

SEMESTER PROJECT

(FASTnFITNESS)

LINKS:

https://drive.google.com/file/d/1sHFhbORLBwrUxdfCgDmFJGt8sxZK-KryR/view?usp=drive_link

GROUP MEMBERS:

- ALI HAMZA
- HUSSNAIN ZIA
- FAIZAN NABI

PROJECT DESCRIPTION: FITNESS APP

Project Goal:

To develop a comprehensive fitness app that empowers users to achieve their health and wellness objectives through personalized workout plans, nutrition tracking, progress monitoring, and a supportive community.

Target Audience:

The app will cater to a diverse range of users, including individuals seeking weight loss, muscle gain, improved fitness levels, and overall well-being.

Key Features:

- **Personalized Workout Plans:** Generate tailored workout routines based on user fitness goals, experience level, and equipment availability.
- **Exercise Library:** Offer a vast database of exercises with detailed instructions, video demonstrations, and difficulty levels.
- **Nutrition Tracking:** Provide tools for calorie counting, macronutrient tracking, and meal planning to support dietary goals.
- **Progress Tracking:** Monitor user progress through metrics like weight, body measurements, and exercise performance.

Technology Stack:

- SQLite
- Java

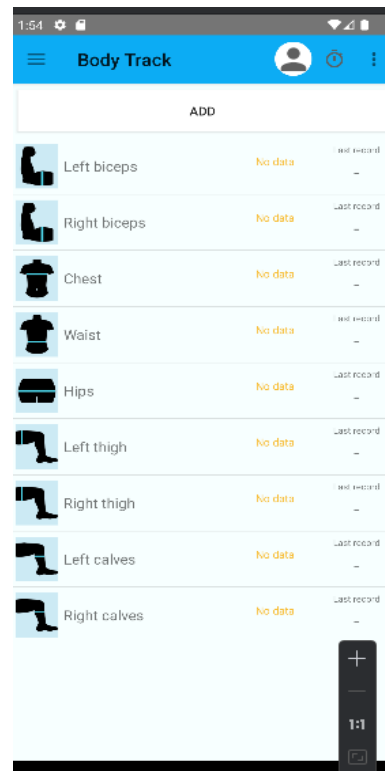
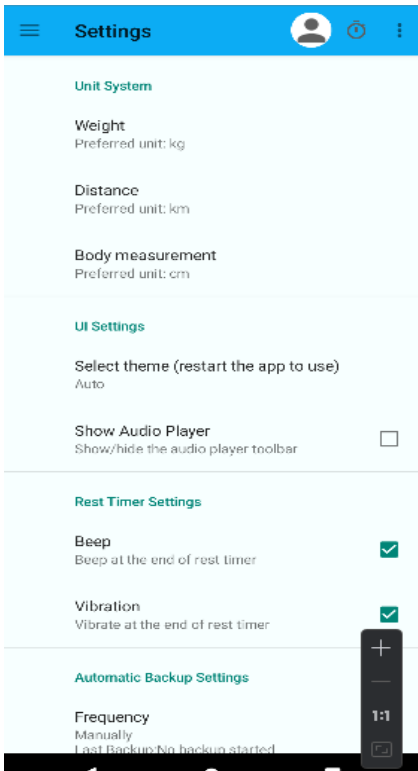
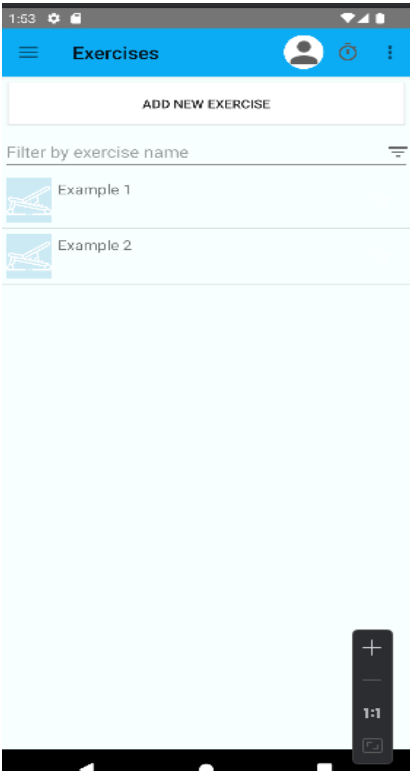
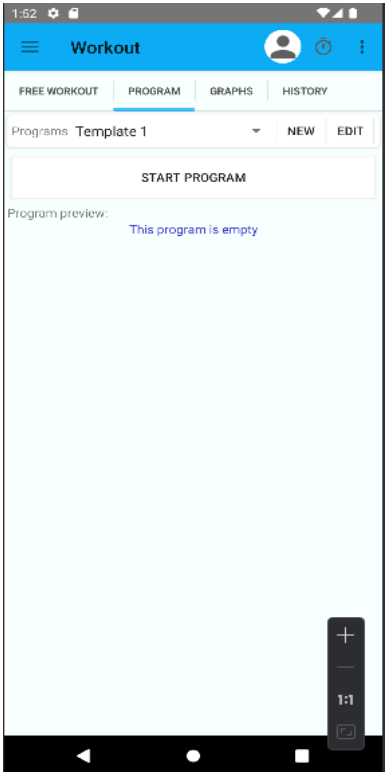
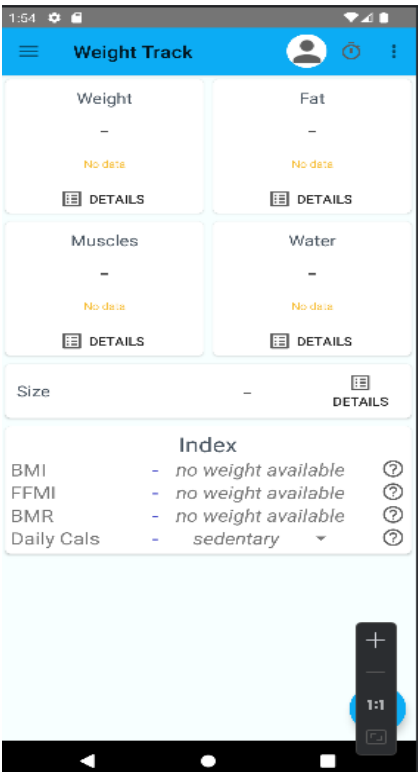
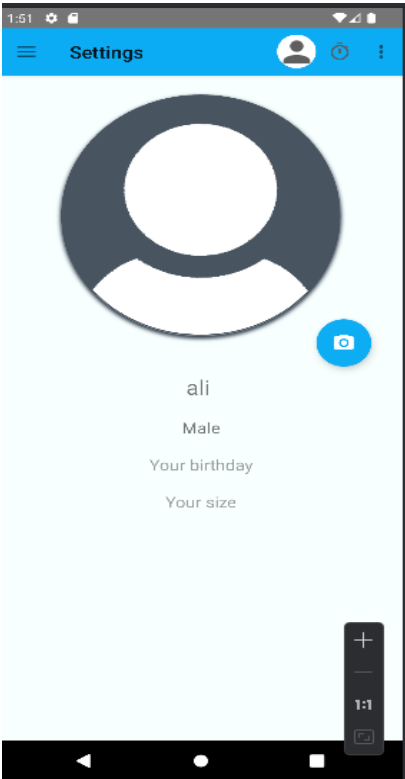
Key features/Functionalities/Screens:

1. **Profile Management:**
 - Creating, selecting, deleting, and renaming profiles.
 - Setting and displaying profile pictures.
2. **Workout Tracking:**
 - Logging and viewing strength training exercises (FontesPagerFragment).
 - Recording and tracking body weight (WeightFragment).
 - Logging cardio exercises and viewing workout history.
3. **Exercise Data:**
 - Viewing and managing exercise machines (MachineFragment).
 - Accessing and following workout programs (ProgramListFragment).
4. **Progress Tracking:**
 - Tracking body measurements (BodyPartListFragment).
 - Viewing progress pictures (ProgressImagesFragment).
5. **Settings and Preferences:**
 - Customizing app settings (SettingsFragment).
 - Day/night mode selection.
6. **Data Management:**
 - Exporting workout data to CSV format.
 - Importing workout data from CSV or ZIP files.
 - Database management (deleting and migrating).
7. **Additional Features:**
 - Music control integration.
 - Chronometer for tracking workout duration.
 - Introduction/tutorial screens for new users.

Dependencies/Libraries Used:

1. **Android Support Libraries:** For compatibility and UI elements.
2. **CircularImageView:** For displaying profile pictures in a circular format.
3. **KToast:** For displaying customized toast messages.
4. **SweetAlert:** For displaying customizable alert dialogs.
5. **SQLite:** For database management.
6. **MusicController:** A custom class for music control.
7. **ImageUtil:** A custom class for image handling.
8. **DateConverter, FileNameUtil, UnitConverter:** Custom utility class

SCREEN SHOT (UI):



```
gradle.properties  </> activity_main.xml  MainActivity.java  </> bodymeasure_row.xml
1132 public class MainActivity extends AppCompatActivity {
1133     private class DrawerItemClickListener implements ListView.OnItemClickListener {
1134         public void onItemClick(AdapterView parent, View view, int position, long id) {
1135             // Insert the fragment by replacing any existing fragment
1136             switch (position) {
1137                 case 0:
1138                     showFragment(PROFILE);
1139                     setTitle(getString(R.string.ProfileLabel));
1140                     break;
1141                 case 1:
1142                     showFragment(FONTESPAGER);
1143                     setTitle(getResources().getText(R.string.menu_Workout));
1144                     break;
1145                 case 2:
1146                     showFragment(MACHINES);
1147                     setTitle(getResources().getText(R.string.MachinesLabel));
1148                     break;
1149                 case 3:
1150                     showFragment(WORKOUTS);
1151                     setTitle(getString(R.string.workout_list_menu_item));
1152                     break;
1153                 case 4:
1154                     showFragment(WEIGHT);
1155                     setTitle(getResources().getText(R.string.weightMenuLabel));
1156                     break;
1157                 case 5:
1158                     showFragment(BODYTRACKING);
1159                     setTitle(getResources().getText(R.string.bodytracking));
1160                     break;
1161                 case 6:
1162                     // ...
1163             }
1164         }
1165     }
1166 }
```

```
Android  gradle.properties  </> activity_main.xml  MainActivity.java  ProfileFragment.java  </> bodymeasure_row.xml
45 public class ProfileFragment extends Fragment {
102 }
103
104 @Override
105 public View onCreateView(LayoutInflater inflater, ViewGroup container,
106     Bundle savedInstanceState) {
107
108     // Inflate the layout for this fragment
109     View view = inflater.inflate(R.layout.tab_profile, container, attachToRoot: false);
110
111     sizeEdit = view.findViewById(R.id.size);
112     birthdayEdit = view.findViewById(R.id.birthday);
113     nameEdit = view.findViewById(R.id.name);
114     genderEdit = view.findViewById(R.id.gender);
115     roundProfile = view.findViewById(R.id.photo);
116     photoButton = view.findViewById(R.id.actionCamera);
117
118     daoProfile = new DAOProfile(view.getContext());
119     daoBodyMeasure = new DAOBodyMeasure(view.getContext());
120     daoBodyPart = new DAOBodyPart(view.getContext());
121
122     appVimMo = new ViewModelProvider(requireActivity()).get(AppVimMo.class);
123
124     /* Initialisation des valeurs */
125     imgUtil = new ImageUtil(this, roundProfile);
126     // ImageView must be set in onStart. Not in onCreateView
127
128     genderEdit.setCustomDialogBuilder(view1 -> {
129         SweetAlertDialog dlo = new SweetAlertDialog(view1.getContext(), SweetAlertDialog.NORMAL_TYPE)
130     });
131 }
```

```
gradle.properties x </> activity_main.xml MainActivity.java FontesPagerFragment.java x ProfileFragment.java </> bodymeasure_row.xml
20 public class FontesPagerFragment extends Fragment {
21
22     @Override
23     public View onCreateView(LayoutInflater inflater, ViewGroup container,
24                             Bundle savedInstanceState) {
25
26         View view = inflater.inflate(R.layout.fontes_pager, container, attachToRoot: false);
27
28         // Locate the viewPager in activity_main.xml
29         mViewPager = view.findViewById(R.id.fontes_viewpager);
30
31         if (mViewPager.getAdapter() == null) {
32
33             // Supply index input as an argument.
34             Bundle freeWorkoutArgs = new Bundle();
35             freeWorkoutArgs.putInt("displayType", DisplayType.FREE_WORKOUT_DISPLAY.ordinal());
36             freeWorkoutArgs.putLong("templateId", -1);
37
38             Bundle guidedWorkoutArgs = new Bundle();
39             guidedWorkoutArgs.putInt("displayType", DisplayType.PROGRAM_RUNNING_DISPLAY.ordinal());
40             guidedWorkoutArgs.putLong("templateId", -1);
41
42             Bundle args = this.getArguments();
43             args.putLong("machineId", -1); // if -1, then display the graph and history fragment with machine selectors
44             args.putLong("machineProfile", -1);
45
46             pagerAdapter = new FragmentPagerAdapter(
47                 getChildFragmentManager(), FragmentPagerAdapterItems.with(this.getContext())
48                 .add("Free workout", FontesFragment.class, freeWorkoutArgs)
49                 .add("Program", ProgramRunnerFragment.class, guidedWorkoutArgs)
50                 .add("Graphs", FontesGraphFragment.class, args)
51             );
52         }
53     }
54 }
```

```
gradle.properties x </> activity_main.xml MainActivity.java MuscleTest.java WeightFragment.java x ProfileFragment.java </> bodymeasure_row.xml
62 public class WeightFragment extends Fragment {
63
64     private BodyPart waterBodyPart;
65     private BodyPart sizeBodyPart;
66     private AppVim appVim;
67     private final OnClickListeners mOnClickListeners = new OnClickListeners() {
68
69         @Override
70         public void onClick(View view) {
71             ValuesEditorDialog editorDialog;
72             switch (view.getId()) {
73                 case R.id.weightInput:
74                     BodyMeasure lastWeightMeasure = mDbBodyMeasure.getLastBodyMeasures(weightBodyPart.getId(), getProfile());
75                     Value lastWeighValue = getValueFromLastMeasure(lastWeightMeasure, SettingsFragment.getDefaultWeightUnit(getActivity()).toUnit(), id: null, "Weight");
76                     editorDialog = new ValuesEditorDialog(getActivity(), new Date(), time: "", new Value[] {lastWeighValue});
77                     editorDialog.setTitle("Add");
78                     editorDialog.setPositiveButton("Add");
79                     editorDialog.setOnDismissListener(dialog -> {
80                         if (!editorDialog.isCancelled()) {
81                             Date date = DateConverter.localDateStrToDate(editorDialog.getDate(), getContext());
82                             Value newValue = editorDialog.getValues()[0];
83                             mDbBodyMeasure.addBodyMeasure(date, weightBodyPart.getId(), newValue, getProfile().getId());
84                             refreshData();
85                         }
86                     });
87                     editorDialog.show();
88                     break;
89                 case R.id.fatInput:
90                     BodyMeasure lastFatMeasure = mDbBodyMeasure.getLastBodyMeasures(fatBodyPart.getId(), getProfile());
91                     final Value lastFatValue = getValueFromLastMeasure(lastFatMeasure, Unit.PERCENTAGE, id: null, "Fat");
92                     editorDialog = new ValuesEditorDialog(getActivity(), new Date(), time: "", new Value[] {lastFatValue});
93                     editorDialog.setTitle("Add");
94                 }
95             }
96     }
97 }
```

```
Dialogbox.java ExpandedListView.java ImageUtil.java MainIntroActivity.java AboutFragment.java AppVimo.java ChronoDialogbox.java CustomDrawerAdapter.java x
26 public class CustomDrawerAdapter extends ArrayAdapter<DrawerItem> { 2 usages Charles Combes <charles.combes> +4
39 }
40
41 @Override Charles Combes <charles.combes> +4
42 public View getView(int position, View convertView, ViewGroup parent) {
43
44     DrawerItemHolder drawerHolder;
45     View view = convertView;
46
47     if (view == null) {
48         LayoutInflater inflater = ((Activity) context).getLayoutInflater();
49         drawerHolder = new DrawerItemHolder();
50
51         view = inflater.inflate(layoutResID, parent, attachToRoot: false);
52         drawerHolder.itemName = view.findViewById(R.id.drawer_itemName);
53         drawerHolder.icon = view.findViewById(R.id.drawer_icon);
54
55         drawerHolder.spinner = view.findViewById(R.id.drawer_spinner);
56
57         drawerHolder.title = view.findViewById(R.id.drawer_title);
58
59         drawerHolder.headerLayout = view.findViewById(R.id.headerLayout);
60         drawerHolder.itemLayout = view.findViewById(R.id.itemLayout);
61         drawerHolder.spinnerLayout = view.findViewById(R.id.spinnerLayout);
62         drawerHolder.roundProfile = view.findViewById(R.id.header_icon);
63
64         view.setTag(drawerHolder);
65
66     } else {
67         drawerHolder = (DrawerItemHolder) view.getTag();
68     }
69 }
```

Run app x

```
© MainActivity.java  </> activity_main.xml  x
1  <androidx.drawerlayout.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"
2      xmlns:app="http://schemas.android.com/apk/res-auto"
3      xmlns:tools="http://schemas.android.com/tools"
4      android:id="@+id/drawer_layout"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent">
7
8      <!-- The main content view -->
9
10     <RelativeLayout
11         android:layout_width="match_parent"
12         android:layout_height="match_parent"
13         android:background="@color/background"
14         android:focusableInTouchMode="true"
15         android:orientation="vertical"
16         android:paddingLeft="0dp"
17         android:paddingTop="0dp"
18         android:paddingRight="0dp"
19         android:paddingBottom="0dp">
20
21         <androidx.appcompat.widget.Toolbar
22             android:id="@+id/action_toolbar"
23             android:layout_width="match_parent"
24             android:layout_height="wrap_content"
25             android:background="@color/toolbar_background"
26             android:elevation="4dp"
27             android:minHeight="?attr/actionBarSize"
28             android:visibility="visible">
29
30             <com.mikhaellopez.circularimageview.CircularImageView
31                 android:id="@+id/image_profile"
32                 android:layout_width="40dp"
33                 android:layout_height="40dp
```

```
© MainActivity.java  </> activity_main.xml  </> exercise_details.xml  x
1  <androidx.coordinatorlayout.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"
2      xmlns:app="http://schemas.android.com/apk/res-auto"
3      xmlns:tools="http://schemas.android.com/tools"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent">
6
7      <androidx.core.widget.NestedScrollView
8          android:layout_width="match_parent"
9          android:layout_height="match_parent"
10         android:paddingLeft="0dp"
11         android:paddingTop="0dp"
12         android:paddingRight="0dp"
13         android:paddingBottom="0dp"
14         app:layout_behavior="com.google.android.material.appbar.AppBarLayout$ScrollingViewBehavior">
15
16         <LinearLayout
17             android:id="@+id/machine_photo_layout"
18             android:layout_width="match_parent"
19             android:layout_height="wrap_content"
20             android:focusable="true"
21             android:focusableInTouchMode="true"
22             android:orientation="vertical"
23             android:paddingLeft="0dp"
24             android:paddingTop="0dp"
25             android:paddingRight="0dp"
26             android:paddingBottom="0dp">
27
28             <ImageView
29                 android:id="@+id/machine_photo"
30                 android:layout_width="match_parent"
31                 android:layout_height="wrap_content"
32                 android:adjustViewBounds="true"
33                 android:background="@color/record_background_even"
34             >
35         </LinearLayout>
36     </NestedScrollView>
37 </CoordinatorLayout>
```

