GitHub Username: neev

Easyfit

Description

Easyfit is a simple fitness app to encourage women to be active and fit with ease and fun. The app enables the users to log physical activities and workouts in a digital journal.

To make it fun, the app starts with a clear rainbow image in the home page whose bows are initially white. The bows get colored automatically one by one when a workout is logged for the current week starting from Sunday. The fun is to complete at least seven workouts within the week to color the rainbow completely. Every Sunday the rainbow clears for the upcoming week workouts.

The user can also see the history of workouts completed for the current month or week.

The calendar view in the app highlights the days a workout was logged in the digital journal.

Weekly workouts can be shared with other users through Whatsapp messaging app.

Daily notifications are sent to users on the weekly progress.

An associated widget graphically displays the weekly progress through the rainbow image.

Intended User

This is a free app available to users in all the countries. The intended target users are those who want a fun way to log and monitor the progress of their weekly workouts to keep themselves fit.

Features

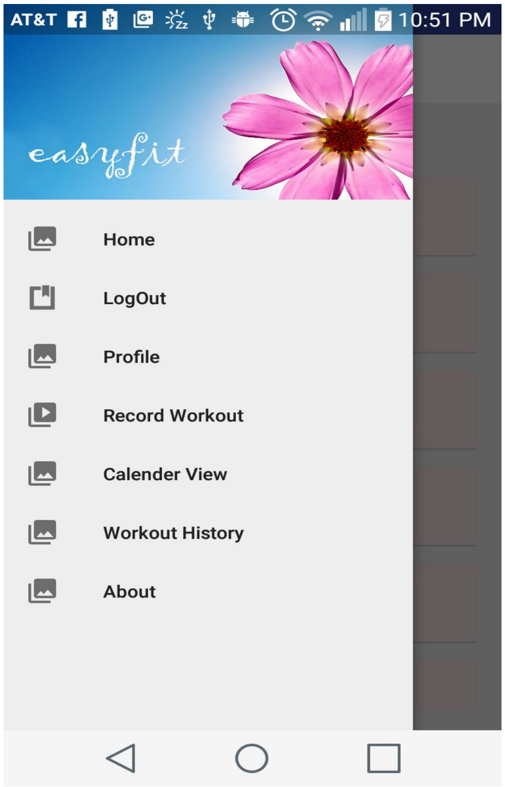
Key features of the app:

- Create a user profile.
- Pick a predefined workout and log the workout duration.
- Calendar view of the workout journal.
- View workout history.
- Rainbow image in home-page to display weekly progress.
- Daily notification on weekly progress.

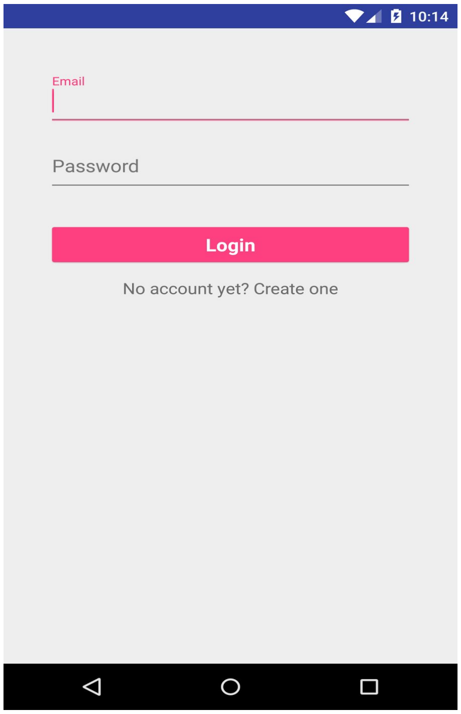
- Share workouts using messaging apps.
- Widget of rainbow image to display weekly progress.
- Localization of the app

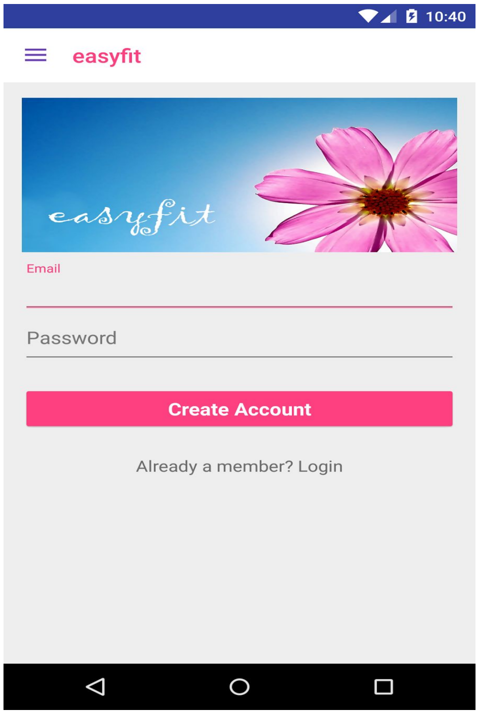
User Interface Mocks

Screen 1: Navigation menu

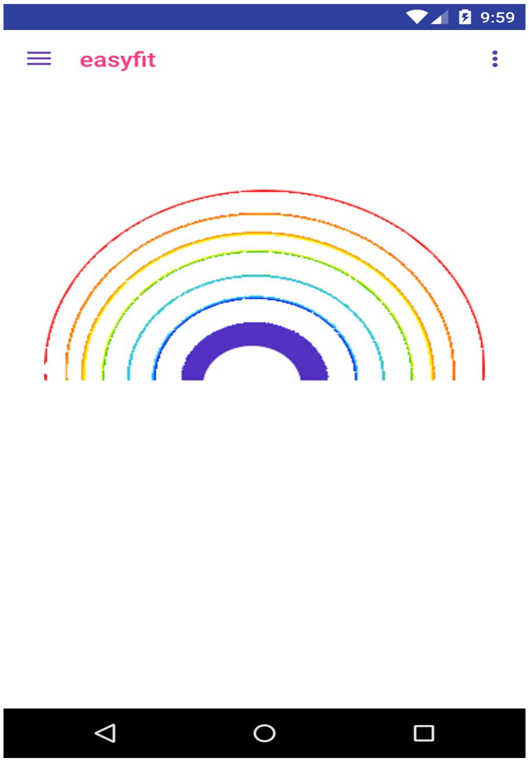


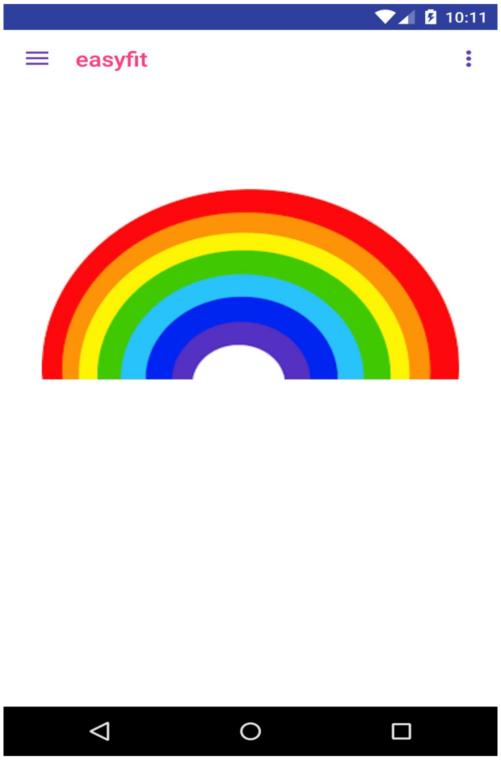
Screen 2: Login Page - Sign Up Page



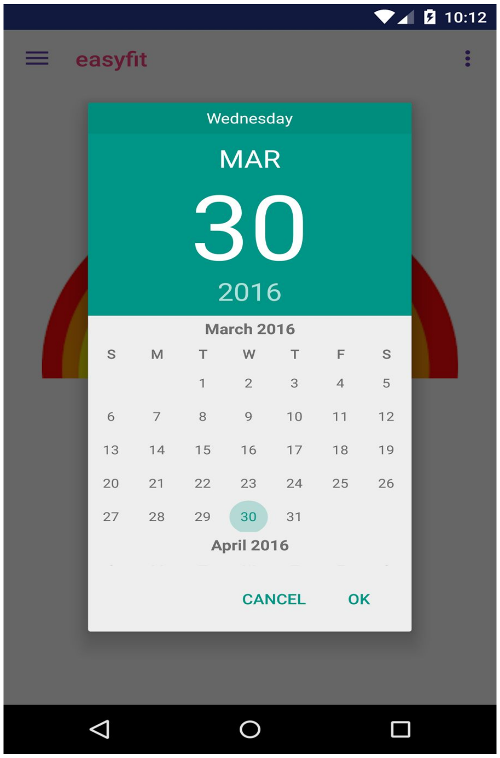


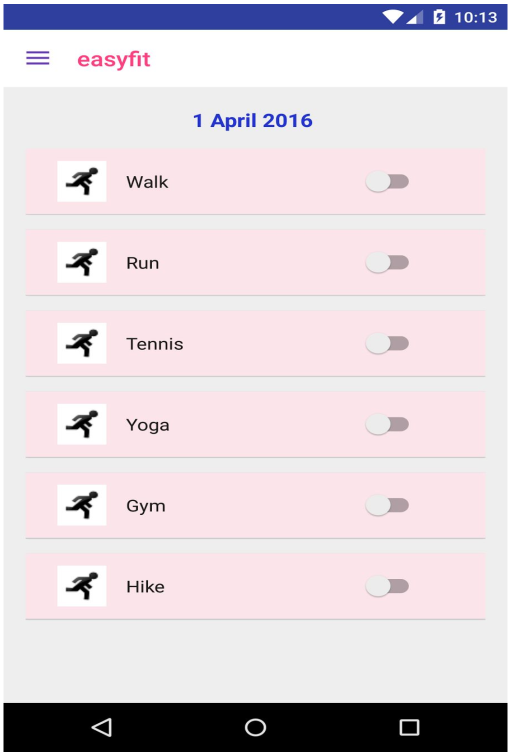
Screen 3: Home Page - transition from clear to colored rainbow



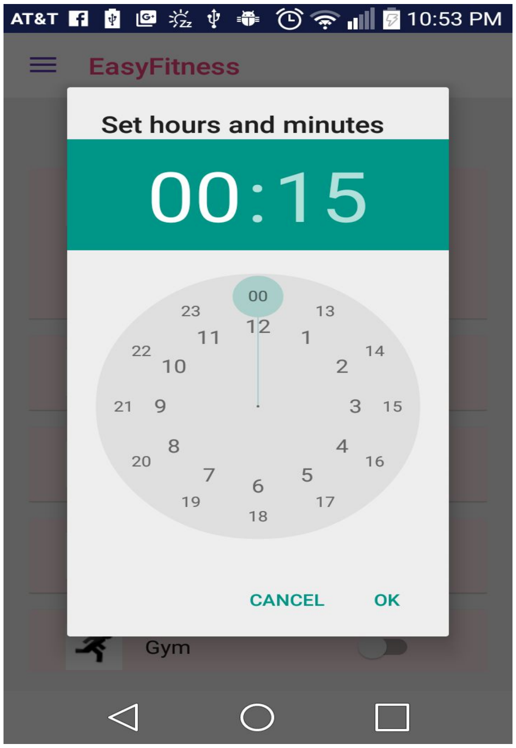


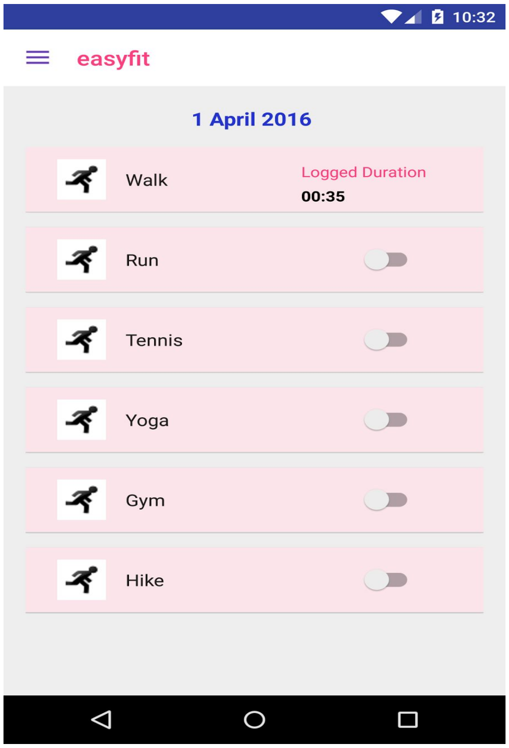
Screen 4 : Record Workout Page - Datepicker - Listview with switch button



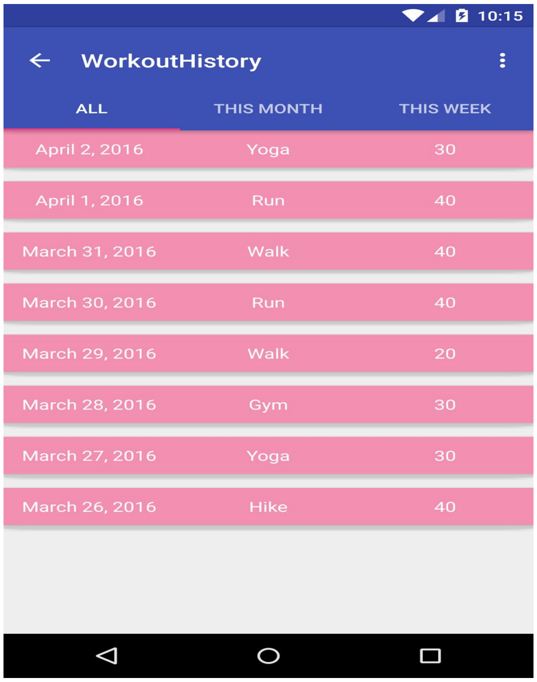


Screen 5 : Record Workout Page - Timepicker



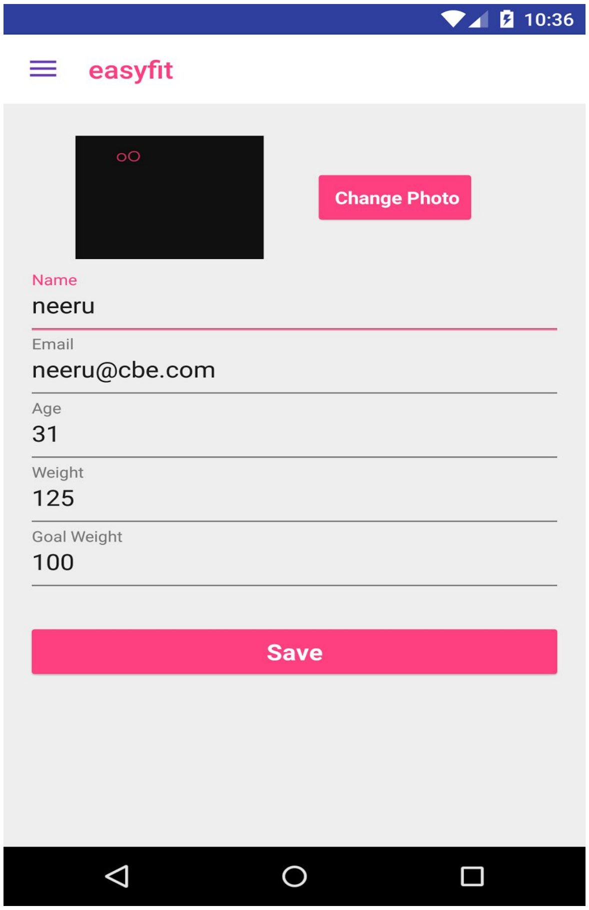


Screen 6 : Workout History - three tabs



April 2, 2016	Yoga	30
April 1, 2016	Run	40
March 31, 2016	Walk	40
March 30, 2016	Run	40
March 29, 2016	Walk	20
March 28, 2016	Gym	30
March 27, 2016	Yoga	30
March 26, 2016	Hike	40

Screen 7 : Profile entry page - Profile display page



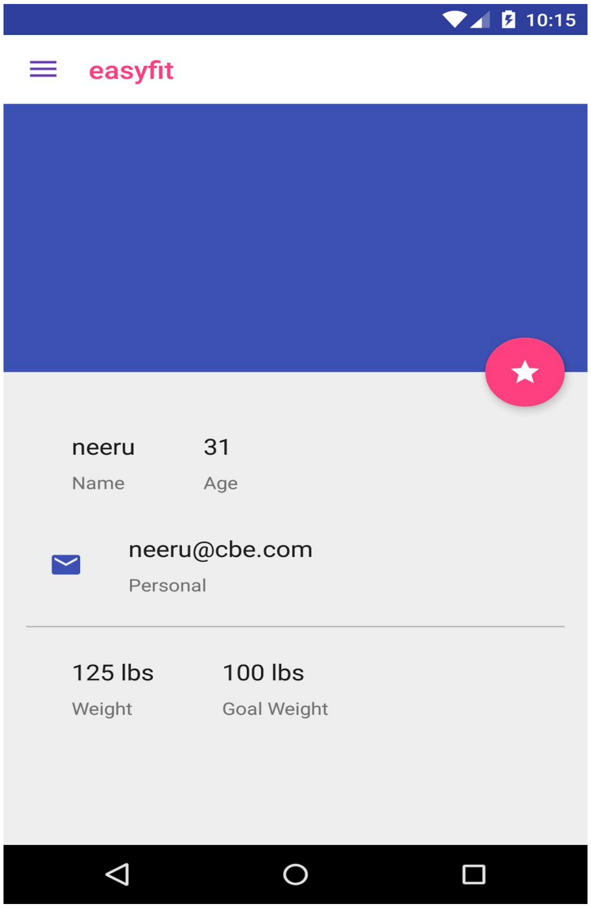
Name
neeru

Email
neeru@cbe.com

Age
31

Weight
125

Goal Weight
100



neeru — Name

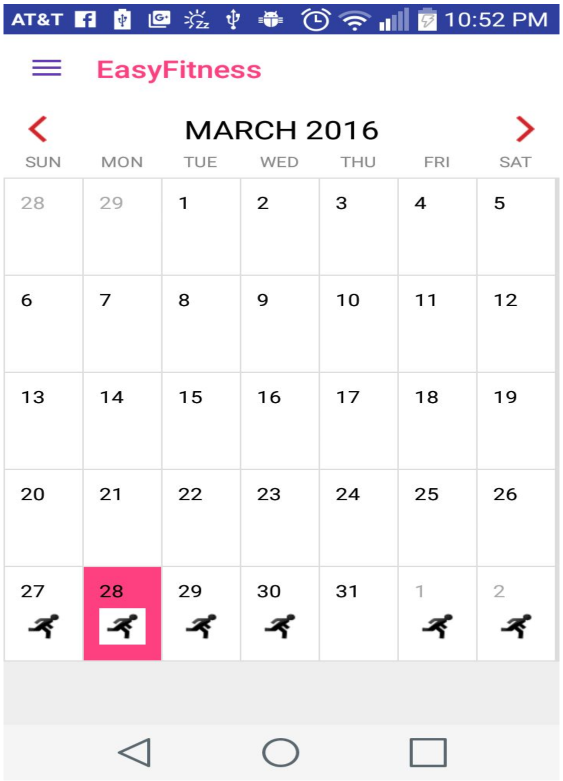
31 — Age

neeru@cbe.com — Personal

125 lbs — Weight

100 lbs — Goal Weight

Screen 8 : Custom Calendar View page

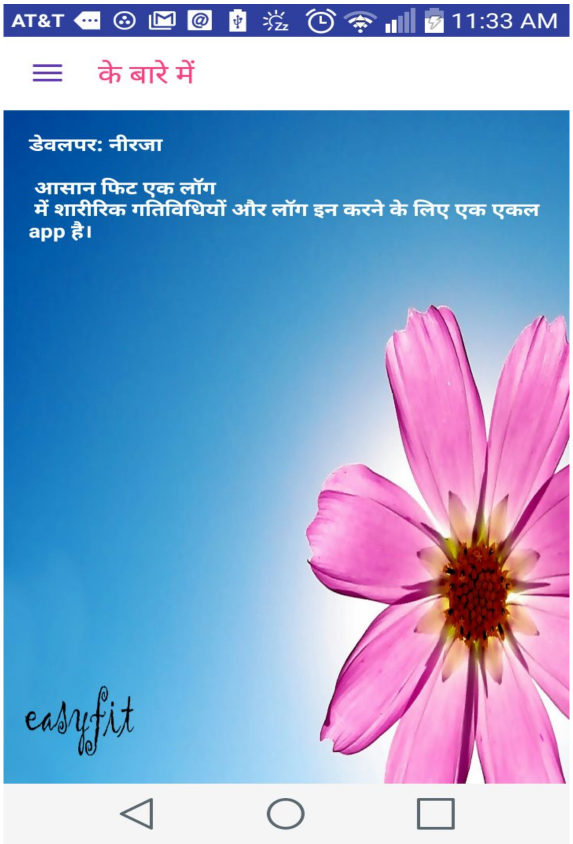


EasyFitness

MARCH 2016

SUN	MON	TUE	WED	THU	FRI	SAT
28	29	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27 🏃	28 🏃	29 🏃	30 🏃	31	1 🏃	2 🏃

Screen 9 : About page



के बारे में

डेवलपर: नीरजा

आसान फिट एक लॉग
में शारीरिक गतिविधियों और लॉग इन करने के लिए एक एकल app है।

Key Considerations

How will your app handle data persistence?

Content providers shall be used to handle data persistence.
They shall be used to store predefined workout options and user entered workout records.
Create custom cursor queries to query and modify the data managed by Content provider.

- Create query to get all the user workout records.
- Create query to get this month user workout record
- Create query to get this week user workout record
- Bulk insert for workout options
- Bulk insert for user user workout records

Describe any corner cases in the UX.

- Display the app bar in all the activities with the navigation drawer menu.
- When the user decides not to log a workout after clicking/sliding the switch button in the workout option page, the switch button is reset to off/false.

Describe any libraries you'll be using and share your reasoning for including them.

- Glide to handle the loading and caching of images.
- Caldroid for custom calendar view.
- Firebase to handle the user authentication in login page and to store user workout records.

Required Tasks

Task 1: Project Setup

- Modify the build.gradle file for the app by adding

```
compile 'com.github.bumptech.glide:glide:3.5.2'
compile 'com.jakewharton:butterknife:7.0.1'
compile 'com.firebase:firebase-client-android:2.5.2+'
compile 'com.roomorama:caldroid:3.0.1'
compile 'com.google.android.gms:play-services-location:7.5.0'
```

- Include the following within the android{} in app - build.gradle file

```
packagingOptions {
   exclude 'META-INF/DEPENDENCIES'
   exclude 'META-INF/NOTICE'
   exclude 'META-INF/LICENSE'
   exclude 'META-INF/LICENSE.txt'
   exclude 'META-INF/NOTICE.txt'
   exclude 'META-INF/LICENSE-FIREBASE.txt'
}
```

Enter all the string values of the app in the string.xml file
Enter all the dimension values of the app in the dimens.xml

Task 2: Implement UI for Each Activity and Fragment

- Build UI for Navigation menu
- Build UI for MainActivity (Home)
- Build UI for Create new account and LogIn
- Build UI for User Profile entry page and Profile display page
- Build UI for Record Workout
- Build UI for Calendar View
- Build UI for Workout History
- Build UI for About
- Build UI for the Navigation bar

Task 2(i):MainActivity (Home Page)

This activity is the home page displaying the rainbow image (please refer : screen 3)

- every sunday the app starts with a clear rainbow image in the home page whose bows leaves are initially white.
- the bows get colored automatically one by one when a workout is logged for the current week.
- every Sunday the rainbow clears for the upcoming week workouts.

Task 2(ii):Log In

This activity is the login page (please refer : screen 2)

- User enters the registered email and password to login to the app and access the logged workouts.
- Validate the user entered values
- Store the authenticated authId in the session using shared preference, to check the user is logged in always.
- Check the user is logged in before navigating to any other activities in the app.
- Toggle LogIn and LogOut menu options in the navigation bar.

Task 2(iii): Create new account

This activity allows the user to enter a valid email address and password to register into the firebase authentication and login account, thus providing the user to create his profile.
(please refer : screen 2)

- User enters the email address in the edit textbox
- User enters the password in the edit textbox
- validate() the entered values
- Create the new account when the “create new account “ button is clicked
- Display a user friendly toast message when the new account is created and intent to the login page for the user to login with new account

Task 2(iii): Profile edit activity and Profile display activity

Using the material design layout with collapsing app bar create the profile page
(please refer : screen 7)

- Allow the user to choose an image from the device gallery when the add photo button is clicked. Use ACTION_PICK_INTENT.
- User enters the name, email address, age, weight and goal weight in the edit textbox.
- Allow the user to enter his current location using the place picker widget.

Task 2(iii): Record Workout activity

This activity allows the user to log the workout by choosing a predefined workout options and picking the duration of the respective workout from the time picker dialog.
(please refer : screen 4 & 5)

- Display the calendar picker dialog when the “Record Workout” menu is clicked in the navigation bar, where the user can choose the date to log in the workout for the chosen date.
- On top below the app bar display the picked date from the calendar picker dialog with predefined list of workout options with switch button.
- When the switch button is clicked /swiped on show a time picker dialog for the user to choose the hours:minutes.
- When clicked ok in the time picker dialog, hide the switch button of the selected workout option and display the message
  “Logged Duration”
  “ 00:30 minutes”

Task 2(iv): Calendar View

Show a calendar view with current month displaying the “run” image on the dates the user logged in the workout duration.
When the user swipes back and forth to the previous and next month show the same as above respect to the that month. (please refer : screen 8)

Task 2(v): Workout History activity

This is a three tabbed activity with the tabs - “All” , “This Month” and “This Week”

The tab “All”

- Displays all the logged workouts

The tab “This Month”

- Displays all the logged workouts for the current month

The tab “This week”

- Displays all the logged workouts for the current week

The workouts are displayed neatly in a listview, showing the date, workout description and duration.

March 31,2016 Walk 15 min
(please refer : screen 6)

Task 2(vi): About activity

This activity shows the simple description of the app with logo image as the background.
(please refer : screen 9)

Task 3: Persisting Data

Implement Firebase server side method to create new user login
Implement authentication Firebase server side methods in the login page
Implement Firebase server side methods to store user entered profile information
Implement Firebase server side methods to retrieve workout options
Implement Firebase server side methods to store entered workout records
Implement Google Play Services for maps and Admob

Task 4: Integration with Google Services, Firebase, Messaging

Integration with the Firebase:

- Inside the key.xml file declare the url string to fetch the data or enter the data into firebase
- Initialize Firebase with the application context

  Firebase.set Android Content(context);
- With reference object

```
Firebase ref = new Firebase(getResources().getString(R.string.firebase_url));
Call createUser method to create new user
Call authenticate method to authenticate the user in login page
Call setValue() with push() to store values in the Firebase server
Call addListenerForSingleValueEvent to retrieve values from the server.
```

Messaging

- Use ShareActionProvider to share content with messaging / whatsapp

Google Services

- Integrate the google maps to store the user current location.

Task 5: Notifications

Notify the user when the rainbow is completely colored.
Send notification three times a day

- At 8 am (locale) “Please log in todays workout”
- At 1 pm (locale) “Please log in today’s workout” - only if today’s workout is not logged in.
- At 8 pm (locale) “Please log in today’s workout” - only if today’s workout is not logged in.

Task 6: Screen orientation

- On screen orientation handle the error cases with proper user friendly dialog messages if the app crashes.
- Restore the edit text values on screen orientation.

Task 7: Widgetization

Display a widget with the colored rainbow image with today and yesterday logged in workouts.

- Create layout for widget with 3x2 dimension
- Display yesterday and today’s logged workouts in a listview
- When clicked on the widget list item or on the widget rainbow image intent to the home page.
- When no workouts have been logged in display “No workouts logged in”

Task 8: Validation, Error and Exception handling

Validation:

- Create a method called validate() to validate
    - user entered email is in email format
    - user entered password is between 4 to 8 alphanumeric characters
    - user entered age is 2 digit numbers only
    - user entered weight is 3 digits numbers only
- Exception handling during orientation are handled in try catch block and the error messages are displayed in user friendly messages.
- Exception handling for place picker to enter/pick the current location,also handle the exception with no internet connection
- Error and exception handling for no internet connection