)
79  +                     }
80  +                 }
81  +                 var newCalorieBarData = CalorieBarCardViewModel(newCalorieCheckmarksList,data.calorieBar.bucketSize,data.calorieBar.color,data.calorieBar.isNum
82  +
83  +                 // checkmarks for hydration
84  +                 var newHydrationCheckmarksList = mutableListOf<Checkmark>()
85  +                 for (checkmark in data.hydrationBar.checkmarks){
86  +                     if(newHydrationCheckmarksList.size < data.hydrationBar.checkmarks.size) {
87  +                         var newHydration = Checkmark(checkmark.timestamp, checkmark.value * habit.hydration.toInt() );
88  +                         newHydrationCheckmarksList.add(newHydration)
89  +                     }
90  +                 }
91  +                 var newHydrationBarData = HydrationBarCardViewModel(newHydrationCheckmarksList,data.hydrationBar.bucketSize,data.hydrationBar.color,data.hydrationBar.isNum
92  +
93  +                 // checkmarks for activity duration
94  +                 var newActivitydurationCheckmarksList = mutableListOf<Checkmark>()
95  +                 for (checkmark in data.activitydurationBar.checkmarks){
96  +                     if(newActivitydurationCheckmarksList.size < data.activitydurationBar.checkmarks.size) {
97  +                         var newActivityduration = Checkmark(checkmark.timestamp, checkmark.value * (habit.activityDuration.toInt()/60) );
98  +                         newActivitydurationCheckmarksList.add(newActivityduration)
99  +                     }
100 +                 }
101 +                 var newActivitydurationBarData = ActivitydurationBarCardViewModel(newActivitydurationCheckmarksList,data.activitydurationBar.bucketSize,data.activitydurationBar.color,data.activitydurationBar.isNum
102 +
103 +                 binding.calorieBarCard.update(newCalorieBarData)
104 +
105 +                 binding.hydrationBarCard.update(newHydrationBarData)
106 +
107 +                 binding.activitydurationBarCard.update(newActivitydurationBarData)
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124

```

```
 30 android/uhabits-android/src/main/res/layout/show_habit.xml
...
... ... @@ -1,5 +1,5 @@
1   1     <!--
2   -   ~ Copyright (C) 2016 Alinson Santos Xavier <isoron@gmail.com>
2   +   ~ Copyright (C) 2016 linson Santos Xavier <isoron@gmail.com>
3   3     ~
4   4     ~ This file is part of Loop Habit Tracker.
5   5     ~
+ @@ -65,13 +65,15 @@
65  65         <org.isoron.uhabits.activities.habits.show.views.ScoreCard
66  66             android:id="@+id/scoreCard"
67  67             style="@style/Card"
68  -           android:gravity="center"/>
68  +           android:gravity="center"/>
69  69
70  70         <org.isoron.uhabits.activities.habits.show.views.BarCard
71  71             android:id="@+id/barCard"
72  72             style="@style/Card"
73  73             android:gravity="center"/>
74  74
75  + +
76  +
75  77         <org.isoron.uhabits.activities.habits.show.views.HistoryCard
76  78             android:id="@+id/historyCard"
77  79             style="@style/Card"
+ @@ -86,6 +88,30 @@
86  88             android:id="@+id/frequencyCard"
87  89             style="@style/Card"/>
88  90
91  +         <org.isoron.uhabits.activities.habits.show.views.CalorieBarCard
92  +             android:id="@+id/calorieBarCard"
93  +             style="@style/Card"
94  +             android:gravity="center"/>
95  +
96  +         <org.isoron.uhabits.activities.habits.show.views.HydrationBarCard
97  +             android:id="@+id/hydrationBarCard"
98  +             style="@style/Card"
99  +             android:gravity="center"/>
100 +
101 +         <org.isoron.uhabits.activities.habits.show.views.ActivitydurationBarCard
102 +             android:id="@+id/activitydurationBarCard"
103 +             style="@style/Card"
104 +             android:gravity="center"/>
105 +
106 +         <Button
107 +             android:id="@+id/googleFitBtn"
108 +             android:layout_width="match_parent"
109 +             android:layout_height="55dp"
110 +             android:backgroundTint="#0075D9"
111 +             android:layout_marginTop="16dp"
112 +             android:layout_marginBottom="20dp"
113 +             android:text="Open Google Fit" />
114 +
89  115     </LinearLayout>
90  116   </ScrollView>
91  117
```

5.0 Reference List

6.0 Appendices