Habit Cookbook

1. Build Instructions

To build this application you are required to install the latest version of Android Studio following the instructions here.

https://developer.android.com/sdk/index.html

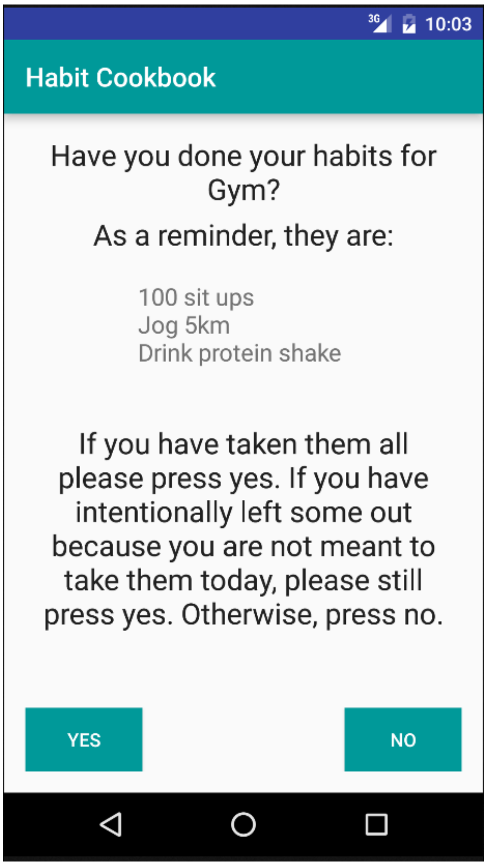
For installation on an emulator you will also need to download the appropriate SDK via android studio for the emulator you wish to test on (by default one is provided for the latest android version). The application requires version 4.4 or higher.

For installation on physical devices you must have developer mode on your android device enabled by following the instructions here.

https://developer.android.com/tools/device.html

2. Application Usage and Editing Notification Phase-out.

By default the alarms are set to repeat every 24 hours multiplied by a rate number. This rate number is determined by a 'count' associated with the routine. Initially the count is 1 but goes up each time the user responds positively to an 'End of Routine' post-notification where they are required to assess whether they have carried out their intended habits as shown below. When they respond negatively the count goes down by 1 (but never below 1 itself).



Have you done your habits for Gym?

As a reminder, they are:

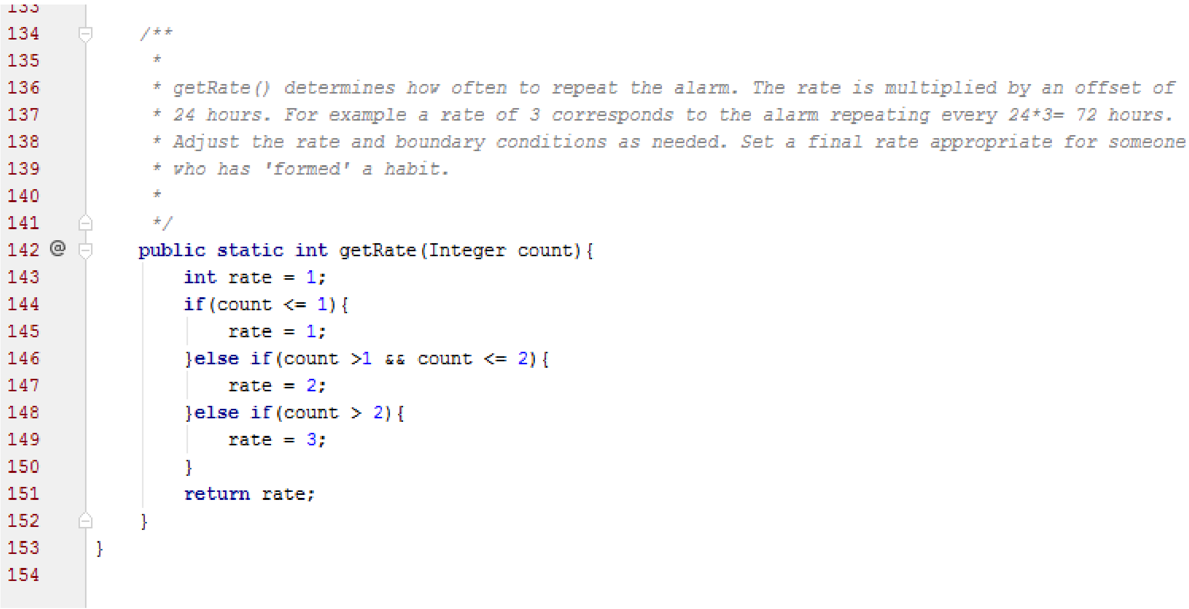
100 sit ups
Jog 5km
Drink protein shake

If you have taken them all please press yes. If you have intentionally left some out because you are not meant to take them today, please still press yes. Otherwise, press no.

To edit how the rate is determined from the count, you can amend the snippet of code below in the AlarmSetUp.java class in the following file directory.

Habit-Cookbook\app\src\main\java\com\example\wahidur\fragments\AlarmSetup.java

The relevant code is on line numbers 142-152 in the method labelled ***getRate()***.



```
133
134     /**
135      *
136      * getRate() determines how often to repeat the alarm. The rate is multiplied by an offset of
137      * 24 hours. For example a rate of 3 corresponds to the alarm repeating every 24*3= 72 hours.
138      * Adjust the rate and boundary conditions as needed. Set a final rate appropriate for someone
139      * who has 'formed' a habit.
140      *
141      */
142 @   public static int getRate(Integer count){
143         int rate = 1;
144         if(count <= 1){
145             rate = 1;
146         }else if(count >1 && count <= 2){
147             rate = 2;
148         }else if(count > 2){
149             rate = 3;
150         }
151         return rate;
152     }
153 }
154
```

This snipped of code can easily be edited so that the notifications phase out at your preferred rate (if you are not familiar with java please feel free to ask us to edit it ourselves!). In the current form a given alarm set will occur once within 24 hours, and then at 48 hours and then finally at 72 hours where it will continue repeating every 72 hours. It is up to you to decide what level of repetition is appropriate for someone who has formed a habit i.e. once a week (rate = 7).

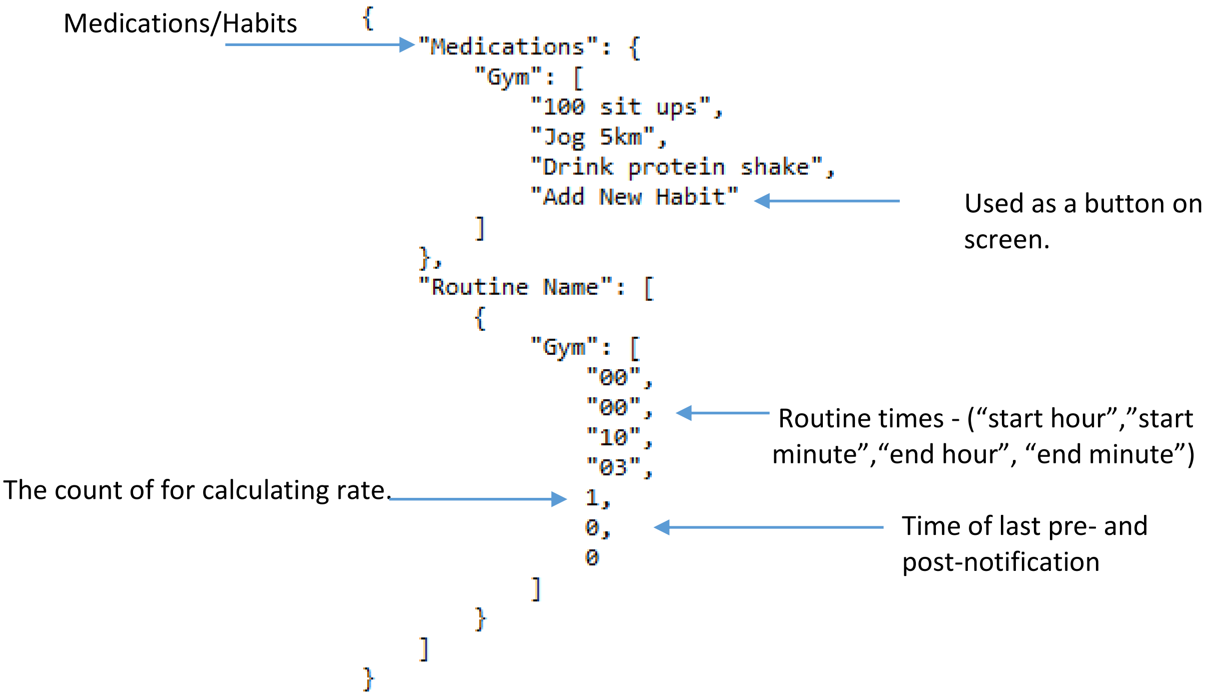
3. Analytics Set-up

The application has tracking ID UA-72545234-1 and can be viewed once you have set up an account here.

https://www.google.co.uk/analytics/

4. JSON File Structure

The JSON file structure of each users routines and habits is given below and described. This structure is sent off to Google Analytics using a custom HitBuilder. It is set up to be sent whenever the user enters the Home Screen as all interactions that lead you to the Home Screen occur only when a change has occurred in the routine structure which immediately updates the JSON file and then sends the JSON String off.



```
{
    "Medications": {
        "Gym": [
            "100 sit ups",
            "Jog 5km",
            "Drink protein shake",
            "Add New Habit"
        ]
    },
    "Routine Name": [
        {
            "Gym": [
                "00",
                "00",
                "10",
                "03",
                1,
                0,
                0
            ]
        }
    ]
}
```

Medications/Habits

Used as a button on screen.

Routine times - (“start hour”,”start minute”,“end hour”, “end minute”)

The count of for calculating rate.

Time of last pre- and post-notification

It should be noted that the time of last pre- and post-notification is measured in milliseconds from the 1st of January 1970 - an odd quirk of Android Studio/Java and it is how it decides when the alarms are set to fire.

Instances of this structure in the analytics for each user should be different from each iteration and therefore allow you to deduce which changes have occurred. If it is not, one possibility is the application crashed on the Home Page (hopefully won’t happen!) and then reloaded the Home Page so the JSON string at the instance was sent twice with no changes in between. The reason could be that they manually close the app in the task stack and then reopen it. This reloads the home screen and sends another JSON string off.

The count in the JSON structure is also reset every time the user edits a routine. This is to prevent ‘cheating’ whereby a user could appear to have formed a habit at night by reaching the final rate number but in reality has done the habits during the day before changing the time to the night.

5. User guidelines

Ideally the user should not set Routines that are too close (say 5 minutes) to the start and end times of another routine as the new notification overwrites the previous one if it has not been acknowledged yet. Each notification represents a unique action which populates the notification screen differently therefore it is not as easy to stack notifications the usual way when the notifications are generic.