

# SELF TRACKER



## GROUP 3

Austin Johny

21

Fidha Febin M A

22

Devika K M

23

George Joseph

24

Jayakrishnan K

25

## Overview

Self Tracker is a user friendly android application useful in this COVID 19 pandemic situation. This app is having less memory. It supports Malayalam and English languages. The app is designed to reduce energy consumption.

Self Tracker can :

- Store the health information of the user
- Store the contact details of the user
- Store the location of the user with timestamp and date automatically
- Store the People the user meet semi with timestamp and date semi automatically
- Store the Vehicle the user uses semi with timestamp and date semi automatically
- Securely maintain the app using the password given by the user at the time of registration, No one other than registered user can open the app later.
- Enable user to see the track/path of the user in the map
- Share the details file using the app itself
- See the list of the people, vehicle and location as a list with timestamp and date
- The details file is stored in the storage as a text file.

Requirements in Smartphone

### Essential

- Location sensor
- Real time clock

### Non-Essential

- Speech recognizer
- Text to Speech
- Accelerometer sensor

## Details Collecting

The below mentioned details are collected by the app:

- Language preferred(Malayalam/English)
- Name of the user
- Age of the user
- Height of the user
- Weight of the user
- BMI of the user
- Gender of the user
- Blood group of the user
- Phone number of the user

- Email ID of the user
- Residence address of the user
- PIN code
- The password to secure the app
- The people met by the user and the timestamp and date
- The vehicle used by the user and the timestamp and date
- The location of the user and the timestamp and date

The details are very essential to the identification of the user. This data is also imported to the details text file.

### Technology and Components

The Self Tracker app uses many components like sensors and storage. It is essential for the smooth working and user reliability of the app. It contains:

- Temporary data storage
- Location sensor
- Speech recognizer
- Text to Speech
- Real time clock
- Accelerometer sensor

The temporary data storage module stores the details mentioned in the “Details Collecting” section entered by the user to the app. It also stores the user preferences using the app.

The location sensor uses GPS to find the location,speed,location accuracy.

The speech recognizer is to detect whether the user talks to a person. The user can also use that to write the data.

Text to Speech module is to remind the user to add the person if the user talks to a person.

Real time clock is to synchronize the app with the system time and to give the time stamp and date to the data storage and the file on request.

The accelerometer sensor is to sense the vibration of the phone and to hide the user details shown in the “Data Entry” screen.

### App Working

The first opening of the app shows the introductory screen as shown in figure below.

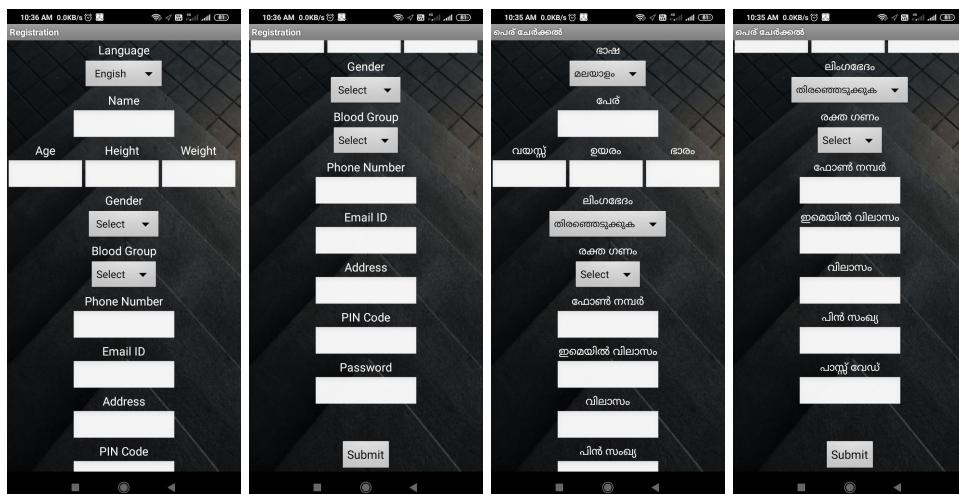
It shows the name of the app “SELF TRACKER” the symbol and the “Guide button”. The name and icon is for aesthetics.The guide button will show you the “Guide” page which shows how to use the app.



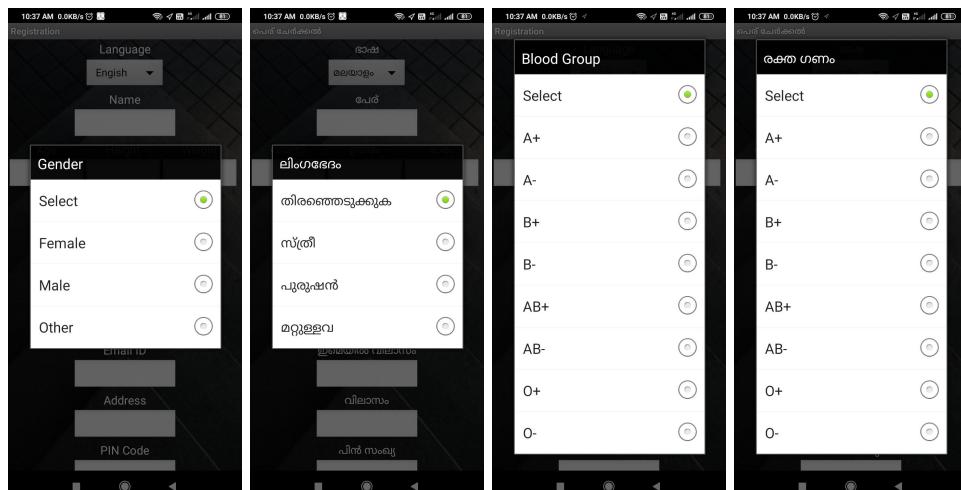
After that two seconds the “Registration” screen comes. Here the user needs to provide the details of the user. The details

- Language preferred(Malayalam/English)
- Name of the user
- Age of the user
- Height of the user
- Weight of the user
- Gender of the user
- Blood group of the user
- Phone number of the user
- Email ID of the user
- Residence address of the user
- PIN code
- The password to secure the app (8 characters)

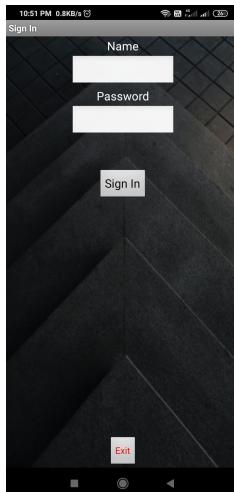
Without adding these details the user can not go to the next screen. Also the details must be correct, for example if the password is not 8 characters long or the email ID is not of the general pattern , the submission will not work. The user can choose Malayalam or English by the drop down spinner on the top of the screen. See the below figures



The user should also select the gender and blood group from the drop down spinner as shown below



After successfully submitting these details a notification of submission is shown and you will see the "Sign In" screen as shown below



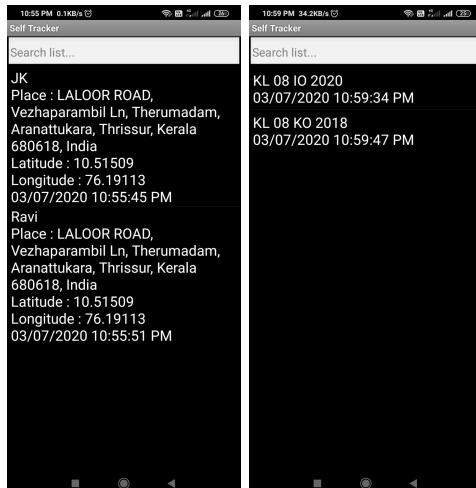
Enter the name and 8 character password that you created in the previous “Registration” screen and click the “Sign In” button. If your password matches with the password user given, the user will see the “Data Entry” screen. If it does not match you will see the alert of incorrect password. The algorithm checks the password and name is the same as registered before and returns what is needed. By default the user need to sign in using this screen always to go to the “Data Entry” Screen, But if the user have no such security issues, the user can check the check box “Don't ask me to sign in later” in the “Data Entry” screen to solve the problem of signing always.

After the successful sign in , the user can see the “Data Entry” screen. The details of the user can be seen at the top of the screen, the user can make it visible or invisible using the shake of the phone. By default the all automatic sensing of location, people, vehicle will be enabled. If the user wants, the user can turn off or on anything specific according to the situation. For example the user does not need the vehicle data entry, if he does not use the vehicle. The user can also enable or disable all three data entries with one click using the “Master” button. The user can see the map with his travel paths or locations by clicking the “Route Map” button and return to this screen by clicking the “Back” button on that screen.

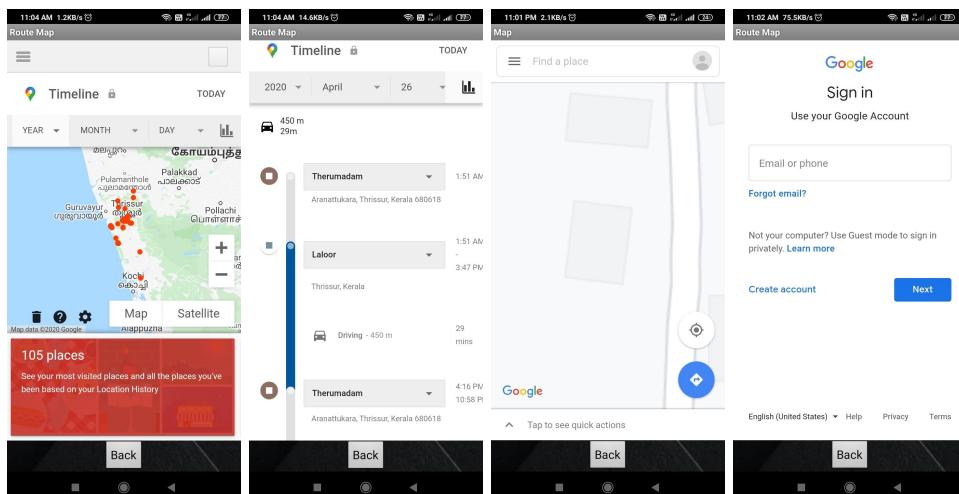


The app will detect a person if you talk with them , the app will sense that using Speech Recognizer and display an alert to add the name of person and tell the user using the Text to Speech module. Also it can detect if the person is travelling in a vehicle(Location and mobile data must be enabled). At that time also the app will sense that using Location Sensor and display an alert to add vehicle registration number details and tell the user using the Text to Speech module.

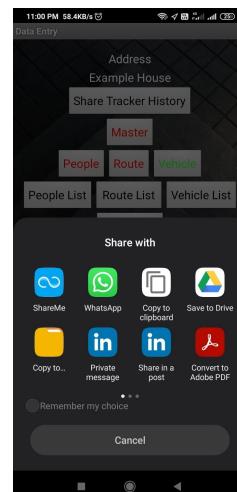
The second drop down menu below the label "Location Accuracy" is the selection to whether to add person only,add vehicle only or both. The user can select the appropriate selection to add the required. If it is a person, enter the name of the person and click the "Add Person" button. If it is a vehicle, enter the vehicle registration number and click the "Add Vehicle" button. If you want to add both, select the "Both" option in the drop down spinner and enter the name of person in and the vehicle registration number in the allotted text boxes and click the "Add" button. The user can only enter the name or registration number only if the location accuracy is less than 50 meters.The location accuracy is visible on the right of "Location Accuracy" label. The user can see the list of; people met, route travelled, vehicles used by clicking the button "People List", "Route List", "Vehicle List" respectively. By clicking there the user can see the list. If the user clicks the required element of that list, another screen with the location specified in that element is shown. Since it uses Google Map, you may need to sign in



By clicking the "Route Map" button and by clicking the any one element in the people list you can see like below respectively



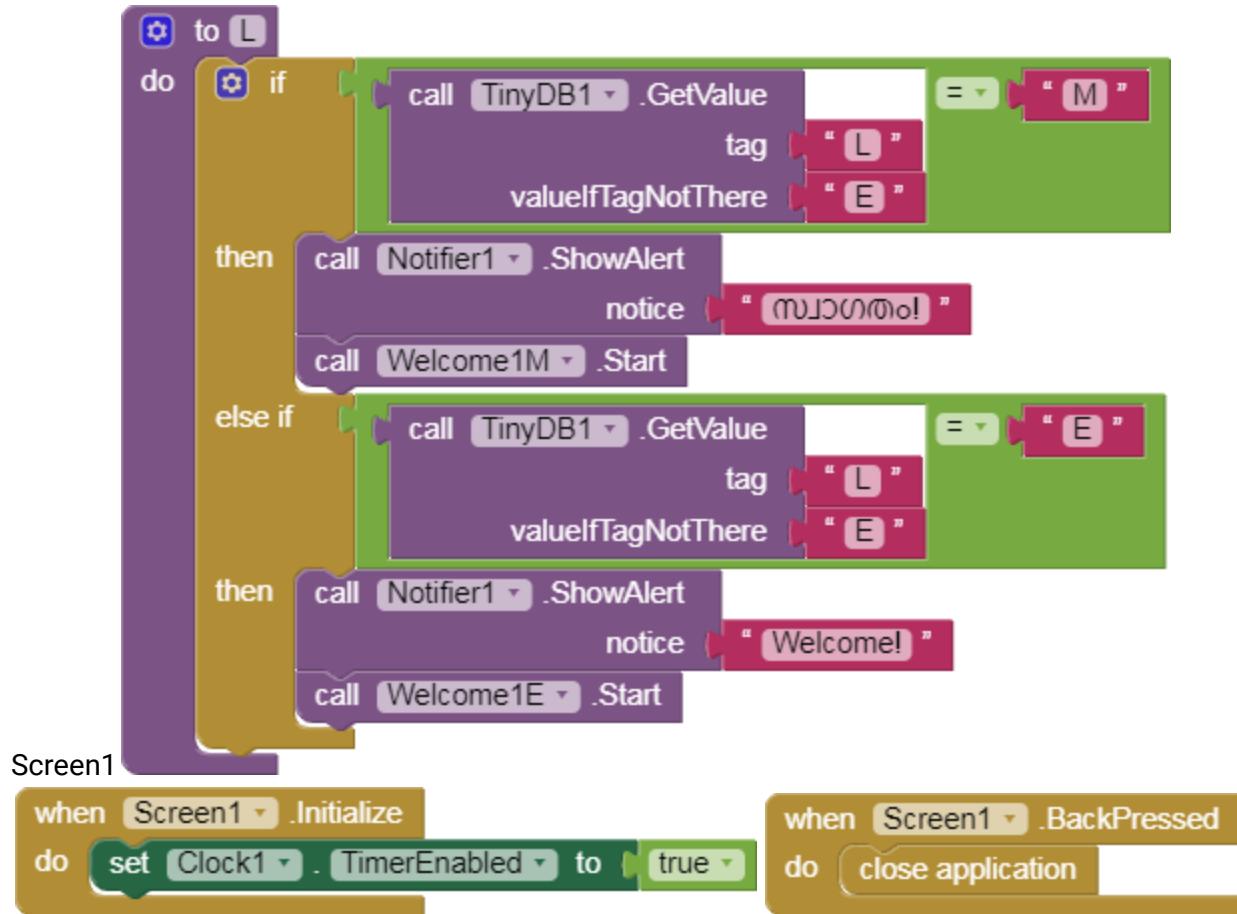
in which the user can add the events of meeting with people, use of any vehicle , and the location change. In this screen using the “Share Tracker History” button the user can share the file containing the activity details.



As said in the previous section, "Don't ask me to sign in later" check box is to reduce the difficulty of the user to sign in always when opening the app.If it is checked after the introduction screen the next screen seen by the user is the “Data Entry” screen. If the checkbox is not checked you should always sign in to enter the “Data Entry” screen.

You can see the text file which includes all details you entered in the app in the internal storage with name Self Tracker Data.txt

## Source Code



```

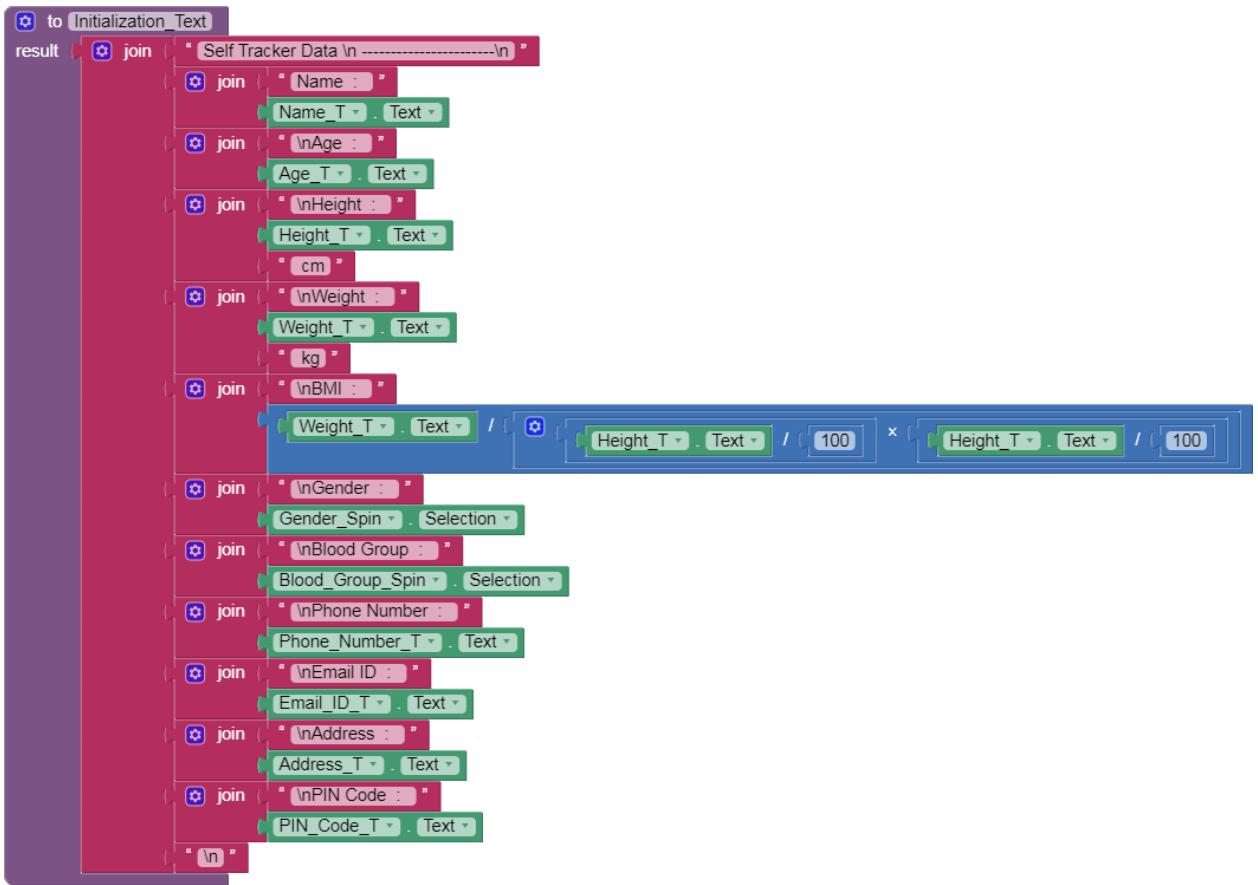
when Clock1 .Timer
do set [Clock1 . TimerEnabled] to [false]
  if [call [TinyDB1 . GetValue] = 0] then
    [tag ["Go Screen"]]
    [valueIfTagNotThere ["0"]]
  then call [File1 . SaveFile]
    [text ["\n"]]
    [fileName ["null.txt"]]
  call [Notifier1 . ShowAlert]
    [notice ["Welcome to Self Tracker, Please register to use ... "]]
  call [Welcome2E . Start]
  open another screen [screenName ["Screen2"]]
else if [call [TinyDB1 . GetValue] = 1] then
  [tag ["Go Screen"]]
  [valueIfTagNotThere ["0"]]
then call [L .]
  open another screen [screenName ["Screen3"]]
else if [call [TinyDB1 . GetValue] = 2] then
  [tag ["Go Screen"]]
  [valueIfTagNotThere ["0"]]
then call [L .]
  open another screen [screenName ["Screen4"]]

when Help_B .Click
do set [Clock1 . TimerEnabled] to [false]
  open another screen [screenName ["Screen6"]]

```

## Screen 2



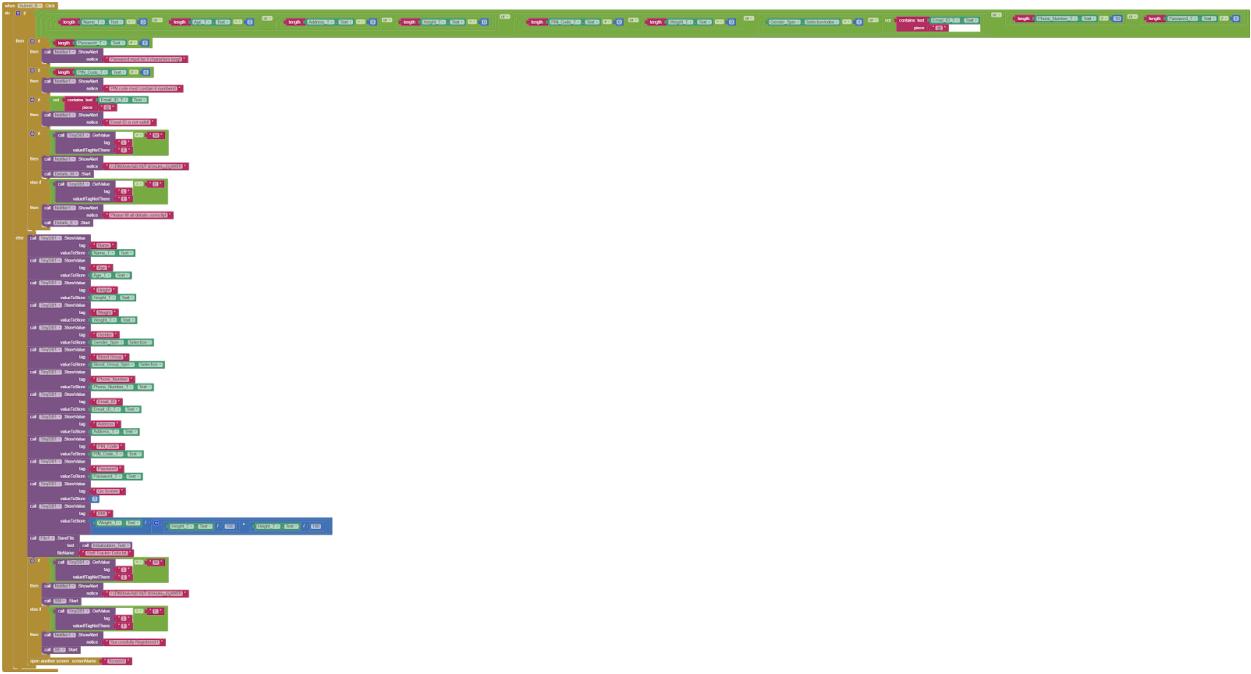


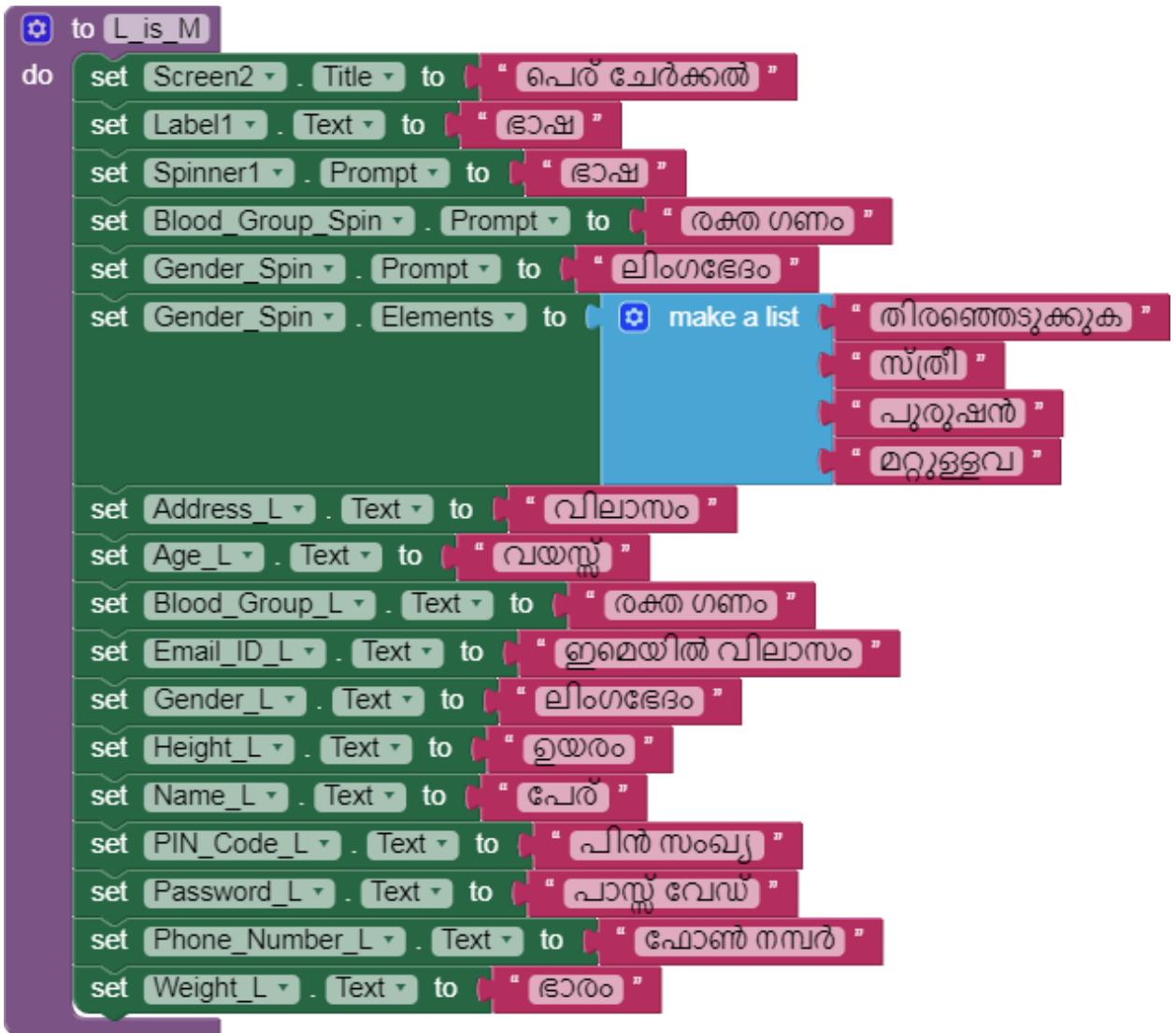
```

when [Spinner1] .AfterSelecting
  selection
    do if [Spinner1 . SelectionIndex = 2]
      then call [TinyDB1 .StoreValue]
        tag ["L"]
        valueToStore ["M"]
        call [L_is_M]
    else if [Spinner1 . SelectionIndex = 3]
      then call [TinyDB1 .StoreValue]
        tag ["L"]
        valueToStore ["E"]
        call [L_is_E]
    end

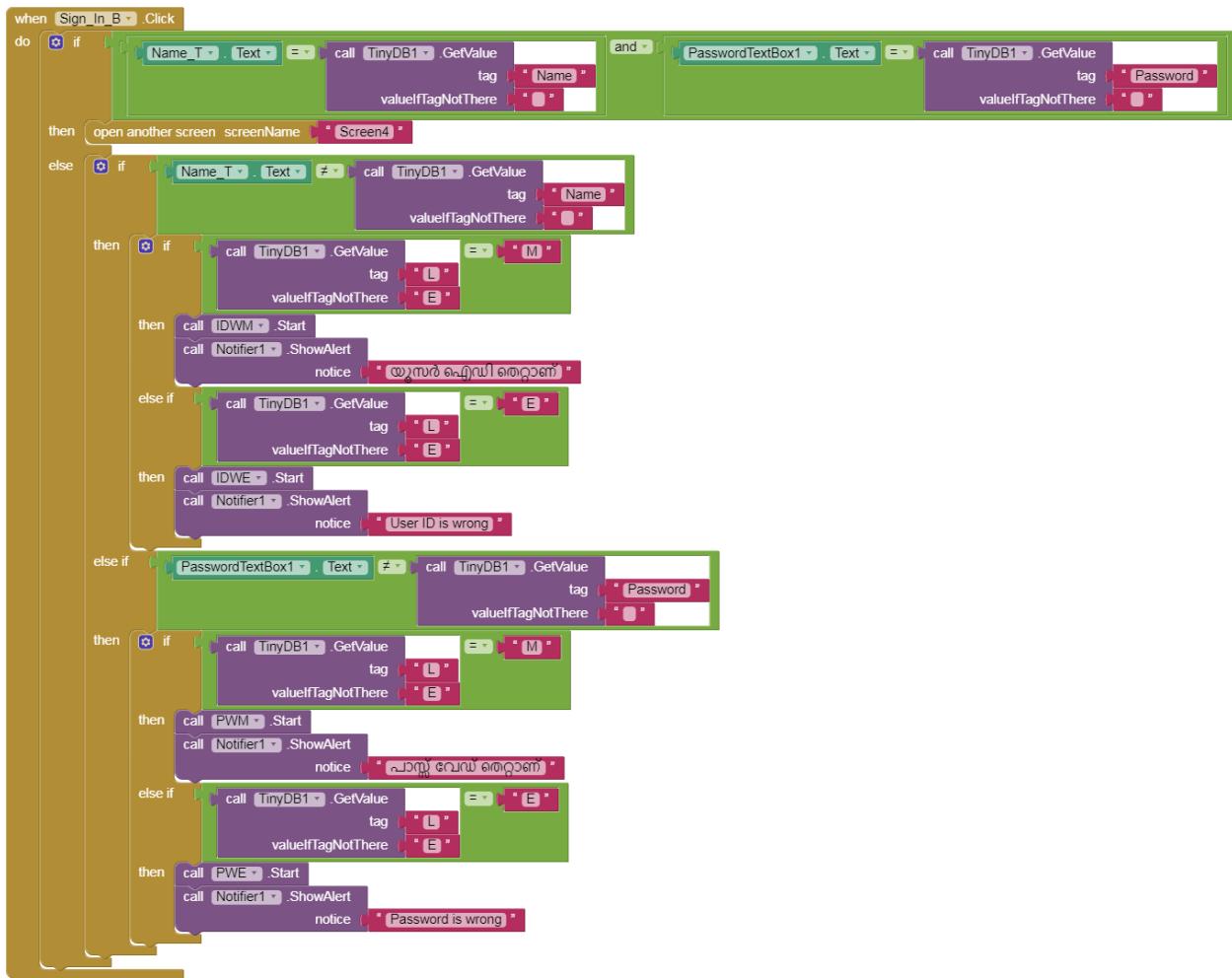
when [Screen2] .Initialize
  do call [TinyDB1 .StoreValue]
    tag ["Go Screen"]
    valueToStore [0]
    if call [TinyDB1 .GetValue]
      tag ["L"]
      valueIfTagNotThere ["E"]
    then call [L_is_M]
    else if call [TinyDB1 .GetValue]
      tag ["L"]
      valueIfTagNotThere ["E"]
    then call [L_is_E]
  end

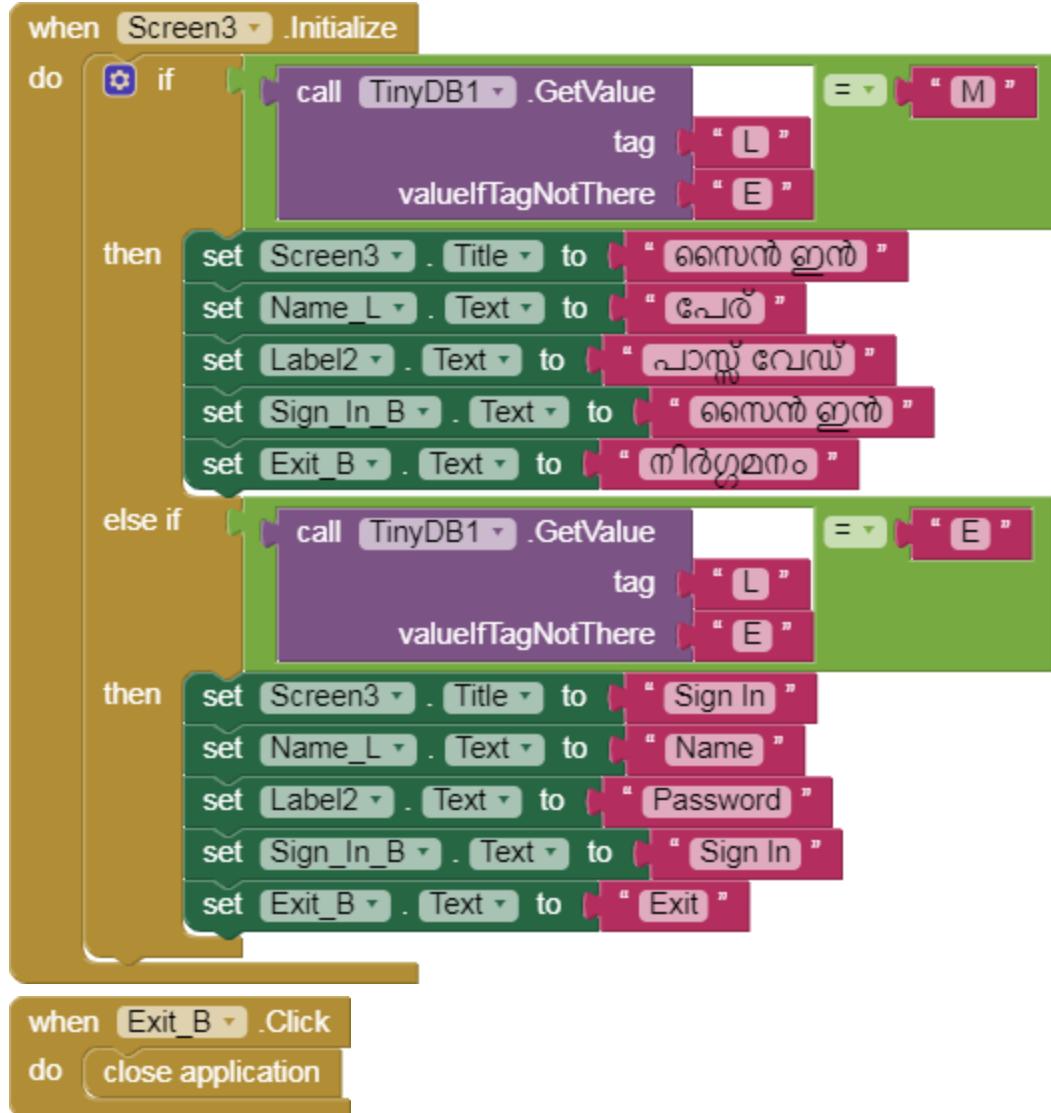
```



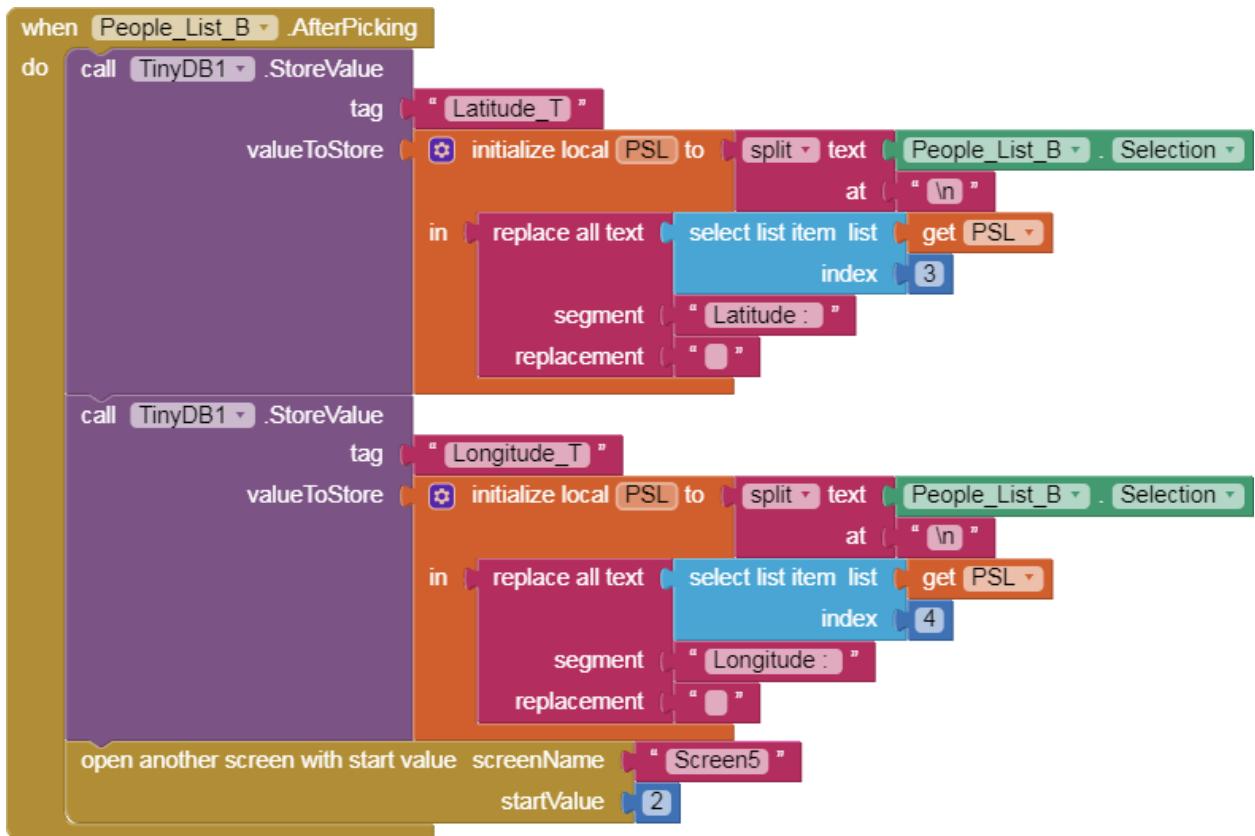


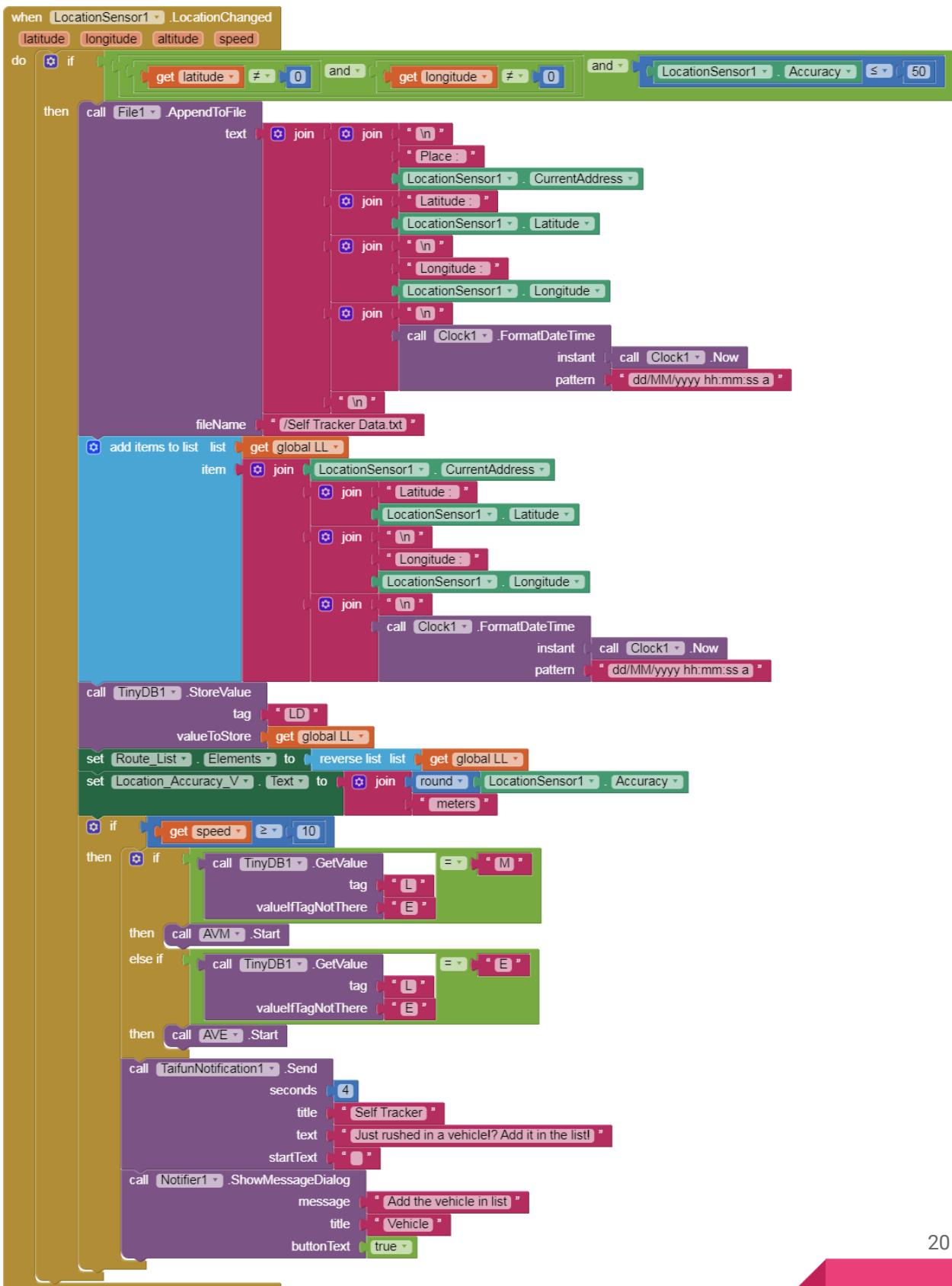
## Screen 3

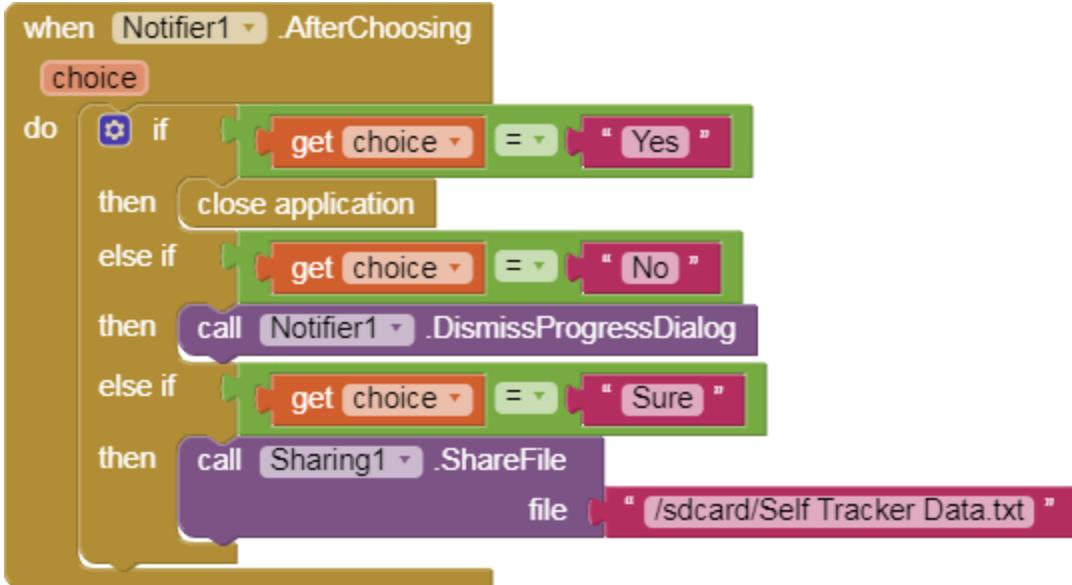


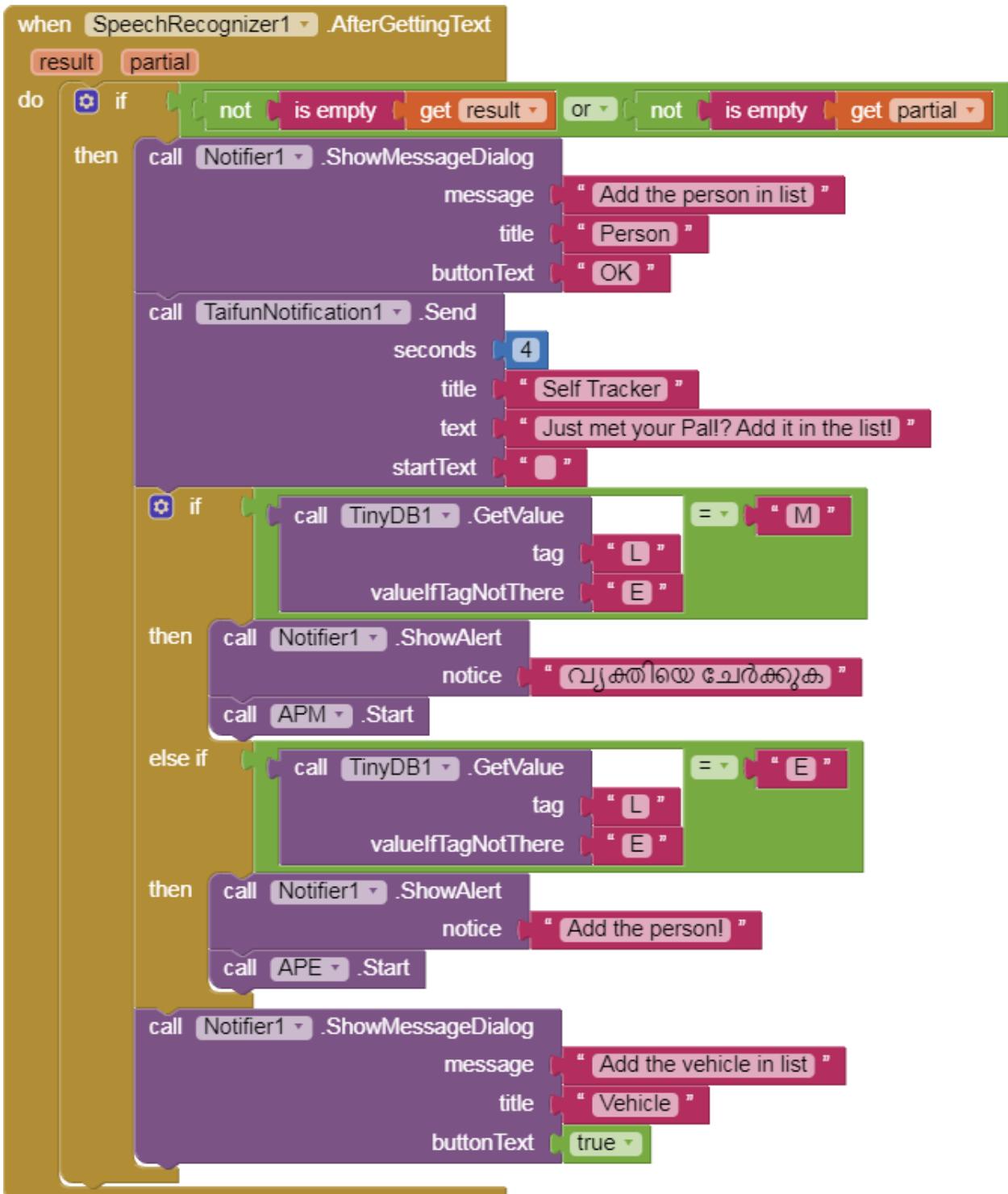


## Screen 4

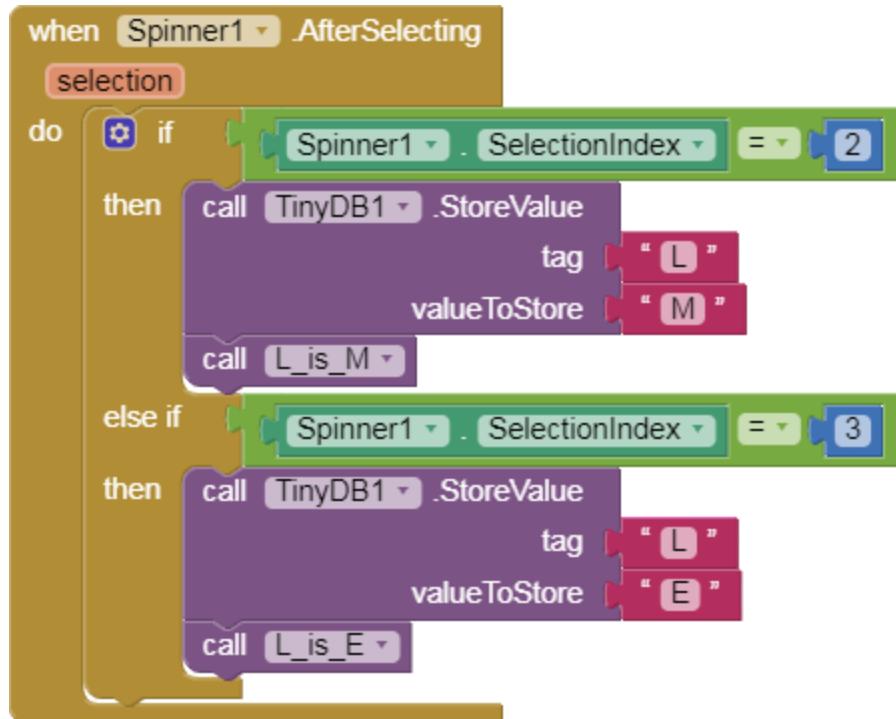








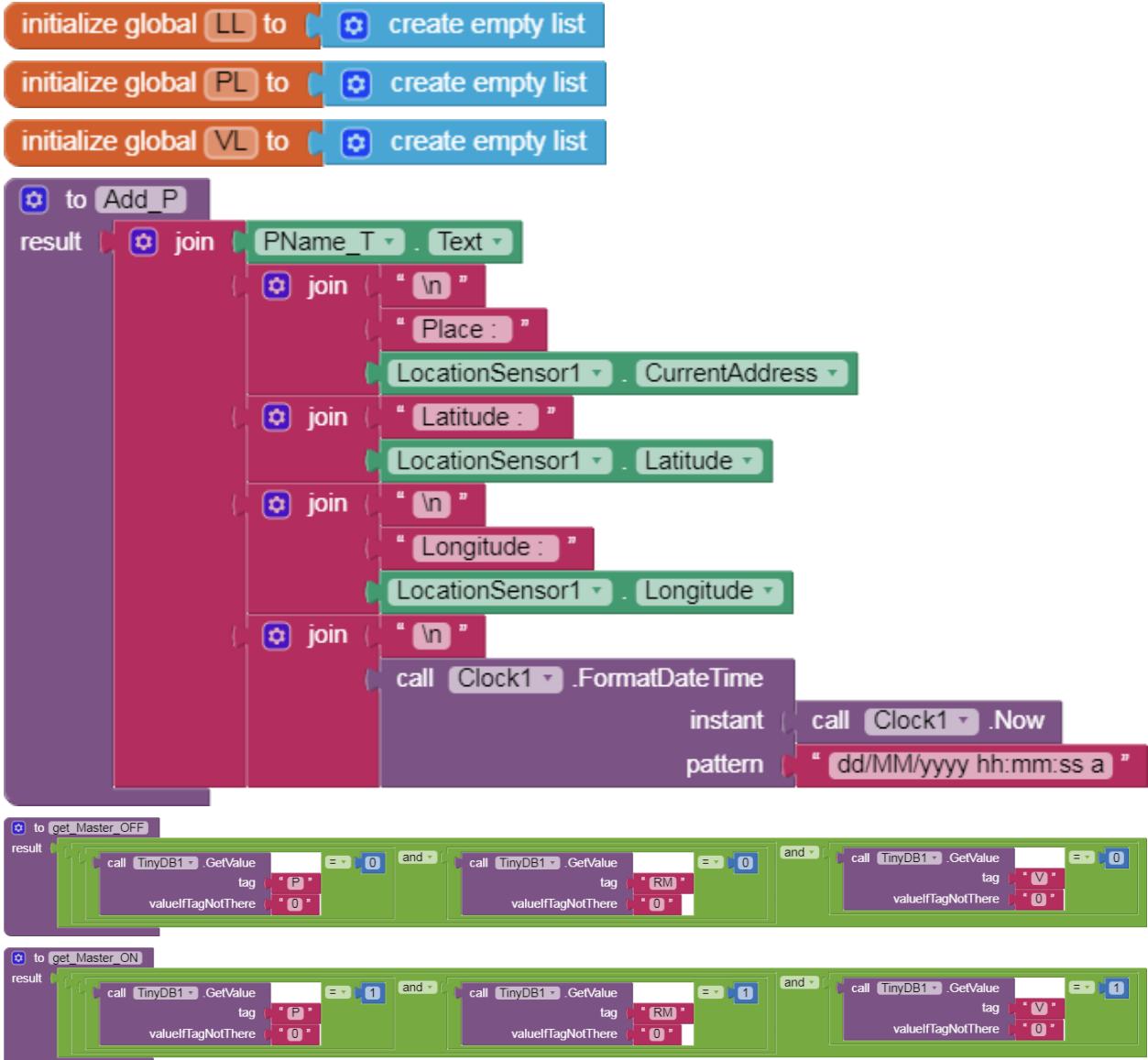


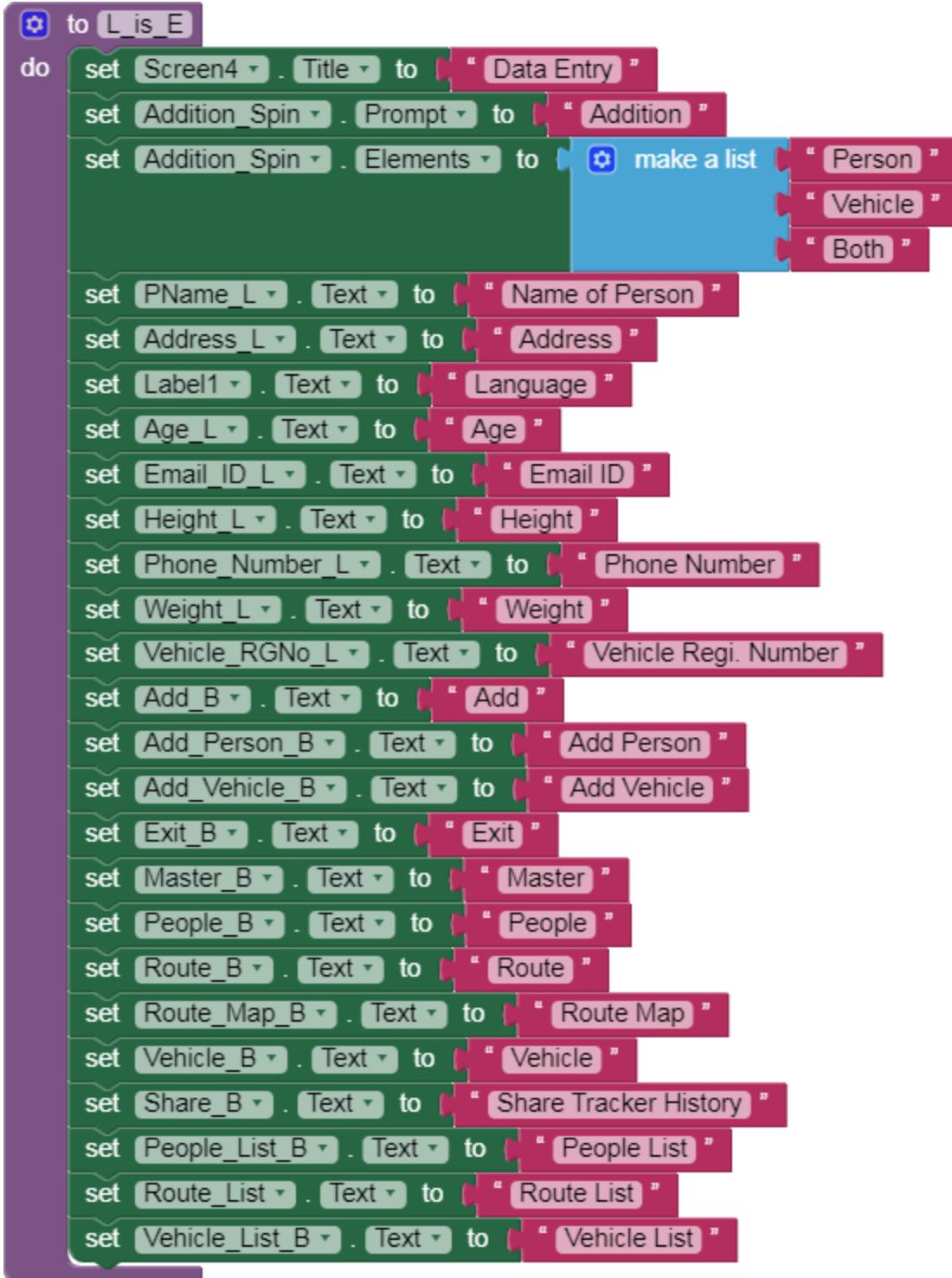


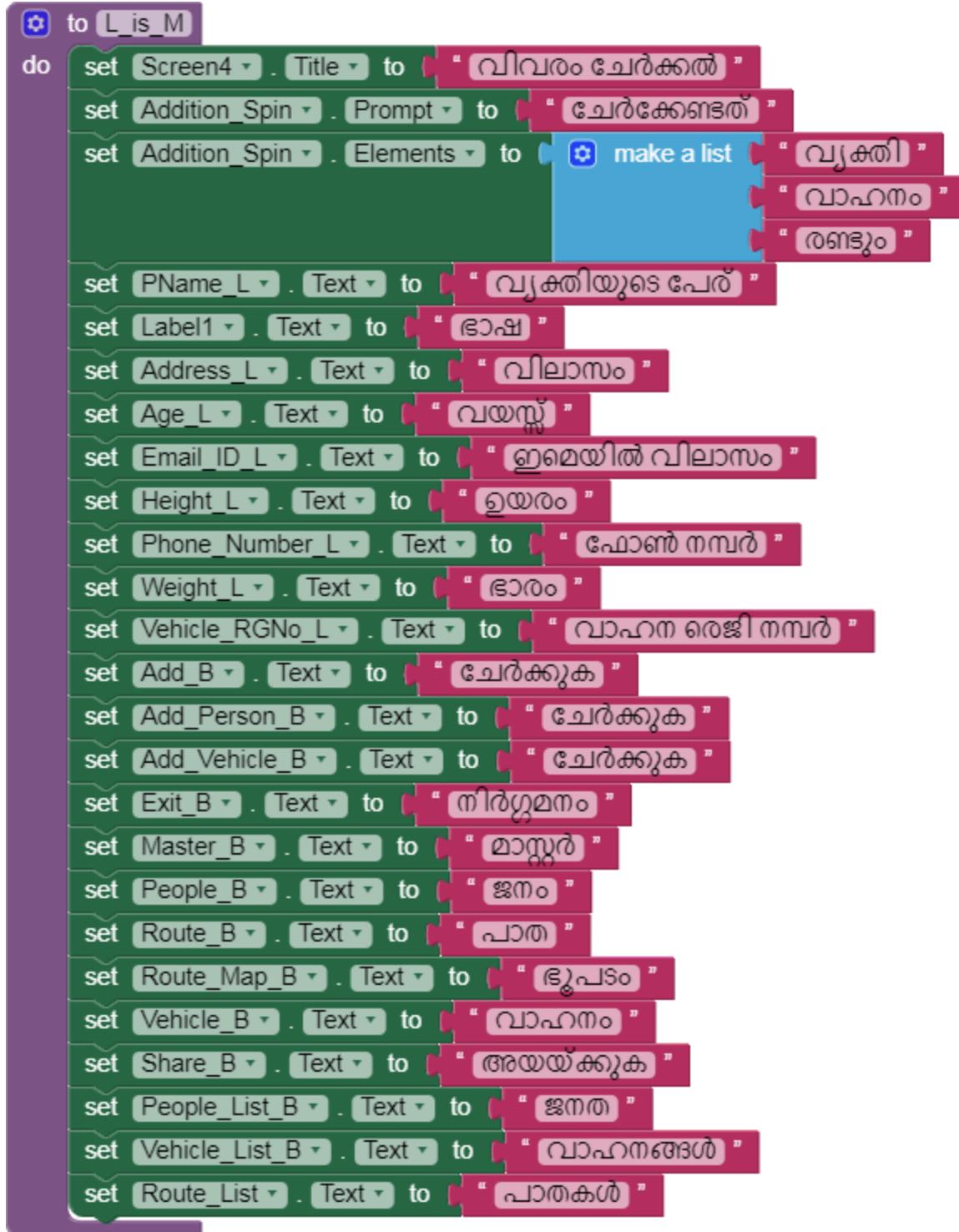
```

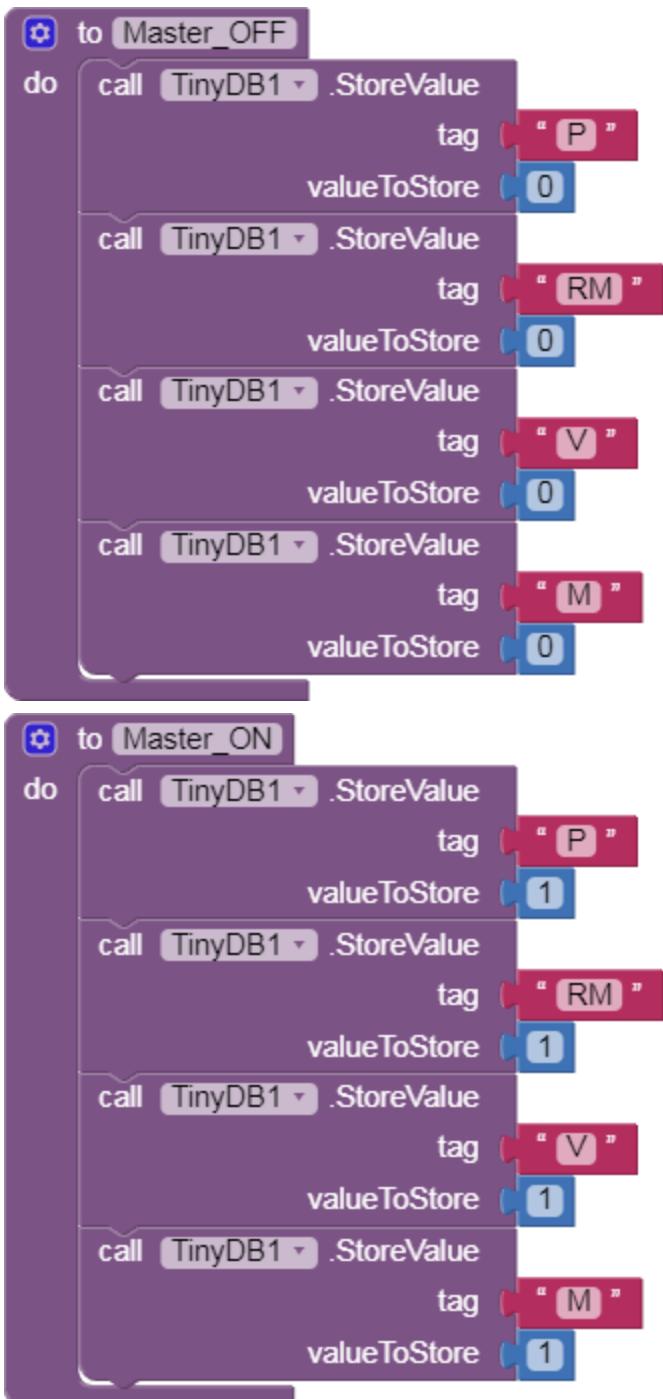
when Screen4 .Initialize
do
  set People_List_B .Elements to reverse list list
    call TinyDB1 .GetValue
      tag " PD "
      valueIfTagNotThere [create empty list]
  set Vehicle_List_B .Elements to reverse list list
    call TinyDB1 .GetValue
      tag " VD "
      valueIfTagNotThere [create empty list]
  set Route_List .Elements to reverse list list
    call TinyDB1 .GetValue
      tag " LD "
      valueIfTagNotThere [create empty list]
  set Name_V .Text to call TinyDB1 .GetValue
    tag " Name "
    valueIfTagNotThere [ ]
  set Address_V .Text to call TinyDB1 .GetValue
    tag " Address "
    valueIfTagNotThere [ ]
  set Age_V .Text to call TinyDB1 .GetValue
    tag " Age "
    valueIfTagNotThere [ ]
  set Blood_Group_V .Text to call TinyDB1 .GetValue
    tag " Blood Group "
    valueIfTagNotThere [ ]
  set Email_ID_V .Text to call TinyDB1 .GetValue
    tag " Email_ID "
    valueIfTagNotThere [ ]
  set Gender_V .Text to call TinyDB1 .GetValue
    tag " Gender "
    valueIfTagNotThere [ ]
  set Height_V .Text to call TinyDB1 .GetValue
    tag " Height "
    valueIfTagNotThere [ ]
  set Name_V .Text to call TinyDB1 .GetValue
    tag " Name "
    valueIfTagNotThere [ ]
  set Phone_Number_V .Text to call TinyDB1 .GetValue
    tag " Phone_Number "
    valueIfTagNotThere [ ]
  set Weight_V .Text to call TinyDB1 .GetValue
    tag " Weight "
    valueIfTagNotThere [ ]
  set Details_VA1 .Visible to true
  set Name_Vehicle_HA .Visible to false
  set Add_People_VA .Visible to false
  set Add_Vehicle_VA .Visible to false
  if [call TinyDB1 .GetValue tag " Go Screen " = 1]
    then
      [1]

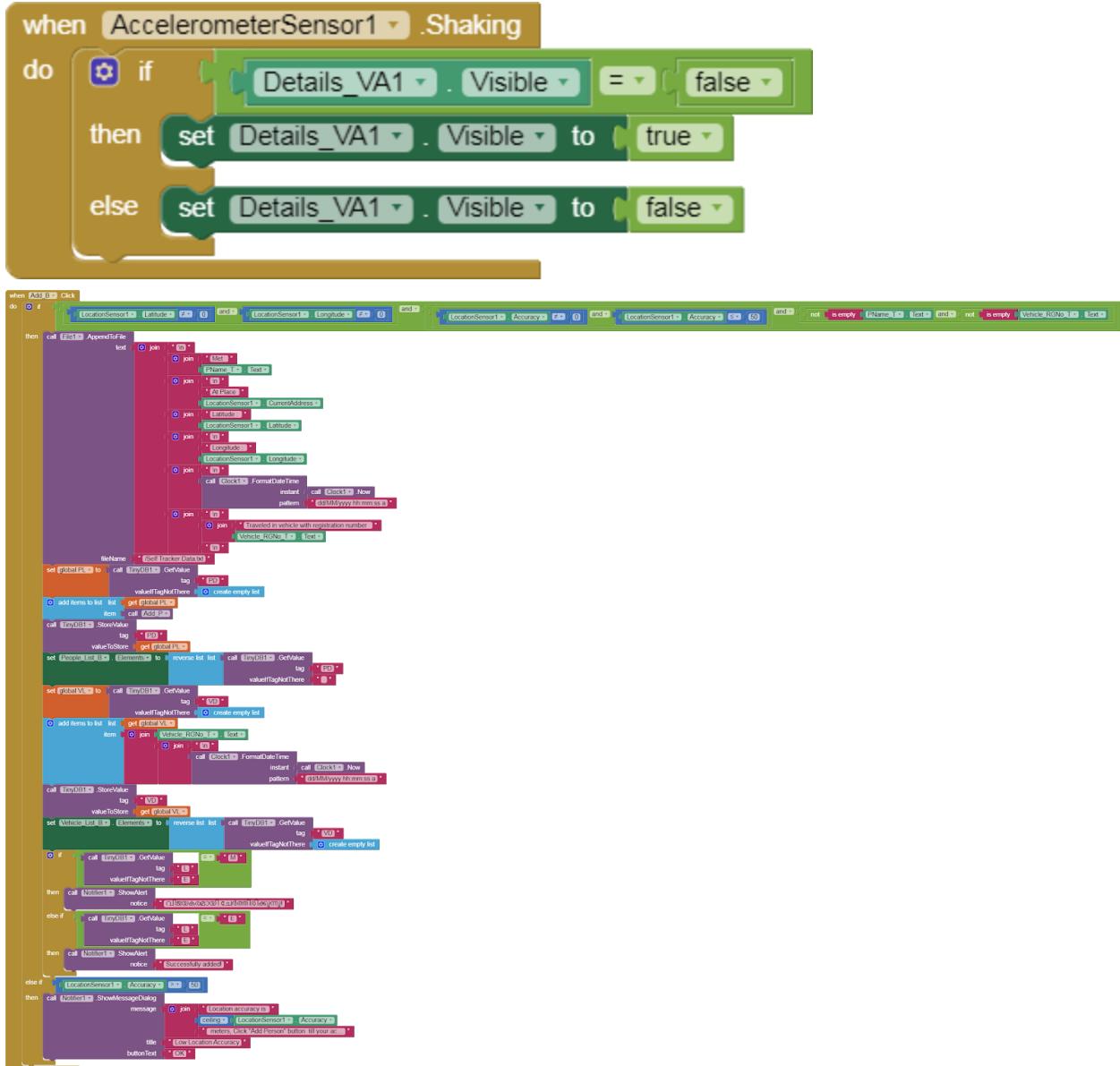
```

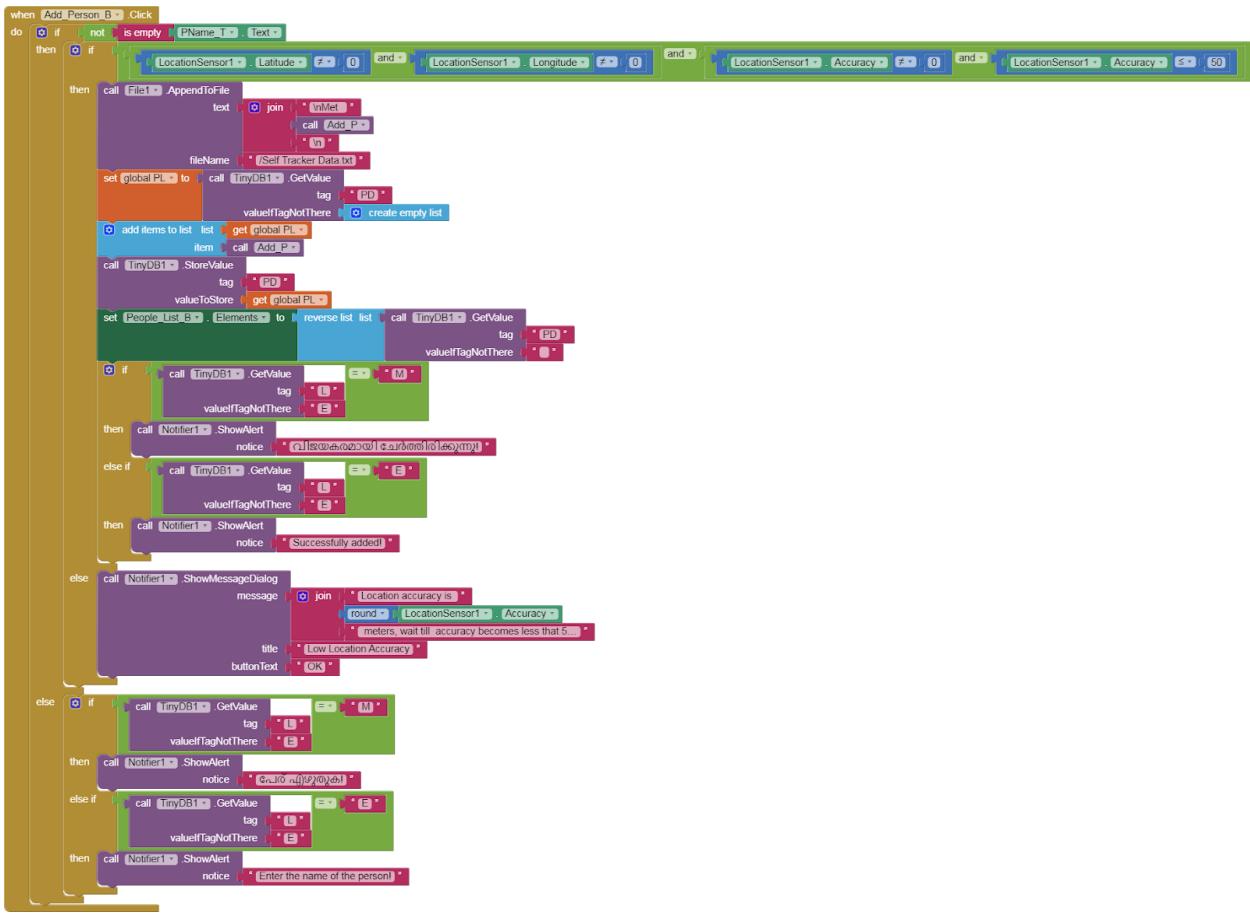


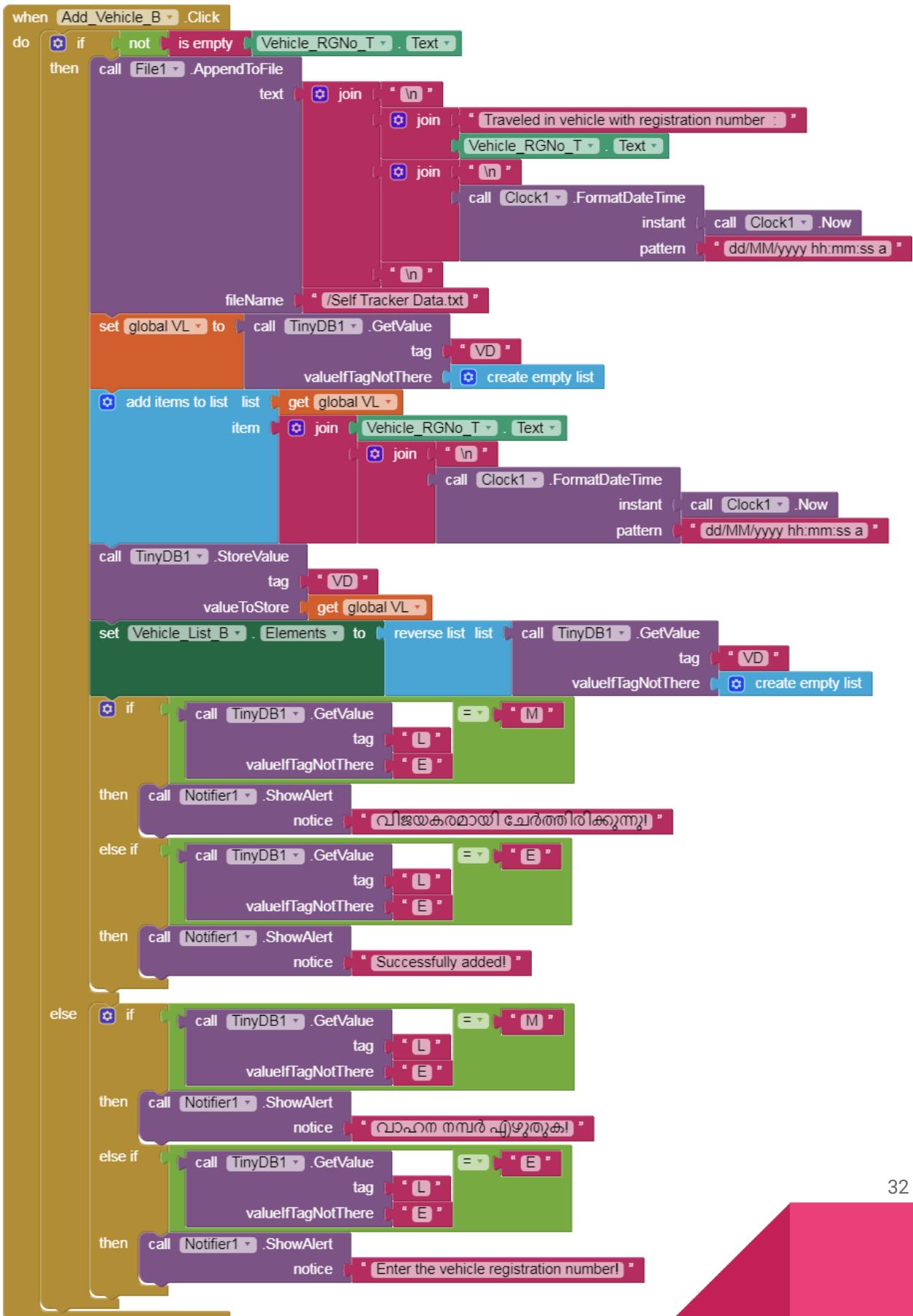


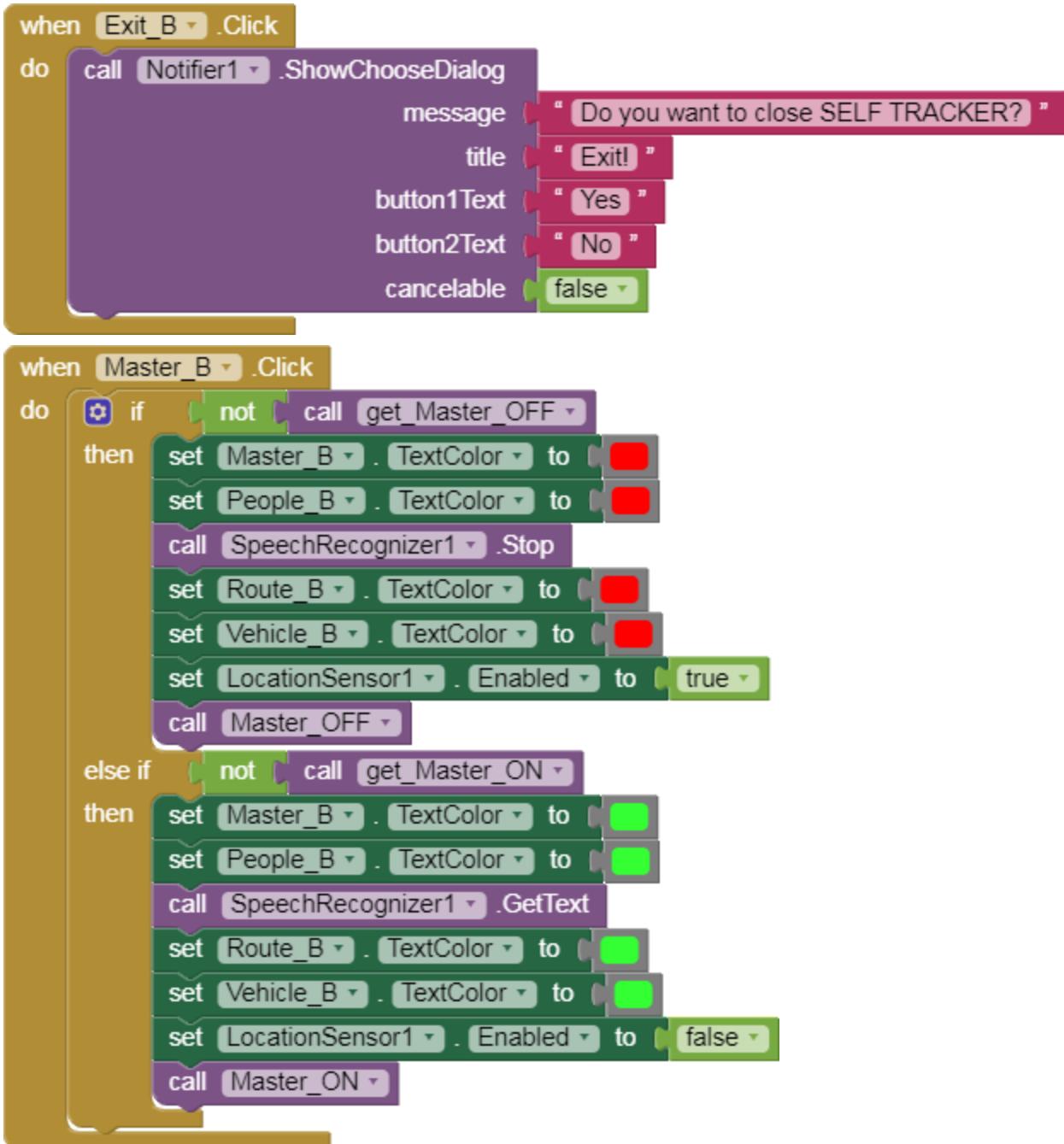


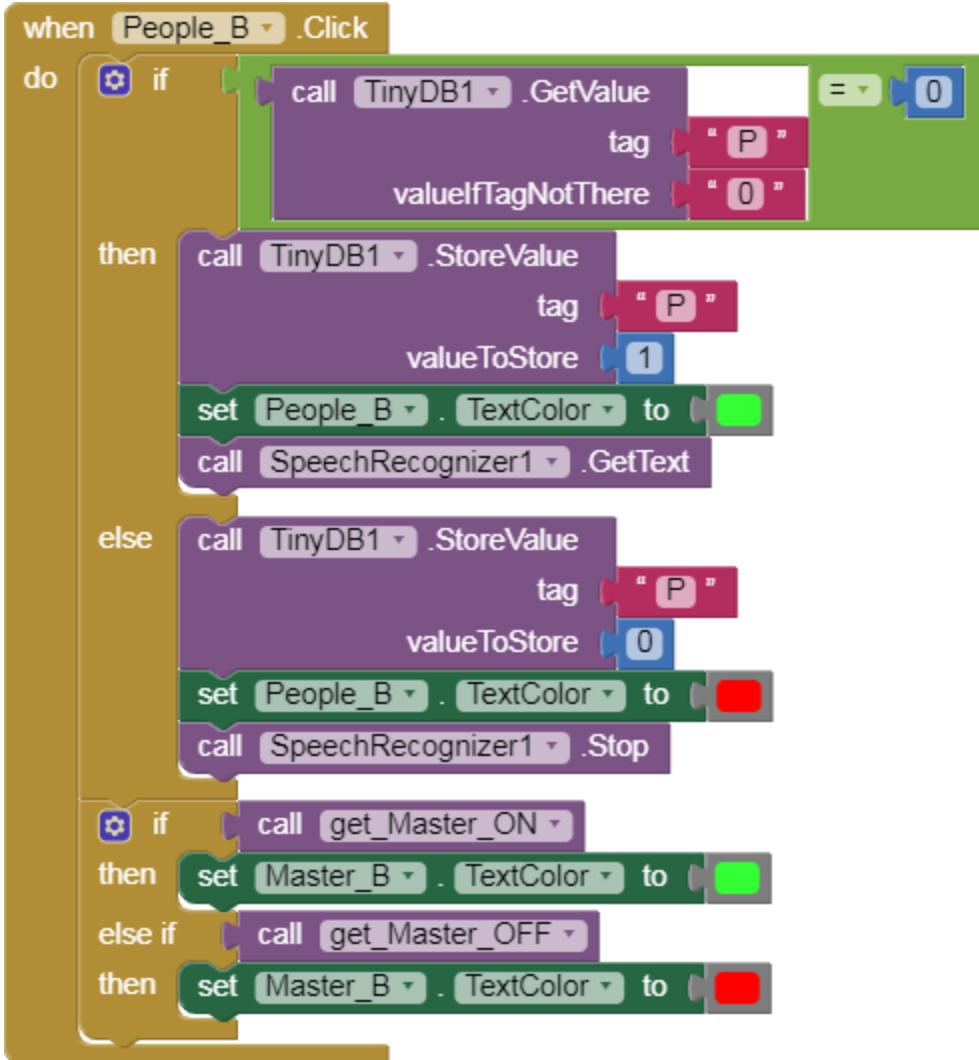








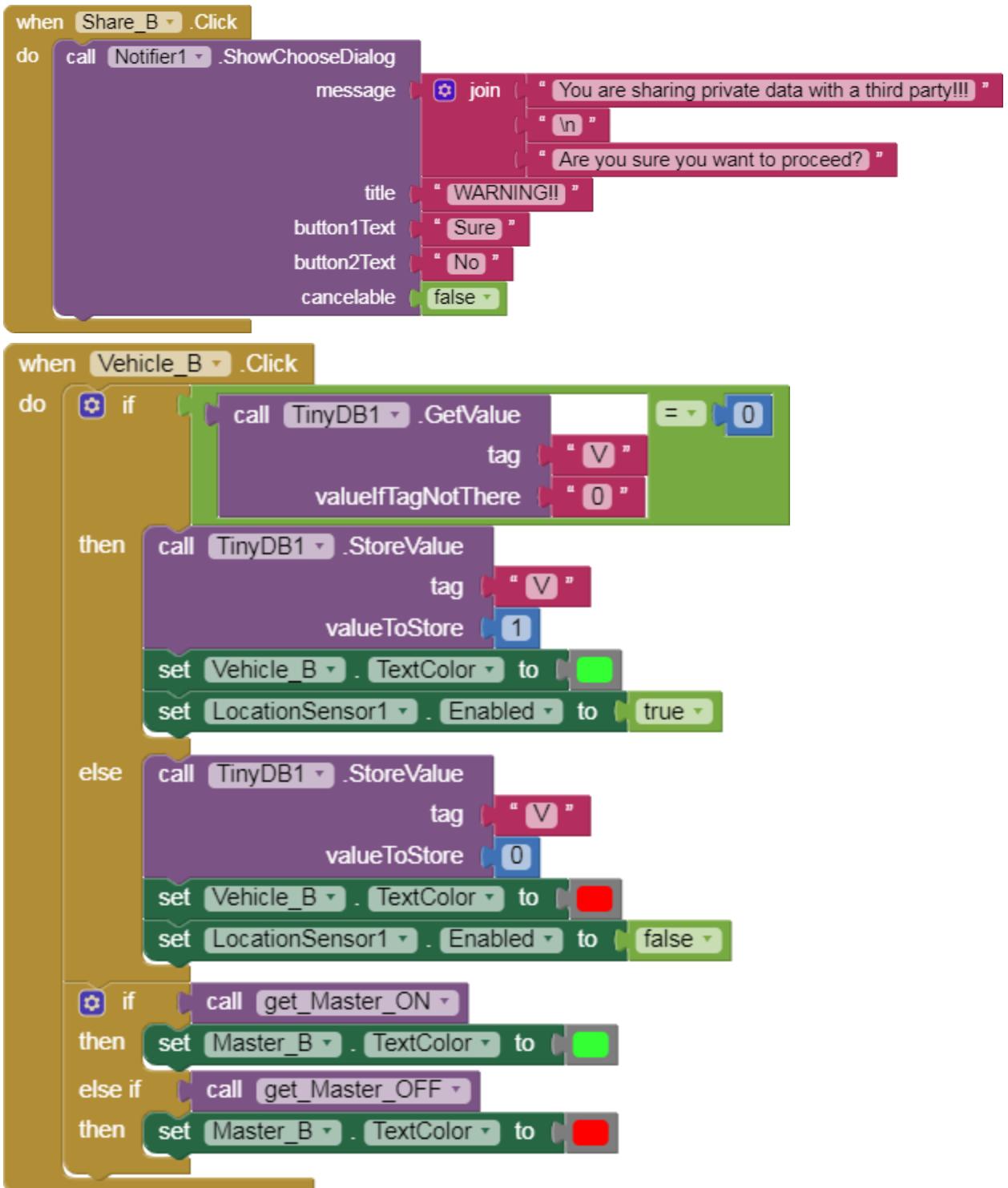


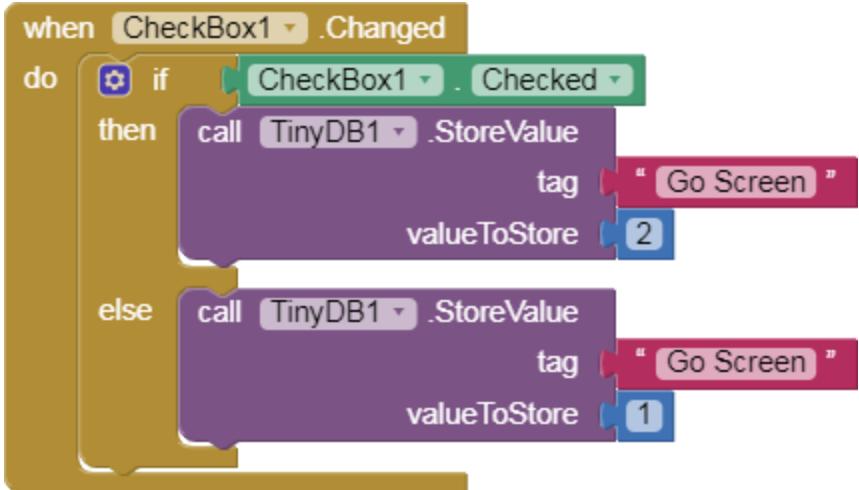


```

when Route_B .Click
do
  if call TinyDB1 .GetValue
    tag "RM"
    valueIfTagNotThere "0"
  then
    call TinyDB1 .StoreValue
      tag "RM"
      valueToStore 1
    set (Route_B .TextColor) to green
    set (LocationSensor1 .Enabled) to true
  else
    call TinyDB1 .StoreValue
      tag "RM"
      valueToStore 0
    set (Route_B .TextColor) to red
    set (LocationSensor1 .Enabled) to false
  if call get_Master_ON
  then
    set (Master_B .TextColor) to green
  else if call get_Master_OFF
  then
    set (Master_B .TextColor) to red
when Route_Map_B .Click
do
  open another screen with start value
    screenName "Screen5"
    startValue 1

```





## Screen 5

```

when Screen5 .Initialize
do
  if [get start value = 1]
    then call WebViewer1 .GoByUrl
      url " http://www.google.com/maps/timeline "
  else if [get start value = 2]
    then call WebViewer1 .GoByUrl
      url [join " https://www.google.com/maps/@"
        call TinyDB1 .GetValue
          tag " Latitude_T "
          valueIfTagNotThere " 0 "
        call TinyDB1 .GetValue
          tag " Longitude_T "
          valueIfTagNotThere " ,21z "
      ]
  if [call TinyDB1 .GetValue = " M "]
    tag " L "
    valueIfTagNotThere " E "
    then set Button1 .Text to " مکان "
      set Screen5 .Title to " مکان "
  else if [call TinyDB1 .GetValue = " E "]
    tag " L "
    valueIfTagNotThere " E "
    then set Button1 .Text to " Back "
      set Screen5 .Title to " Map "
when Button1 .Click
do open another screen screenName " Screen4 "
when Screen5 .BackPressed
do open another screen screenName " Screen4 "

```

## Screen 6

```

when Screen6 .Initialize
do if call TinyDB1 .GetValue tag = " M "
    valueIfTagNotThere " L "
    " E "
then set Button1 .Text to " ഫോറ്റ് "
set Screen6 .Title to " മാർഗ്ഗവേദി "
else if call TinyDB1 .GetValue tag = " E "
    valueIfTagNotThere " L "
    " E "
then set Button1 .Text to " Back "
set Screen6 .Title to " Guide "
when Screen6 .BackPressed
do open another screen screenName " Screen1 "
when Button1 .Click
do open another screen screenName " Screen1 "

```

Thank You