

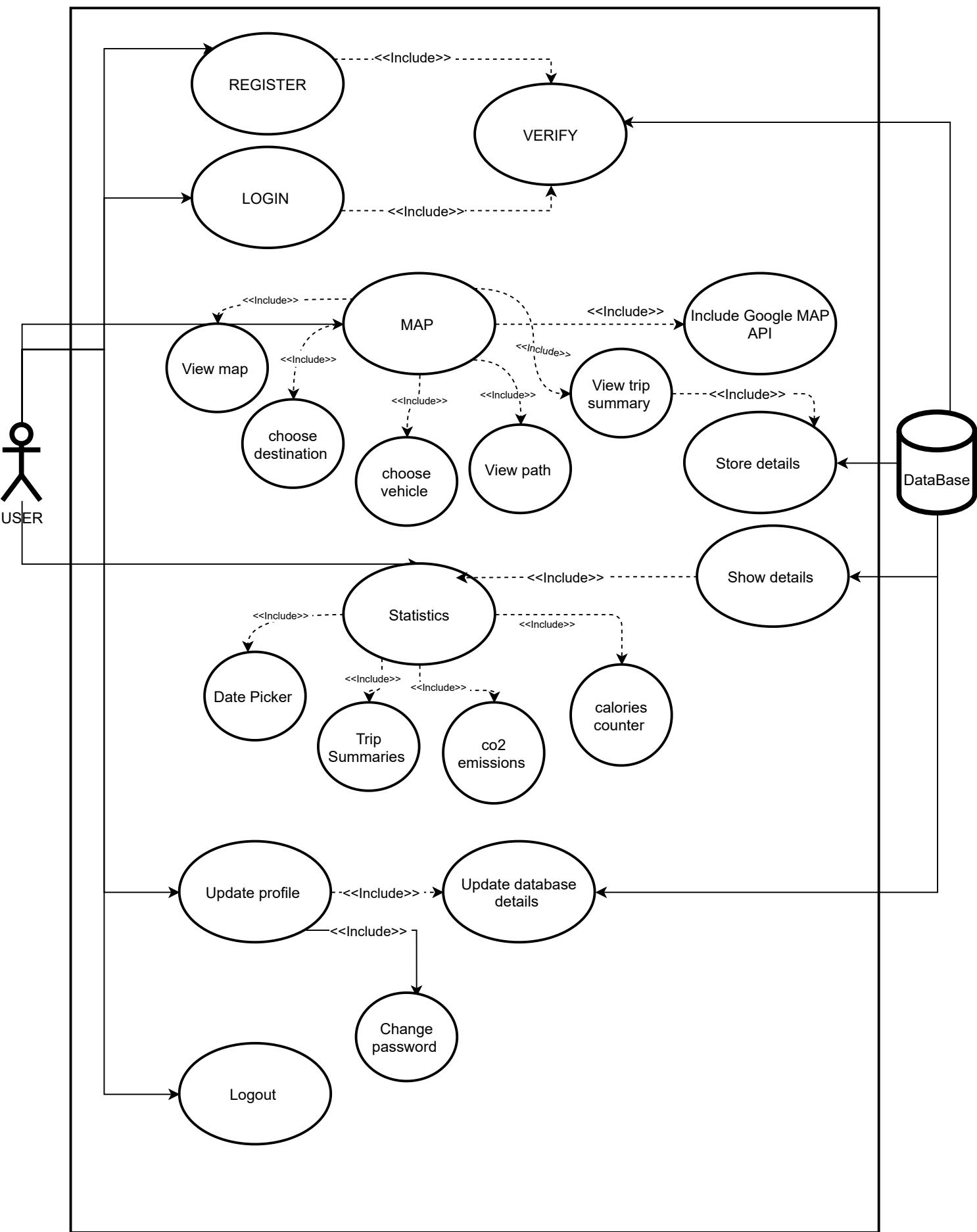
Appendix A

Extra Information

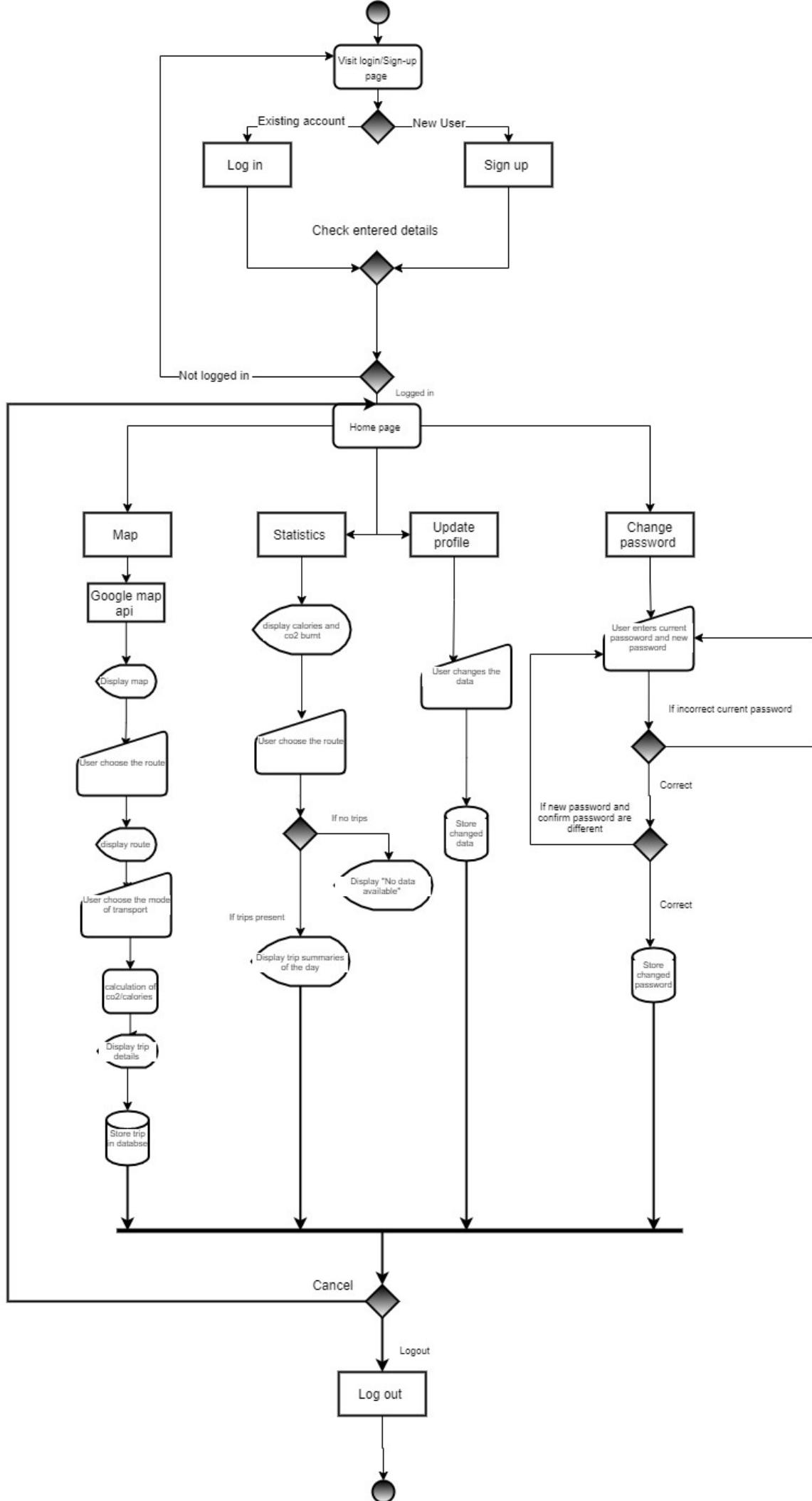
A.1 Software Architecture

Use Case Diagram

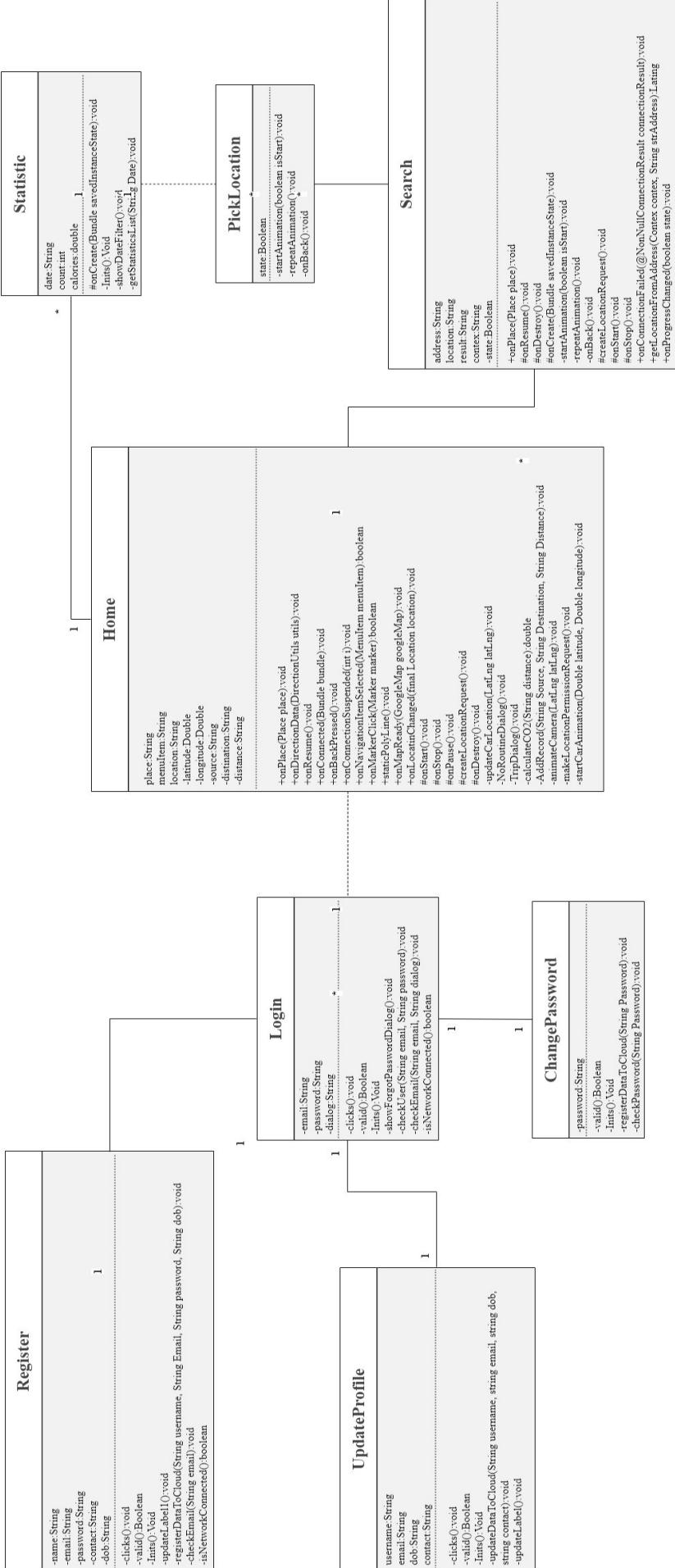
PROENV-Y



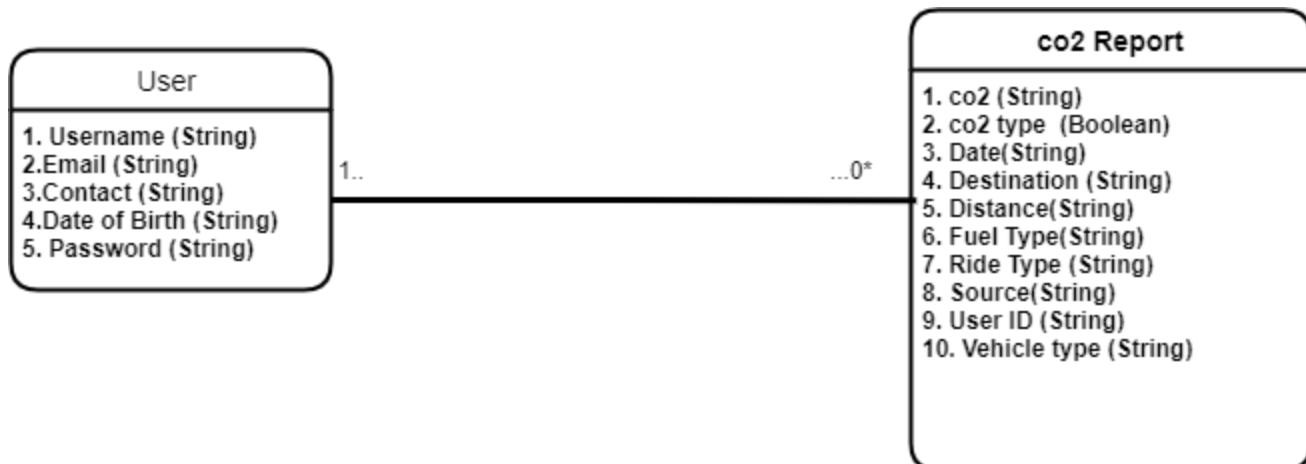
Activity Diagram



Class Diagram

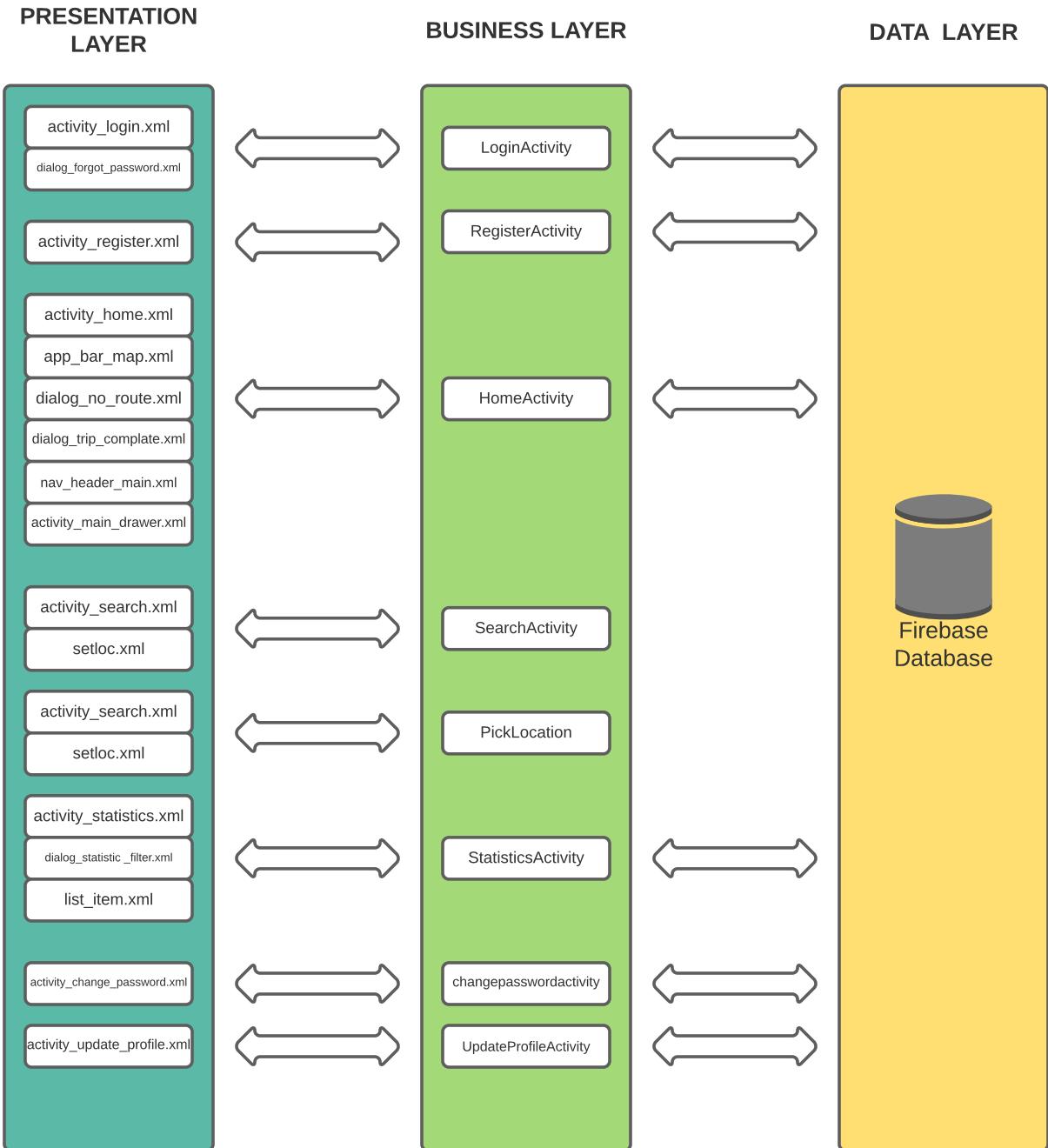


ER Diagram



Three Tier Architecture

THREE TIER ARCHITECTURE



A.2 Design

Personas



Matt Johnson

AGE: 15

GENDER: MALE

LOCATION: LONDON

WORK : STUDENT AT
KINGS CHRISTIAN
SCHOOL

"I need to curb down my CO2 emissions for a better tomorrow and healthy environment for my grandpa".

Goals :

1. Find out a way to shorten the route all the way to school by collecting all the children from one point to reduce the carbon emission
2. Substitute the bus with a cycle for a healthier lifestyle as well as reducing bus travel

Needs :

1. An easy to use, tool that helps him calculate the carbon emission of different transports
2. A tool that gives him alternatives to travel which will reduce emissions of toxins
3. A feeling of concern for the environment and the other generations
4. A feeling of giving back to the society by helping the cause

Matt, a 15 year old boy travels to school everyday via carpool. His carpool is a bus that picks up all the children in his neighbourhood at a specific time and drops them off to school which is around 35 minutes away. On the way, the bus stops multiple times to pick up children and takes longer routes to cover the entire neighbourhood. This increases the emission of carbon from the long journeys everyday as the bus also makes a second round of travel to drop the children back home. Matt, who studies in the 9th grade has recently read about carbon emissions and toxins that transport such as cars and buses give out leading to acute air pollution. This pollution then causes severe health issues like asthma and chest pains, especially in the older generation and since Matt lives in a joint family, with his grandparents, he is very concerned about their health.

Scenario:

Matt often takes the carpool to school but sometimes he is unable to catch the bus so early in the morning because of oversleeping or losing track of time and because of this, his father ends up driving him to school in their personal car. While returning from school, Matt has extra curricular activities for which he has to stay in school till late and the carpool is gone by then. He then has to call for his car again which picks him up from school and drops him off back home.

Matt, being concerned about the carbon emissions polluting the environment, wants to come up with a way through which he can calculate the amount of carbon emissions he uses up per week and then look for solutions to reduce the same. He has decided to make an effort to not miss his carpool in the morning and also cycle to school, if needed.

This will be extremely helpful to him as the tool will also have a calorie counter through which Matt can keep track of how much exercise he needs in a day. This will help him reach a healthy lifestyle as well as protect the environment. He is also quite driven to extend the same knowledge to his other classmates and encourage them to reduce carbon emissions by doing their part.



Julien Foster

AGE: 37

GENDER: FEMALE

LOCATION: LONDON

WORK : FINANCIAL
ANALYST AT JP
MORGAN

shutterstock.com • 655180042

"It is also cheaper to take eco-friendly mode of transport and thus helps me saves more extra money".

Goals :

1. Reducing the amount of times the car is driven by sending the children to school via tube/bus or cycles
2. Developing a concern for pollution in the society

Needs :

1. An app that will alert her every time her car exceeds the carbon emission average of the previous week
2. A calorie counter to track how much she burns in a day by merely walking or running.

Julien, a single parent of 2 children aged 11 and 14, and a financial analyst at JP Morgan, is a 37 year old woman who lives in London. She is also a part time health advisor at the gym in her neighbourhood as she is very keen on helping people live a healthy lifestyle like she does. She drives a car to drop off her children to school in the morning and then drives herself to work which is 20 minutes away from the school. However, she has to come back home, before going to the office to prepare for her meetings and cook for herself and the children for the day and the travel time from the house to the office takes about 30 minutes. She sometimes has to drive to grocery stores to shop for meals and daily products and also has to care for her children's needs. Her neighbour's children also go to the same school, however they don't talk very often and are not very sociable. They travel by public transport and are very used to it.

Scenario:

Julien, being extremely health conscious, needs an app that will monitor her calories burnt every time she steps out of the house to run, cycle or walk. In order to reduce carbon emissions, she needs to encourage her children to take public transport such as the tube or the bus along with her neighbours children if she is worried about their safety. They could also cycle together to school as the travel time is not more than 15 minutes and renting cycles in London is very convenient. Moreover, through the app calculator she can keep track of how much carbon her car is emitting and try reducing the same by walking or cycling to the grocery store. She is also keen to provide this app to her children who can keep track of the carbon emission every time they travel by transport. Moreover, she wants to spread the word to her colleagues and clients at the part time job where she works as an advisor so that they contribute to the society as well. She is also determined to instill such values in her children as well. She also noticed that not only taking eco-friendly modes of transport is beneficial for the planet but also budget friendly, allowing her to save up money.



Ricky Davis

AGE: 25

GENDER: MALE

LOCATION: AMSTERDAM

WORK :
ENVIRONMENTALIST
AND CYCLE SHOP
OWNER

"Cycling produces no co2 emissions and also helps in maintaining a healthy life".

Goals :

1. To reduce pollution in his residing city for a better sustainability
2. To promote and come up with alternative methods of travelling to reduce toxin emission

Needs :

1. Cycles and other sustainable transport at a cheap price for majority population
2. An app that provides exact calculation of pollution spread by an individual

Ricky, a 25 year old environmentalist at a firm as well as a cycle salesman, is constantly analysing the pollution levels in his city. He noticed that the toxins from motor vehicles are increasing day by day as people are buying cars constantly or upgrading their old cars. He is determined to help to reduce the pollution by taking extreme measures to change his lifestyle and those around him. Ricky does not own a car and usually walks to places or rents a cycle.

Scenario:

Ricky being a cycle salesman, promotes sustainability by selling cycles at a price lower than market value. This he does, to promote a better environment for the future generation and in return just wants the customers to download the app that calculates pollution level per person. He is extremely concerned about the environment and wants the future generation to follow in his footsteps. He spreads awareness about pollution levels in his neighborhood by putting up fliers for a better tomorrow and promoting his ideas all around. He himself does not own a car and travels everywhere on his cycle.



Morgan Freeman

AGE: 75

GENDER: MALE

LOCATION: ROTTERDAM

WORK : RETIRED

"I wish to spread awareness about the global warming for a healthy future of my children".

Goals :

- 1.A neighbourhood with next to nil pollution through cars and motor vehicles
- 2.To spread more awareness in the neighbourhood to decrease pollution levels keeping in mind the old residents.

Needs :

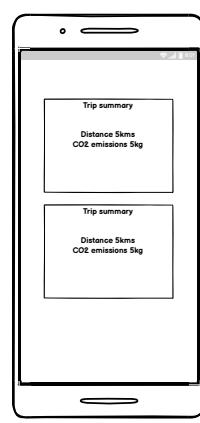
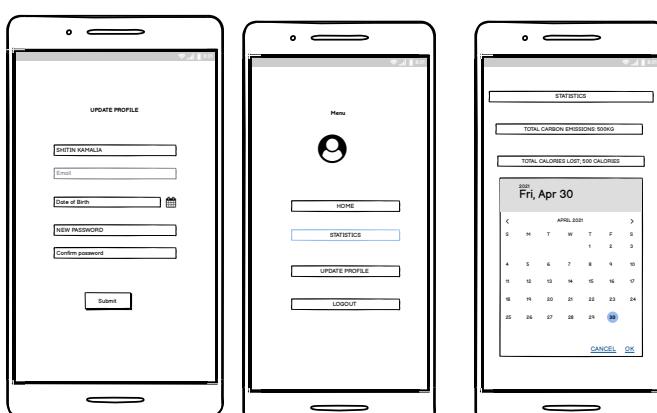
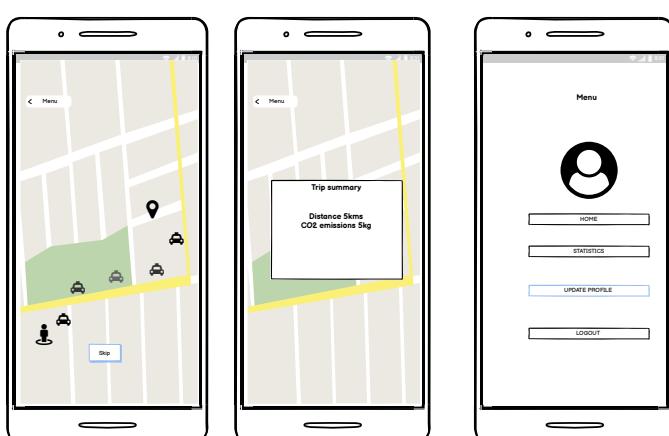
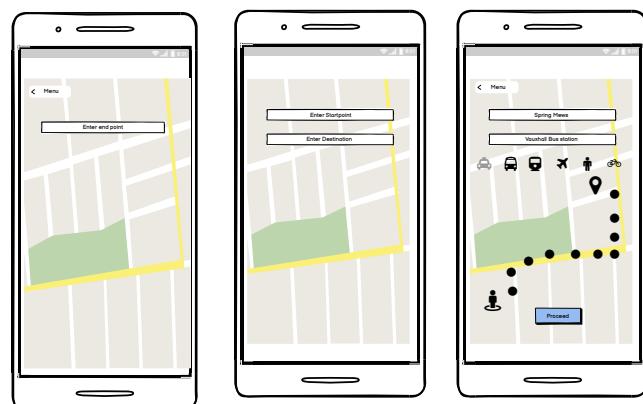
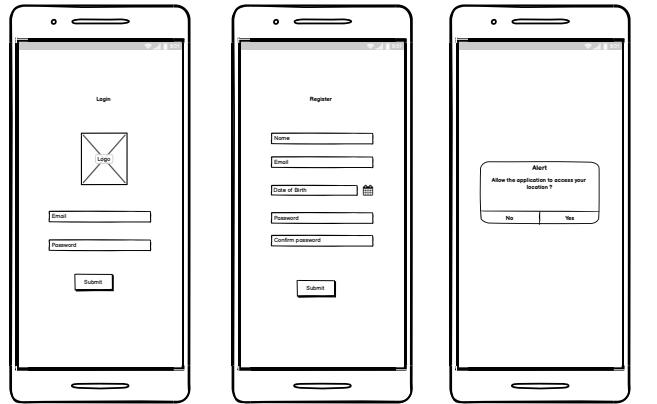
- 1.An app to calculate the pollution level in the neighbourhood for every individual
- 2.Constant reminder to check the pollution level and group commitment to find ways to reduce it like neighbourhood carpooling.

Morgan, a 75 year old retired businessman lives in Rotterdam with his wife who is also 75 years old. Both their children work and are married in different cities. Jordan's wife has breathing problems because of asthma and her lungs are damaged because of breathing bad quality air. The doctor has warned them that this problem may be transferred genetically to their children.

Scenario:

Jordan is really concerned about his wife and his children and he wants his wife to be able to go around the neighbourhood for brisk walks without fearing the bad quality air. He is determined to go around the neighbourhood and spread awareness about the same by giving solutions to the cause. He wants a tool that will keep every individual notified of their daily toxin emission through vehicles and other equipment. He also wishes to speak to construction workers to limit their usage of trucks and other heavy machinery in the neighbourhood, to reduce toxins and pollute the air. Moreover, he wishes to encourage all the young people in his neighbourhood to switch to more sustainable methods of travelling instead of taking their personal cars. He hopes that this will make a difference and his wife will once again learn to enjoy fresh nature.

Prototypes



A.3 Results/Evaluation

A.3.1 Manual Testing

A.3.1.1 Positive/Negative Testing

| Test Type: Positive | Test Number: P1 |
|---|-----------------|
| Objective: Testing the login functionality of the application. | |
| Set Up: An account was registered by the author with the email: 'test@gmail.com' and password: 'TestingApp123'. | |
| Expected Results: 1- The account is registered in the database. 2- Password entered is correct and matches with the database. 3- The application should redirect to the Home page of the application. | |
| Test: 1- The email 'test@gmail.com' was entered in the E-mail text field. 2- The password 'TestingApp123' was entered in the Password field. 3- The login button was clicked on the screen. | |
| Test Record: 1- The progress dialog was shown in which the authentication was done from the database. 2- The home page successfully loads up after the dialog was dismissed. | |
| Result: Passed | |

| Test Type : Negative | Test Number: N1 |
|--|-----------------|
| Objective: Testing the login functionality of the application. | |
| Set Up: | |
| 1- A random email is entered which does not match with the database. 2- Password entered is incorrect and does not match with the database. | |
| Expected Results: | |
| 1- The application should show an error message of the same. | |
| Test: | |
| 1- The email 'xyz@gmail.com' was entered in the E-mail text field. 2- The password 'Qwerty123' was entered in the Password field. 3- The login button was clicked on the screen. | |
| Test Record: | |
| 1- The progress dialog was shown in which the authentication was checked from the database. 2- An error message pop up 'Invalid Username and Password' . | |
| Result: Passed | |

| Test Type : Negative | Test Number: N2 |
|--|-----------------|
| Objective: Testing the login functionality of the application. | |
| Set Up: | |
| 1- No email is entered in the email dialog. 2- Password is entered. | |
| Expected Results: | |
| 1- The application should show a pop up requesting for email. | |
| Test: | |
| 1- The password 'TestingApp123' was entered in the Password field. 2- The login button was clicked on the screen. | |
| Test Record: | |
| 1- No progress dialog was displayed. 2- Instantly the app requested to enter email near the email field. | |
| Result: Passed | |

| Test Type : Negative | Test Number: N3 |
|---|------------------------|
| Objective: Testing the register functionality of the application . | |
| Set Up: The user clicks at the register button on the login page. | |
| Expected Results: | |
| 1- The application should show a pop up requesting for the missing field. | |
| Test: | |
| 1- All the fields except date of birth field was not filled. | |
| Test Record: | |
| 1- No progress dialog was displayed. | |
| 2- Instantly the app requested to enter Date of Birth near the Date of birth field. | |
| Result: Passed | |

| Test Type : Negative | Test Number: N4 |
|---|------------------------|
| Objective: Testing the register functionality of the application . | |
| Set Up: The user clicks at the register button on the login page. | |
| Expected Results: | |
| 1- The application should show a pop up requesting for a valid email. | |
| Test: | |
| 1- In the email text field 'Hello' was typed. | |
| 2- All the other text fields were filled correctly. | |
| Test Record: | |
| 1- No progress dialog was displayed. | |
| 2- Instantly the app requested to enter a valid email address near the email field. | |
| Result: Passed | |

| Test Type : Negative | Test Number: N5 |
|---|------------------------|
| Objective: Testing the register functionality of the application. | |
| Set Up: The user clicks at the register button on the login page. | |
| Expected Results: | |
| 1- The application should show a pop up to check for password. | |
| Test: | |
| 1- The password 'TestingApp123' was entered in the Password field. 2- The confirm password 'TestingApp13' was entered in the Password field. 3- The login button was clicked on the screen. | |
| Test Record: | |
| 1- No progress dialog was displayed. 2- Instantly the app gave an error message saying the passwords do not match. | |
| Result: Passed | |

| Test Type : Negative | Test Number: N6 |
|---|------------------------|
| Objective: Testing the register functionality of the application. | |
| Set Up: The user clicks at the register button on the login page. | |
| Expected Results: | |
| 1- The application should show an alert dialog informing that the email address already exists. | |
| Test: | |
| 1- All the fields were filled in correctly. 2- Email was entered as 'test@gmail.com' for which the account already exists. 3- The login button was clicked on the screen. | |
| Test Record: | |
| 1- The progress dialog was displayed. 2- An alert box was popped up saying 'Email already exists'. | |
| Result: Passed | |

| Test Type: Positive | Test Number: P2 |
|--|------------------------|
| Objective: Testing the register functionality of the application. | |
| Set Up: The user clicks at the register button on the login page. | |
| Expected Results: | |
| 1- The register page should be redirected to the login page 2- A message saying “Registration Successful” should be displayed. | |
| Test: | |
| 1- The user fills in all the text fields correctly. 2- The password and confirm password matches. | |
| Test Record: | |
| 1- The progress dialog was shown in which the registration was done in the database. 2- The login page successfully loads up after the dialog was dismissed. 3- The message “Registration Successful” was displayed at the bottom of the screen on login page. | |
| Result: Passed | |

| Test Type: Positive | Test Number: P3 |
|---|------------------------|
| Objective: Testing the Map functionality of the application. | |
| Set Up: | |
| 1-The user logs into the application. 2- The user clicks on the destination card. | |
| Expected Results: | |
| 1- A new page should be loaded, in which the user can type in the destination location. 2- Drop down menu of the recommend places should be shown while typing. | |
| Test: | |
| 1-The user clicks on the destination card with the map as soon as they login. 2- The user starts entering Ams in the destination card. | |
| Test Record: | |
| 1- A new page is opened as soon as the clicks on the destination card. 2-As soon as the user types Ams in the destination card, a list of suggested places is shown in a drop-down menu. | |
| Result: Passed | |

| Test Type: Positive | Test Number: P4 |
|---|-----------------|
| Objective: Testing the Map functionality of the application. | |
| Set Up: | |
| 1-The user logs into the application. 2- The user clicks on the destination card. 3- The user clicks on the option to choose destination on map. | |
| Expected Results: | |
| 1- Map should be made visible near the user's current location. 2- A pin pointer should be made available which is movable around the map. 3- As soon as the user drops the pin pointer a button should be made visible. 4-On clicking the button, the app should go back to the home page and the route and transport buttons should be made visible. | |
| Test: | |
| 1-The user clicks on the destination card with the map as soon as they login. 2- The user clicks on the option to set location on map. 3- The user drops the pin pointer at a location. | |
| Test Record: | |
| 1- A new page is opened as soon as the clicks on the destination card. 2-As soon as the user clicks on the option Choose Destination of Map, a map is made available. 3- The address of location is retrieved as the user chooses a drop point of the pin pointer. 4- As the user presses the done button, the page is redirected with a route drawn on the map and mode of transport is made available. | |

Result: Passed

| | |
|-----------------------------|------------------------|
| Test Type : Negative | Test Number: N7 |
|-----------------------------|------------------------|

Objective: Testing the Map functionality of the application.

Set Up:

- 1-The user logs into the application.
- 2- The user clicks on the destination card.

Expected Results:

- 1- The application should not crash or give any drop-down menu for random location typed.

Test:

- 1-The user clicks on the destination card with the map as soon as they login.
- 2- The user starts entering 'Loreum ndfdfkdknfdkfdf' in the destination card.

Test Record:

- 1- Nothing was displayed in the drop-down menu and user could not select anything.

Result: Passed

| | |
|----------------------------|------------------------|
| Test Type: Positive | Test Number: P5 |
|----------------------------|------------------------|

Objective: Testing the Map functionality of the application.

Set Up:

- 1-The user logs into the application.
- 2- The user clicks on the destination card.
- 3-The user clicks on the current location text view.

Expected Results:

- 1- A new page should be loaded, in which the user can type in the start location.
- 2- Drop down menu of the recommend places should be shown while typing.

Test:

- 1-The user clicks on the destination card with the map as soon as they login.
- 2- The user clicks on the current location card.
- 2- The user starts entering London in the current location card. (Pick Location)

Test Record:

- 1- A new page is opened as soon as the clicks on the current location card.
- 2-As soon as the user types London in the destination card, a list of suggested places is shown in a drop-down menu.

Result: Passed

| Test Type: Positive | Test Number: P6 |
|--|-----------------|
| Objective: Testing the Map functionality of the application. | |
| Set Up: | |
| 1-The user logs into the application. 2- The user clicks on the destination card. 3-The user clicks on the current location card. 3- The user clicks on the option to set location on map. | |
| Expected Results: | |
| 1- Map should be made visible near the user's current location. 2- A pin pointer should be made available which is movable around the app. 3- As soon as the user drops the pin pointer a button should be made visible. 4-On clicking the button, the app should go back to the destination page and the current location is the user's chosen location through pin pointers. | |
| Test: | |
| 1-The user clicks on the destination card with the map as soon as they login. 2-The user clicks on the current location card. 3- The user clicks on the option to set location on Map. 4- The user drops the pin pointer at a location. | |
| Test Record: | |
| 1- A new page is opened as soon as the clicks on the current location card. 2-As soon as the user clicks on the option to set location of map, a map is made available. 3- The address of location is retrieved as the user chooses a drop point. 4- As the user presses the done button, the page is redirected to set destination page with the current address as the chosen pin pointer location. | |
| Result: Passed | |

| Test Type: Positive | Test Number: P7 |
|---|-----------------|
| Objective: Testing the Map functionality of the application. | |
| Set Up: <ul style="list-style-type: none"> 1-The user logs into the application. 2- The user sets up the location and the route are drawn. | |
| Expected Results: <ul style="list-style-type: none"> 1- Map should be made visible with the route drawn. 2- The vehicle option should be made visible. 3- The start button should be made visible. | |
| Test: <ul style="list-style-type: none"> 1-The user logs in the application. 2- The user sets the location from current to Bush House. | |
| Test Record: <ul style="list-style-type: none"> 1- The page redirects and the route of the destination is made visible. 2-The different of mode of transports button is made available from current location to Bush House. 3- The proceed button is visible. | |
| Result: Passed | |

| Test Type: Positive | Test Number: P8 |
|---|-----------------|
| Objective: Testing the Map functionality of the application. | |
| Set Up: | |
| 1-The user logs into the application 2- The user sets up the location and the route are drawn. 3- The user sets the vehicle as Bus. 3-The user clicks on the start button at the bottom of the screen. | |
| Expected Results: | |
| 1- The moving vehicle animation should be made active. 2- The start button should be changed to Skip (to skip the animation). 3- When the user clicks on the skip or when the animation ends, the trip dialog should be shown | |
| Test: | |
| 1-The user logs in the application. 2- The user sets the location from current to Bush House. 3- The user clicks on the bus. 4- The user clicks on the start button. | |
| Test Record: | |
| 1- The animation of bus is shown. 2- As the animation is completed the trip dialog is shown or when the user clicks the skip button on the bottom screen. | |
| Results: Passed | |

| Test Type: Positive | Test Number: P9 |
|--|-----------------|
| Objective: Testing the Map functionality of the application. | |
| Set Up: | |
| 1-The user logs into the application. 2- The user sets up the location and the route are drawn. 3- The user sets the vehicle as WALK. 3-The user clicks on the start button at the bottom of the screen. | |
| Expected Results: | |
| 1- The moving vehicle animation should be made active. 2- The start button should be changed to Skip (to skip the animation). 3- When the user clicks on the skip or when the animation ends, the trip dialog should be shown. 4- The trip dialog should have calories instead of Co2 emitted. 5- The disclaimer message should be disabled. | |
| Test: | |
| 1-The user logs in the application. 2- The user sets the location from current to Bush House. 3- The user clicks on the walk/cycle. 4- The user clicks on the start button. | |
| Test Record: | |
| 1- The animation of walk/cycle is shown. 2- As the animation is completed the trip dialog is shown or when the user clicks the skip button on the bottom screen. 3- In the trip dialog, it shows the calories burnt instead of co2 emitted. | |
| Results: Passed | |

| Test Type: Positive | Test Number: P10 |
|---|------------------|
| Objective: Testing the Map functionality of the application. | |
| Set Up: <ul style="list-style-type: none"> 1-The user logs into the application 2- The user sets up the location and the route are drawn. 3- The user sets the vehicle as CAR. 3-The user clicks on the start button at the bottom of the screen. | |
| Expected Results: <ul style="list-style-type: none"> 1- The moving vehicle animation of car should be made active. 2- The start button should be changed to Skip (to skip the animation). 3- When the user clicks on the skip or when the animation ends, the trip dialog should be shown. 4- The trip dialog should display 2 scroll bars through which the car type and fuel type can be selected. 5- The trip dialog should change the display of the co2 emissions accordingly with the car type and fuel chosen and an additional disclaimer for electric/PHEV cars should be displayed. | |
| Test: <ul style="list-style-type: none"> 1-The user logs in the application. 2- The user sets the location from current to Bush House 3- The user clicks on the walk/cycle. 4- The user clicks on the start button. | |
| Test Record: <ul style="list-style-type: none"> 1- The animation of car is shown. 2- As the animation is completed the trip dialog is shown or when the user clicks the skip button on the bottom screen. 3- In the trip dialog, two scroll bars are made available through which the user chooses compact car and electric. 4- The value of co2 emitted changes. 5- Two disclaimer messages are shown. | |
| Results: Passed | |

| Test Type: Positive | Test Number: P11 |
|---|------------------|
| Objective: Testing the Map functionality of the application. | |
| Set Up: <ul style="list-style-type: none"> 1-The user logs into the application 2- The user sets up the location and the route are drawn. (distance greater than 50kms) | |
| Expected Results: <ul style="list-style-type: none"> 1- Map should be made visible with the route drawn. 2- The vehicle option should be made visible. 3- The start button should be made visible. 4- Available vehicle options should not include walking and cycling | |
| Test: <ul style="list-style-type: none"> 1-The user logs in the application. 2- The user sets the location from current to Cambridge. | |
| Test Record: <ul style="list-style-type: none"> 1- The page redirects and the route of the destination is made visible. 2-The different mode of transports button is made available from current location to Cambridge and the walking and cycling options are not visible. 3- The proceed button is visible. | |
| Results: Passed | |

| | |
|---|-------------------------|
| Test Type: Positive | Test Number: P12 |
| Objective: Testing the Map functionality of the application. | |

Set Up:

- 1-The user logs into the application
- 2- The user sets up the location and the route are drawn. (distance greater than 500kms)

Expected Results:

- 1- Map should be made visible with the route drawn.
- 2- The vehicle option should be made visible.
- 3- The start button should be made visible.
- 4- Available vehicle options should not include walking and cycling.
- 5- Flight option should be made available with other modes of transport.

Test:

- 1-The user logs in the application.
- 2- The user sets the location from current to Amsterdam.

Test Record:

- 1- The page redirects and the route of the destination is made visible.
- 2-The different of mode of transports button is made available from current location to Amsterdam and the walking and cycling options are not visible.
- 3- The flight option is available.
- 3- The proceed button is visible.

Results: Passed

| Test Type: Negative | Test Number: N8 |
|--|-----------------|
| Objective: Testing the Map functionality of the application. | |
| Set Up: | |
| 1-The user logs into the application. 2- The user sets up the location and is redirected to home page. | |
| Expected Results: | |
| 1- The application should not crash and give an alert dialog stating, route not found. | |
| Test: | |
| 1- The user logs in the application. 2- The user chooses current location to Australia as destination and source. | |
| Test Record: | |
| 1- The page is redirected to the home page after setting up the location. 2- The app waits for few minutes and then a dialog is made visible with okay button stating the route not found and on clicking okay, the map is restarted. | |
| Result: Passed | |

| Test Type: Positive | Test Number: P13 |
|--|------------------|
| Objective: Testing the statistics functionality of the application. | |
| Set Up: <p>1-The user logs into the application. 2- The user clicks on the navigation bar. 3- The user clicks on the statistics window.</p> | |
| Expected Results: <p>1- Total c02 emissions of the day should be displayed. 2-Total calories burnt of the day should be displayed. 3- Trip summaries of the day should be displayed.</p> | |
| Test: <p>1-The logs into the application. 2- The user tracks his daily activities which include: <ul style="list-style-type: none"> • Walk: London Eye to Shard • Cab: Shard to Hounslow • Train: Hounslow to Chelsea 3- The user clicks on the nav button and selects the statistics page.</p> | |
| Test Record: <p>1-The statistics page is loaded. 2- The total co2 emitted is displayed. 3- The total calories burnt is displayed. 4- The trip summaries of each trip are listed below with the co2 and calories burnt respectively.</p> | |
| Result: Passed | |

| Test Type: Positive | Test Number: P14 |
|--|------------------|
| Objective: Testing the statistics functionality of the application. | |
| Set Up: | |
| 1-The user logs into the application 2- The user clicks on the navigation bar. 3- The user clicks on the statistics window. 4- The user clicks on the filter option and choose the previous date. | |
| Expected Results: | |
| 1- Total c02 emissions of the chosen day should be displayed. 2-Total calories burnt of the chosen day should be displayed. 3- All the trip summaries of the chosen day should be displayed | |
| Test: | |
| 1-The logs into the application. 2- The user had tracked his previous day activities which include: <ul style="list-style-type: none"> • Walk: London Eye to Shard • Cab: Shard to Hounslow • Train: Hounslow to Chelsea 3- The user clicks on the nav button and selects the statistics page. | |
| Test Record: | |
| 1-The statistics page is loaded. 2- The total co2 emitted of the chosen day is displayed. 3- The total calories burnt of the chosen day is displayed. 4- The trip summaries of each trip of the chosen day are listed below with the co2 and calories burnt of each trip. | |
| Result: Passed | |

| Test Type: Negative | Test Number: N9 |
|---|-----------------|
| Objective Testing the statistics functionality of the application. | |
| Set Up: | |
| 1-The user logs into the application 2- The user clicks on the navigation bar. 3- The user clicks on the statistics window. 4- The user clicks on the filter option and choose the date on which he had no trips. | |
| Expected Results: | |
| 1- No data should be displayed on the screen. 1-The logs into the application. 2- The user clicks on the nav button and selects the statistics page. 3- The user clicks on the filter button and selects the date with no trip data (2 months ago) | |
| Test Record: | |
| 1- The page displays no trip data message on the screen | |
| Result: Passed | |

Test Type: Positive

Test Number: P15

Objective: Testing the Update Profile functionality of the application.

Set Up:

- 1-The logs into the application.
- 2- The user clicks on the nav button and selects the update profile page.
- 3- The user changes his phone number.

Expected Results:

- 1- The register page should show a progress dialog while it makes changes in the database.
- 2- A profile update message should be displayed as the progress dialog disappears.

Test:

- 1: The user logs into the application.
- 2- The user selects the Profile option from the navigation drawer.
- 3- The user finds all his details in editable format expect for email and a submit button
- 4- The user changes his phone number
- 5- The user presses the submit button

Test Record:

- 1- The progress dialog was shown in which the details were being updated on the database.
- 2- The message “Profile Updated” was displayed at the bottom of the screen on login page.

Result: Passed

| | |
|---|-------------------------|
| Test Type: Positive | Test Number: P16 |
| Objective: Testing the change password functionality of the application. | |

Set Up:

- 1-The user logs into the application.
- 2- The user clicks on the nav button and selects the change password option.

Expected Results:

- 1- The change password page should show a progress dialog while it changes the password in the database.
- 2- A change password message should be displayed as the progress dialog disappears.
- 3- The page should be redirected to the Home page after the password changes.

Test:

- 1: The user logs into the application.
- 2- The user selects the change password option from the navigation drawer.
- 3- The user enters his current password.
- 4- The user enters his new password and confirms the new password.
- 5- The user presses the submit button

Test Record:

- 1- The progress dialog was shown in which the password was being updated on the database.
- 2- The message “Change password” was displayed at the bottom of the screen.
- 3- Page redirected to the home screen of the application.

Result: Passed

Test Type: Negative

Test Number: N10

Objective: Testing the change password functionality of the application.

Set Up:

- 1-The user logs into the application.
- 2- The user clicks on the nav button and selects the change password option.
- 3- Incorrect current password is entered by the user.
- 4- Correct new password and confirm new password is entered by the user.

Expected Results:

- 1- Progress dialog should be made visible when the user presses on the submit button.
- 2- A dialog which states that the current password entered is incorrect.

- 1-The user logs into the application.
- 2- The user clicks on the nav button and selects the change password page.
- 3- The user enters an incorrect current password.
- 4- The user enters the new password and the confirm new password and they match.
- 5- The user clicks on submit button

Test Record:

- 1- The progress dialog is made visible.
- 2- An alert dialog pop up is displayed stating the current password is incorrect

Result: Passed

Test Type: Negative

Test Number: N11

Objective: Testing the change password functionality of the application.

Set Up:

- 1-The user logs into the application.
- 2- The user clicks on the nav button and selects the change password option.
- 3- Correct current password is entered by the user
- 4- Incorrect new password and confirm new password is entered by the user (they do not match).

Expected Results:

- 1- Progress dialog should be made visible when the user presses on the submit button.
- 2- A dialog which states that the current password entered is incorrect.

- 1-The user logs into the application.
- 2- The user clicks on the nav button and selects the change password page.
- 3- The user enters an incorrect current password.
- 4- The user enters the new password and the confirm new password and they match.
- 5- The user clicks on submit button

Test Record:

- 1- No progress dialog was displayed.
- 2- Instantly the app gave an error message saying the passwords do not match.

Result: Passed

| | |
|--|-------------------------|
| Test Type: Positive | Test Number: P17 |
| Objective: Testing the logout functionality of the application. | |

Set Up:

- 1-The user logs into the application.
- 2- The user clicks on the nav button and selects the logout option.

Expected Results:

- 1- A Confirm Logout pop up dialog with the option to cancel or okay.
- 2- Cancel should redirect to home screen.
- 3- okay should redirect the application to log in screen.

Test:

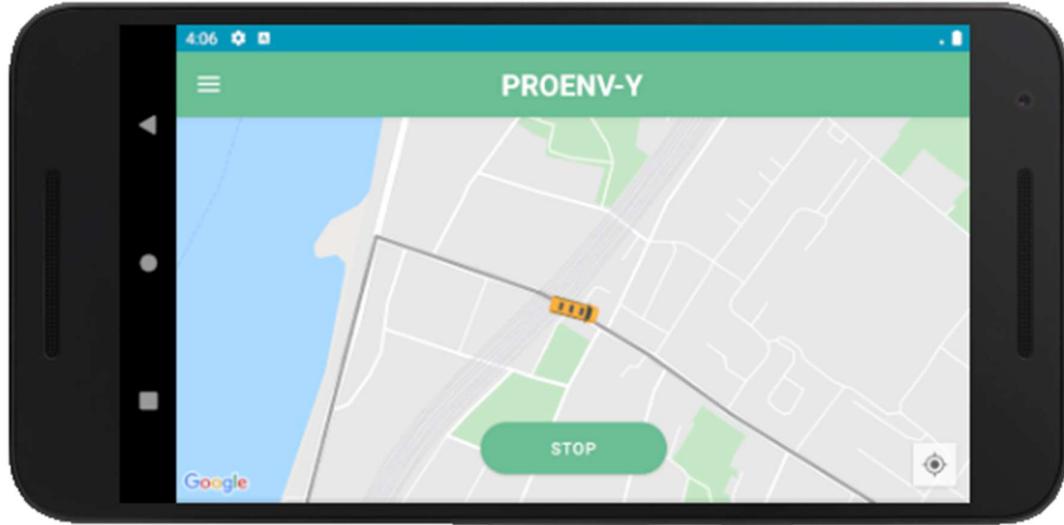
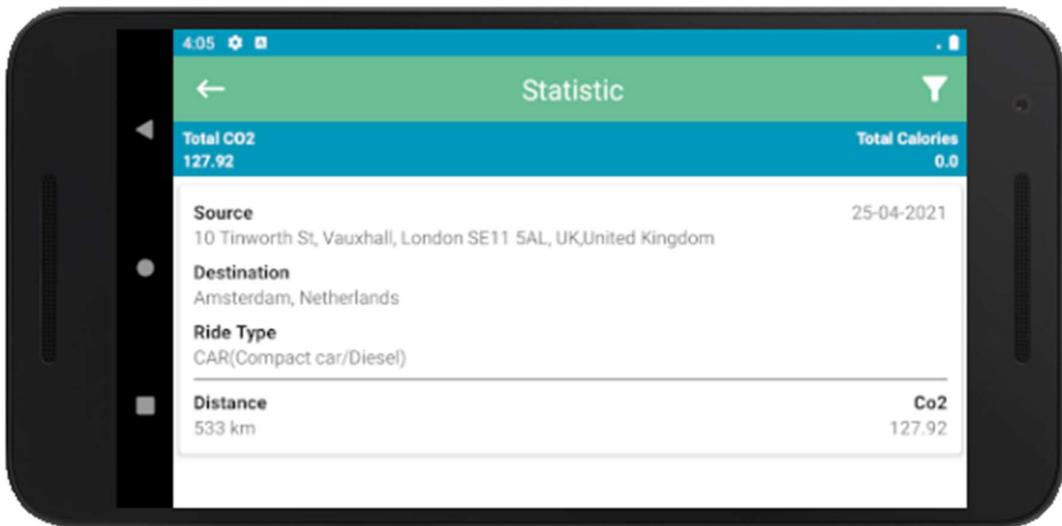
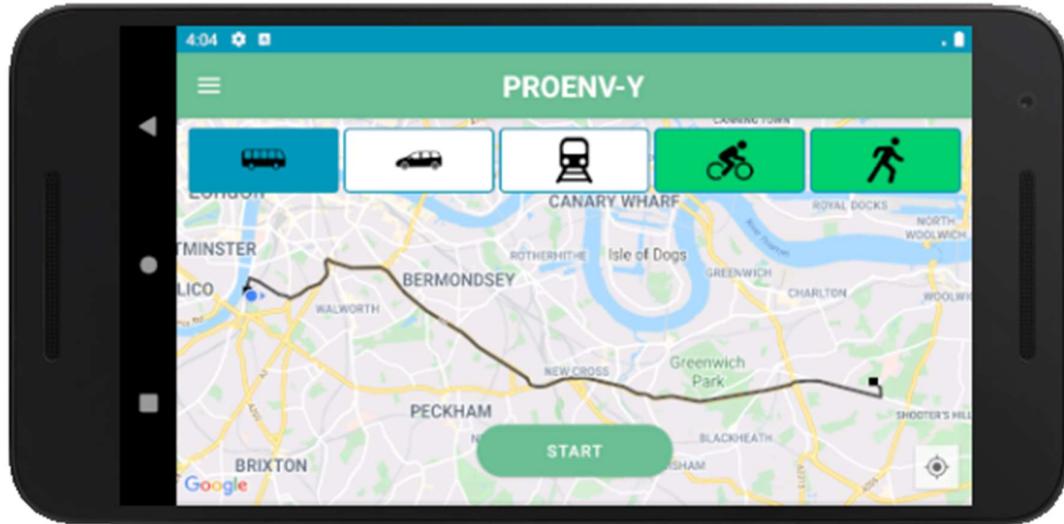
- 1: The user logs into the application.
- 2- The user selects the change Logout option from the navigation drawer.
- 3- An confirm logout password dialog is displayed.
- 4- First the user presses on cancel
- 5- Later the user presses on okay after repeating step 2 again.

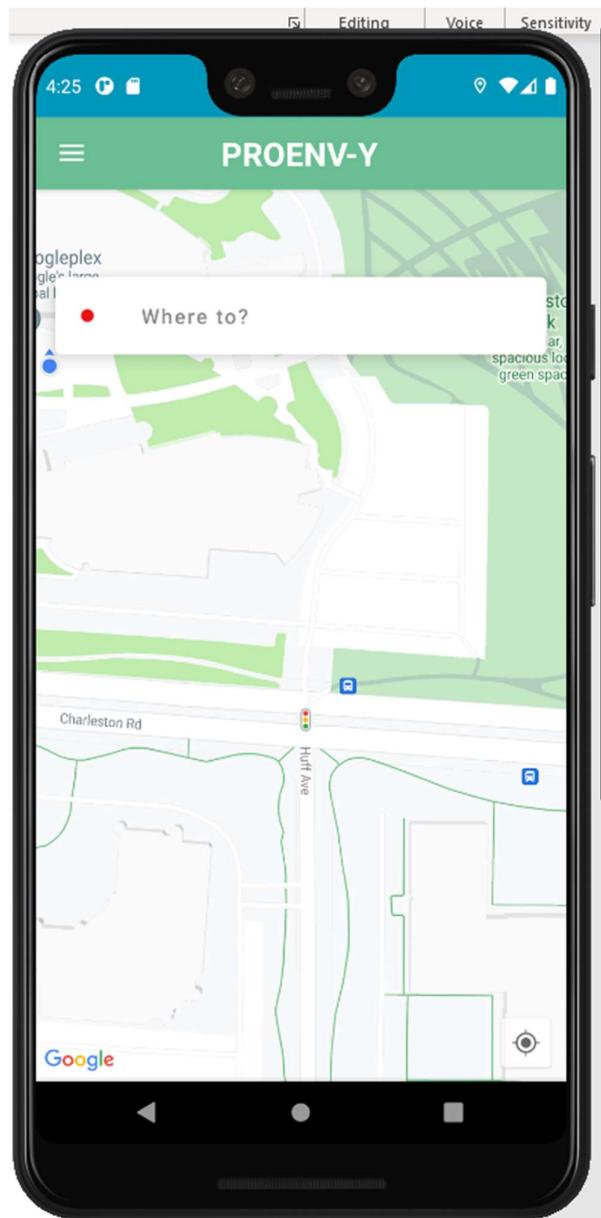
Test Record:

- 1- The alert logout dialog was shown with the option to cancel or okay.
- 2- The cancel redirected to the home screen of the application.
- 3- The Okay, redirected the app to the log in screen of the application and now the user was logged out of the application.

Result: Passed

Test on Emulators





Usability Testing

Surveys

PROENV-Y

An Environmental Advisor Application



i. Would you like to use this application frequently ? *

- Strongly agree
- Agree
- Maybe
- Disagree
- Strongly disagree

2. Did you find the application to be unnecessarily complex ? *

Strongly agree

Agree

Maybe

Disagree

Strongly disagree

3. Do you think the system is easy to use ? *

Strongly agree

Agree

Maybe

Disagree

Strongly disagree

4. Would you need the support of a technical person to use the application ? *

Strongly agree

Agree

Maybe

Disagree

Strongly disagree

5. Do you think the various functions integrated in the application are well integrated ? *

Strongly agree

Agree

Maybe

Disagree

Strongly disagree

6. Do you think there is too much inconsistency in the application ? *

Strongly agree

Agree

Maybe

Disagree

Strongly disagree

7. Do you think that most people would learn to use this system quickly ? *

Strongly agree

Agree

Maybe

Disagree

Strongly disagree

8. Do you find the application to be very inconvenient to use ? *

- Strongly agree
- Agree
- Maybe
- Disagree
- Strongly disagree

9. Do you feel confident while using the application ? *

- Strongly Agree
- Agree
- Maybe
- Disagree
- Strongly disagree

10. Do you think that you need to learn a lot of things before you get going with the system ? *

- Strongly agree
- Agree
- Maybe
- Disagree
- Strongly disagree

This content is neither created nor endorsed by Google.

Google Forms

PROENV-Y

An Environmental Advisor Application



i. Would you like to use this application frequently ? *

- Strongly agree
- Agree
- Maybe
- Disagree
- Strongly disagree

2. Did you find the application to be unnecessarily complex ? *

- Strongly agree
- Agree
- Maybe
- Disagree
- Strongly disagree

3. Do you think the system is easy to use ? *

- Strongly agree
- Agree
- Maybe
- Disagree
- Strongly disagree

4. Would you need the support of a technical person to use the application ? *

- Strongly agree
- Agree
- Maybe
- Disagree
- Strongly disagree

5. Do you think the various functions integrated in the application are well integrated ? *

Strongly agree

Agree

Maybe

Disagree

Strongly disagree

6. Do you think there is too much inconsistency in the application ? *

Strongly agree

Agree

Maybe

Disagree

Strongly disagree

7. Do you think that most people would learn to use this system quickly ? *

Strongly agree

Agree

Maybe

Disagree

Strongly disagree

8. Do you find the application to be very inconvenient to use ? *

- Strongly agree
- Agree
- Maybe
- Disagree
- Strongly disagree

9. Do you feel confident while using the application ? *

- Strongly Agree
- Agree
- Maybe
- Disagree
- Strongly disagree

10. Do you think that you need to learn a lot of things before you get going with the system ? *

- Strongly agree
- Agree
- Maybe
- Disagree
- Strongly disagree

This content is neither created nor endorsed by Google.

Google Forms

PROENV-Y

An Environmental Advisor Application



i. Would you like to use this application frequently ? *

- Strongly agree
- Agree
- Maybe
- Disagree
- Strongly disagree

2. Did you find the application to be unnecessarily complex ? *

- Strongly agree
- Agree
- Maybe
- Disagree
- Strongly disagree

3. Do you think the system is easy to use ? *

- Strongly agree
- Agree
- Maybe
- Disagree
- Strongly disagree

4. Would you need the support of a technical person to use the application ? *

- Strongly agree
- Agree
- Maybe
- Disagree
- Strongly disagree

5. Do you think the various functions integrated in the application are well integrated ? *

Strongly agree

Agree

Maybe

Disagree

Strongly disagree

6. Do you think there is too much inconsistency in the application ? *

Strongly agree

Agree

Maybe

Disagree

Strongly disagree

7. Do you think that most people would learn to use this system quickly ? *

Strongly agree

Agree

Maybe

Disagree

Strongly disagree

8. Do you find the application to be very inconvenient to use ? *

- Strongly agree
- Agree
- Maybe
- Disagree
- Strongly disagree

9. Do you feel confident while using the application ? *

- Strongly Agree
- Agree
- Maybe
- Disagree
- Strongly disagree

10. Do you think that you need to learn a lot of things before you get going with the system ? *

- Strongly agree
- Agree
- Maybe
- Disagree
- Strongly disagree

This content is neither created nor endorsed by Google.

Google Forms

PROENV-Y

An Environmental Advisor Application



i. Would you like to use this application frequently ? *

- Strongly agree
- Agree
- Maybe
- Disagree
- Strongly disagree

2. Did you find the application to be unnecessarily complex ? *

Strongly agree

Agree

Maybe

Disagree

Strongly disagree

3. Do you think the system is easy to use ? *

Strongly agree

Agree

Maybe

Disagree

Strongly disagree

4. Would you need the support of a technical person to use the application ? *

Strongly agree

Agree

Maybe

Disagree

Strongly disagree

5. Do you think the various functions integrated in the application are well integrated ? *

Strongly agree

Agree

Maybe

Disagree

Strongly disagree

6. Do you think there is too much inconsistency in the application ? *

Strongly agree

Agree

Maybe

Disagree

Strongly disagree

7. Do you think that most people would learn to use this system quickly ? *

Strongly agree

Agree

Maybe

Disagree

Strongly disagree

8. Do you find the application to be very inconvenient to use ? *

- Strongly agree
- Agree
- Maybe
- Disagree
- Strongly disagree

9. Do you feel confident while using the application ? *

- Strongly Agree
- Agree
- Maybe
- Disagree
- Strongly disagree

10. Do you think that you need to learn a lot of things before you get going with the system ? *

- Strongly agree
- Agree
- Maybe
- Disagree
- Strongly disagree

This content is neither created nor endorsed by Google.

Google Forms

Appendix B

User Guide

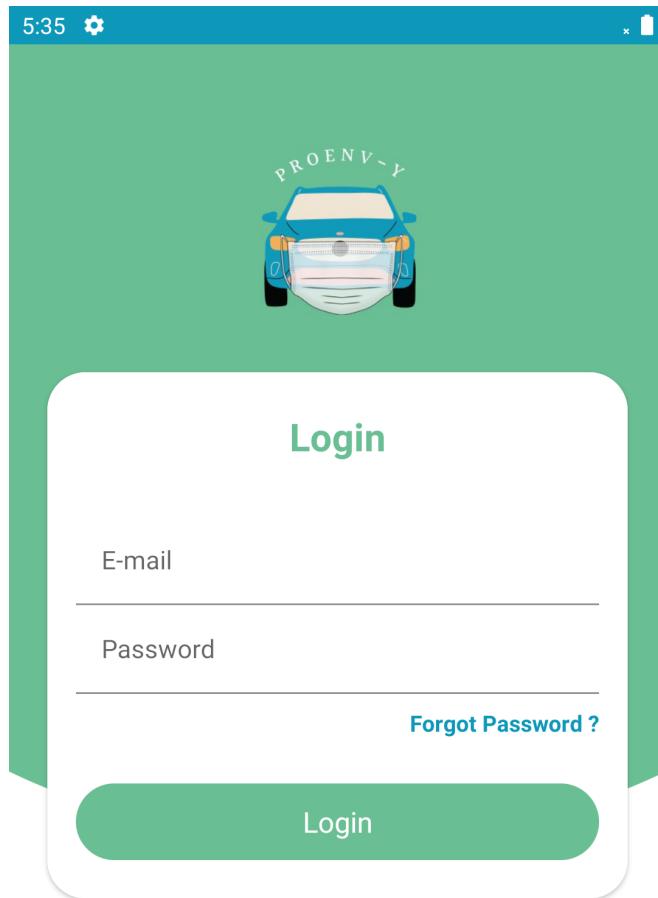
B.1 Introduction

A user guide is a manual that instructs the user on how to use the application on a step to step basis. It provides a breakdown of all the features of the application to serve those users who are not very technology friendly. It also organises every aspect of the app to facilitate full usage of the app and its benefits that is aimed to cater every individual and help them reduce the emission of CO₂, thus serving and making a difference in society.

B.2 About the Application

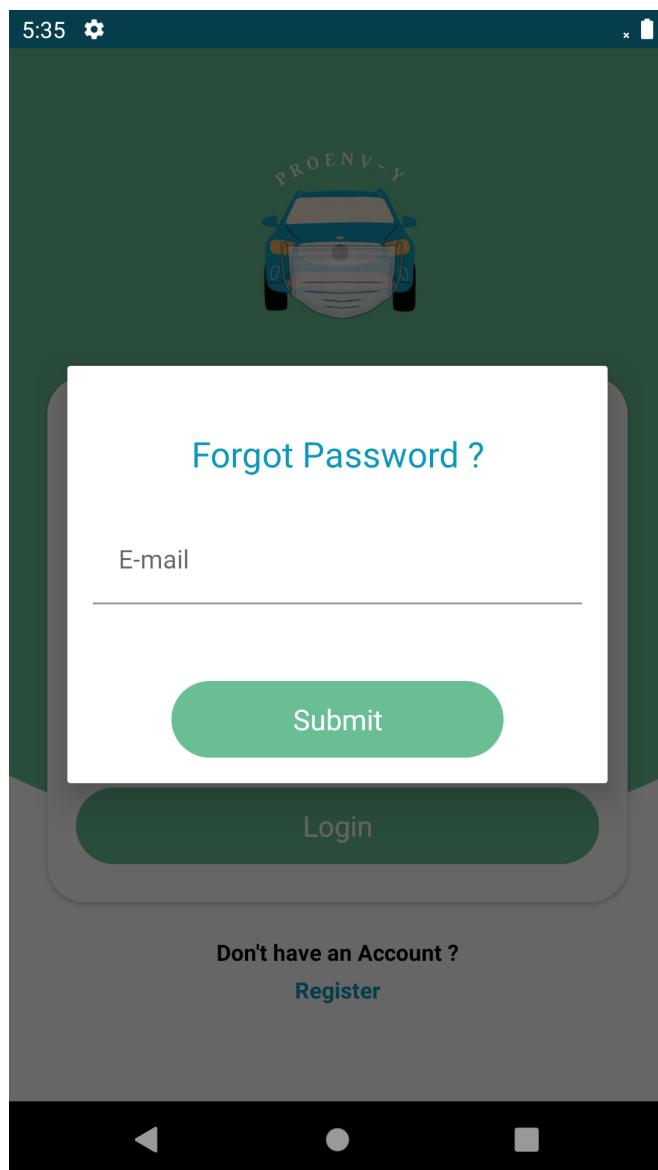
The principle intention of the project is to help the users estimate their carbon emission (a.k.a carbon footprint) while travelling from one location to the other. The app would not just estimate and calculate the emission but also persuade the user to choose a more conservative mode of transport, therefore help in diminishing a critical danger to mankind i.e. Global warming. The application is intended to spread mindfulness and awareness particularly among younger individuals. Monitoring the user's carbon impression just through their daily commute, will assist the users to reflect upon their decision and spur them to take up a more sustainable mode of transport. In addition to this, the app will also implement calorie calculation for sustainable methods like cycling or walking, in order to motivate the users to transition to a healthier lifestyle.

B.3 Login screen



Login Screen

The login page of the app is carefully curated with the app logo on the front top and a soothing and visually appealing colour scheme of green and white. This page has a text field for the user's email and password with the option of "forgot password" for those who don't remember their password after registration.

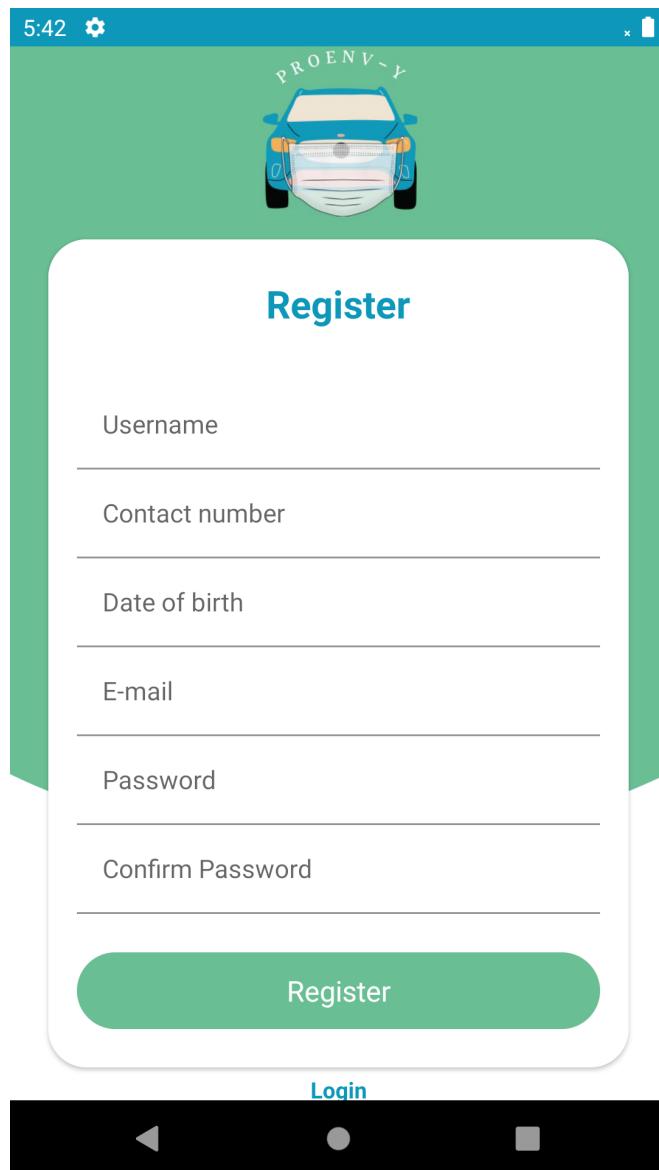


Forget Password dialog

A simple entry of their email used during registration would help them recover their password, hassle free. In case, the user is not registered in the app, simply clicking the Register option would lead them to a page where they would have to put in some personal information and then after registering, login again to gain benefits of the app. The user must enter their full email and the correct password. In case, the user forgets to enter one of the fields, the application will prompt a small message seeking users to enter the missing field. If the entered email by the user is not registered by the user or the password entered is incorrect, a dialog informing the user of the incorrect password or email will be prompted. The user can either check their password with the forgot password option by entering the email, but

if the email does not exist (i.e. the email used by the user is not registered) a dialog box will open up stating “This email is not found. Please enter your registered email”. If the entered password and email is registered with the application and are correct, the application will be redirected to the home page of the application.

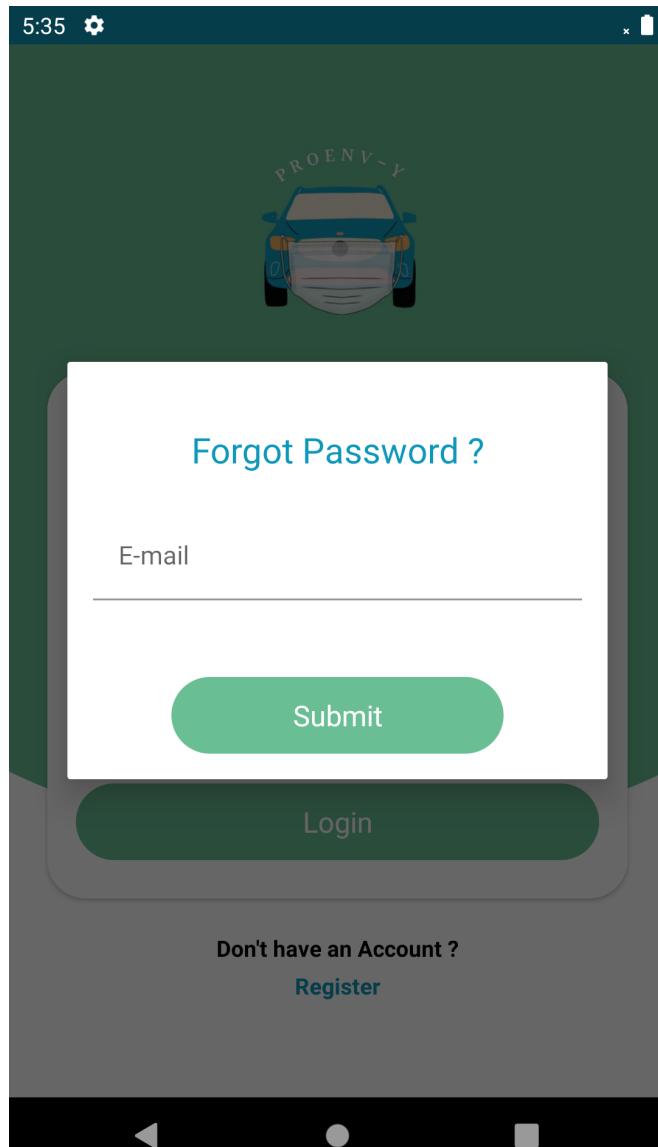
B.4 Register Screen



Register Screen

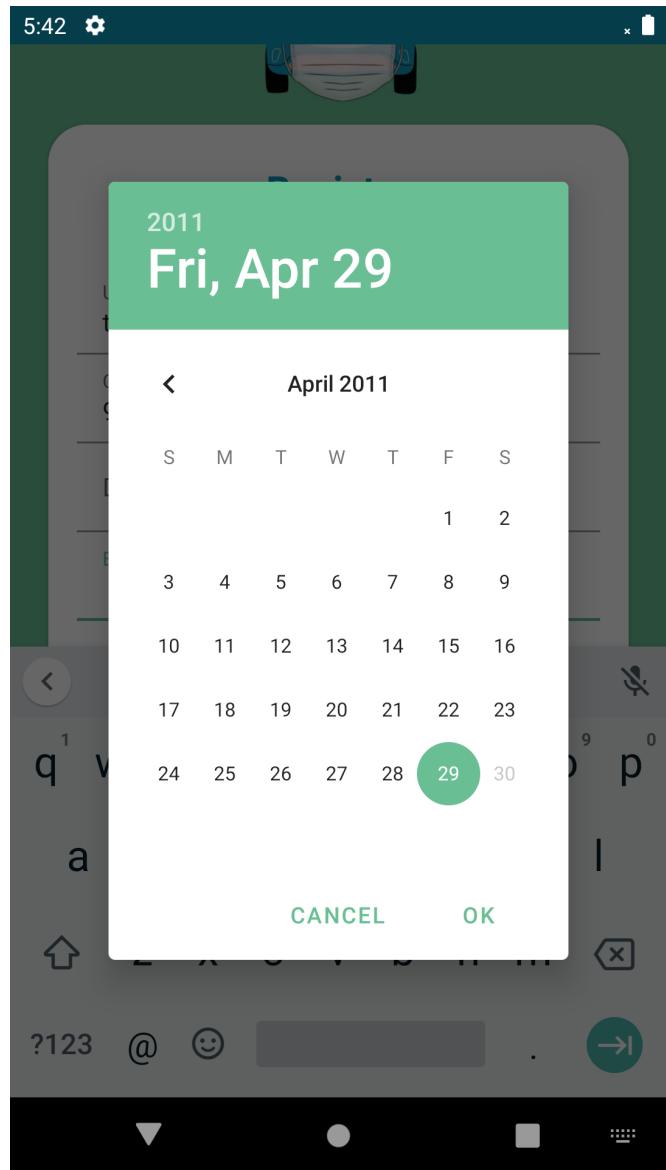
The login screen of the app has the option to “Register” in case the user is new to the application. In this case, the user simply has to click on the register option, which will lead them

to a page containing edit text boxes for personal information such as username, password, contact number, date of birth and email. The app would require the client to confirm their password to ensure the correct password is entered by the user. The user should also remember that the password needs to be of 5 characters at least.



Password Validation

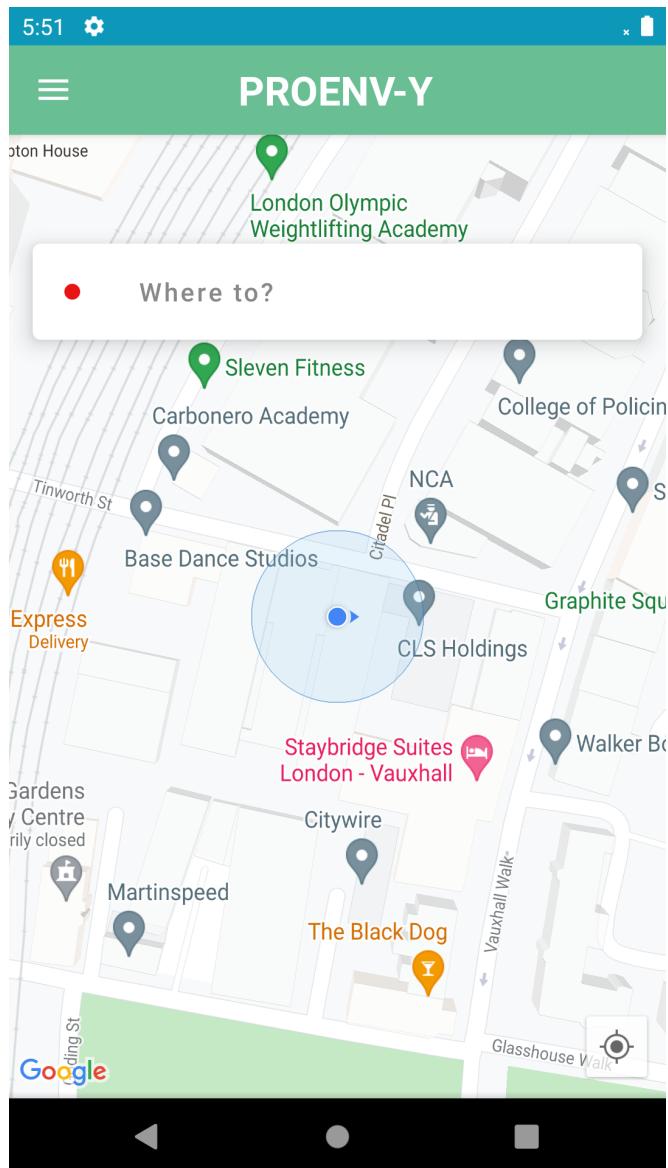
In case the entered password is less than that, the user will be prompted with the same error message near the Password field. The minimum age to use the application is 10 years old and this has been directly implemented in the calendar date picker.



Password Validation

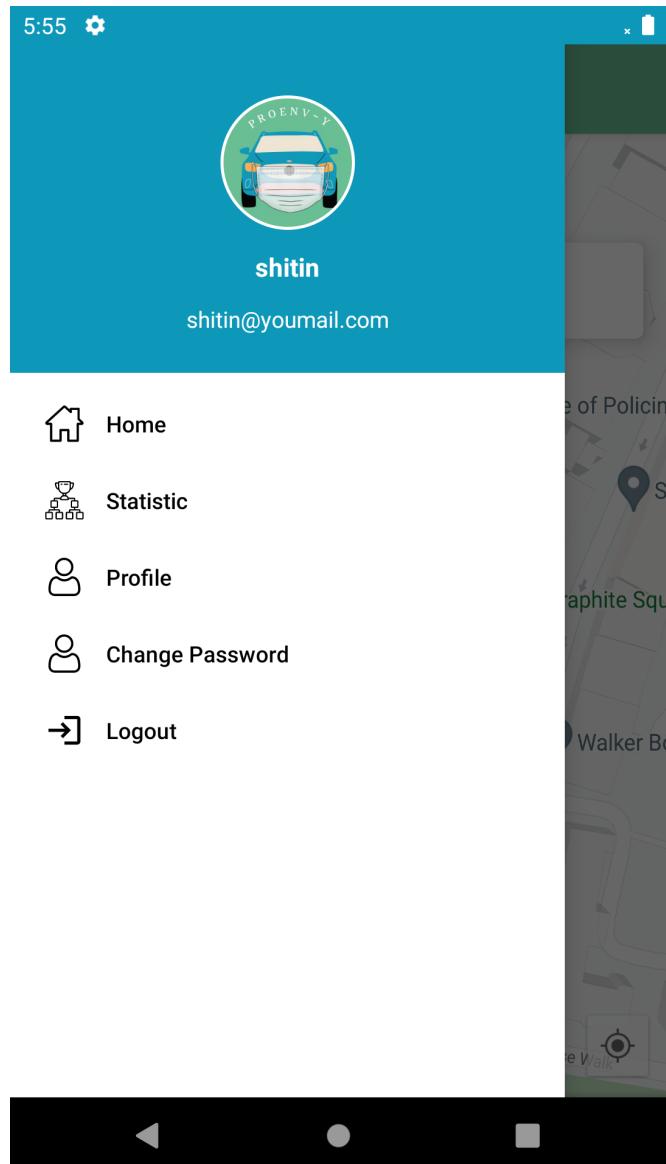
Once the client has clicked on the green register symbol after providing their correct details, a message stating “REGISTRATION SUCCESSFUL” will be displayed at the bottom of the screen and application will be automatically redirected to the login page. If the email entered by the user is already registered, a dialog box informing the user of the same will be displayed.

B.5 Home Screen



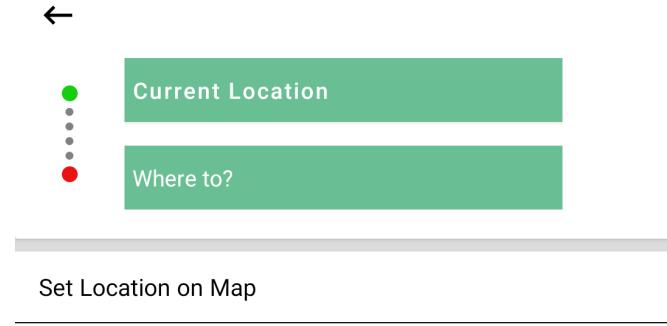
Home Screen

Post logging in, the app leads the user to a map of their location containing a blue dot to show the users exact location. It also shows tags of different landmarks around the user's location for ease in understanding of their surroundings. On the top left of the screen, a 3 line option is provided containing options such as home, statistic, profile, change password and logout and this serves as the navigation of the application.

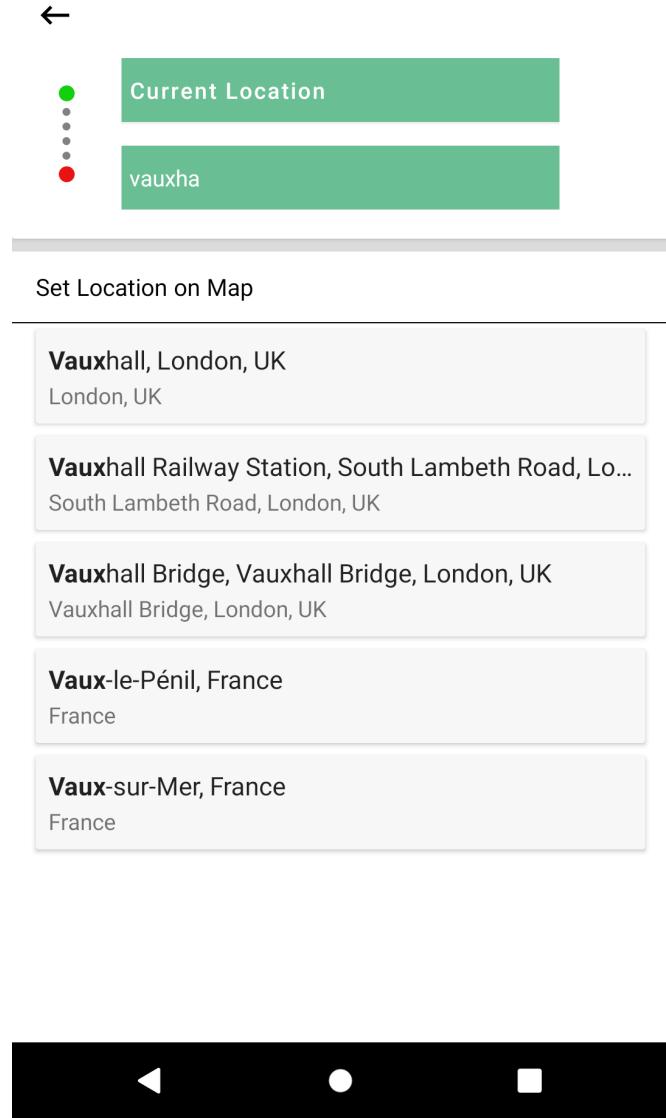


Navigation Toggle

Enter destination- On the top of the home screen, there is a white box saying “where to” and a red dot symbolising the destination is displayed. On clicking on that option, the user is redirected to a new page. The user may choose the destination by typing in the location and a set of recommended places will be displayed and the user may click on a location in the drop down list.

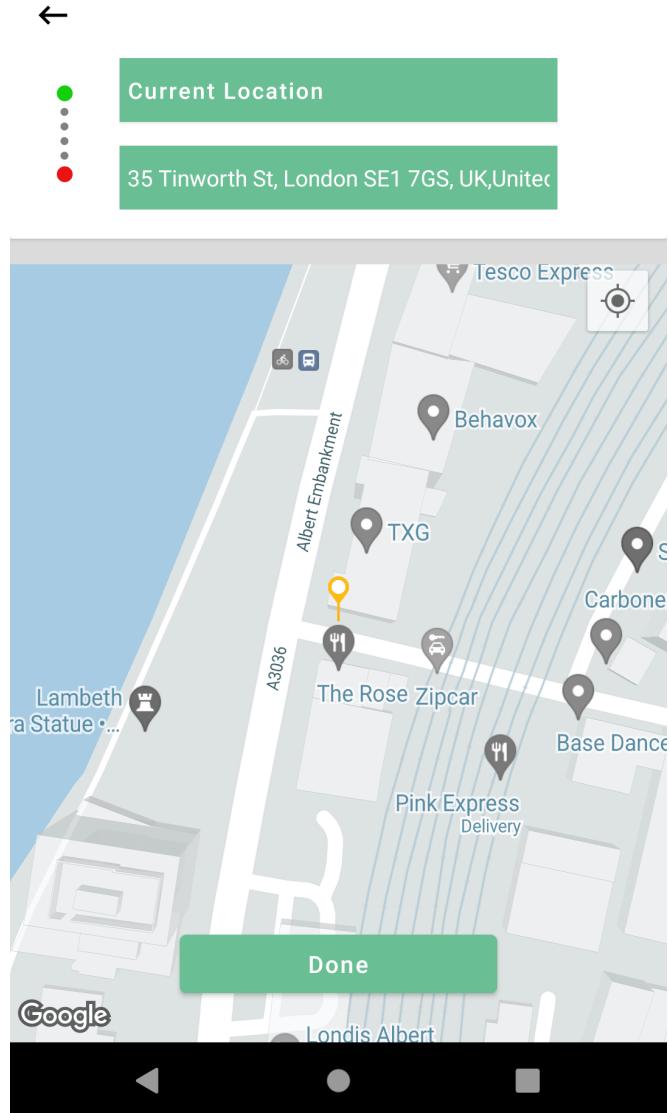


Destination selection page



Drop down recommended places

If the user wishes to set the location on map, the user needs to click on Set Location on Map option below the destination. The user is then led to a new page in which a yellow pin-pointer is made available to the user. The user may feel free to choose the location directly on the map, the address of the chosen location will be fetched and a done button will be made available. By choosing the location on map or through the drop down list, the app redirects to the Home of the application and further functionalities are made available.



Set Location on Map

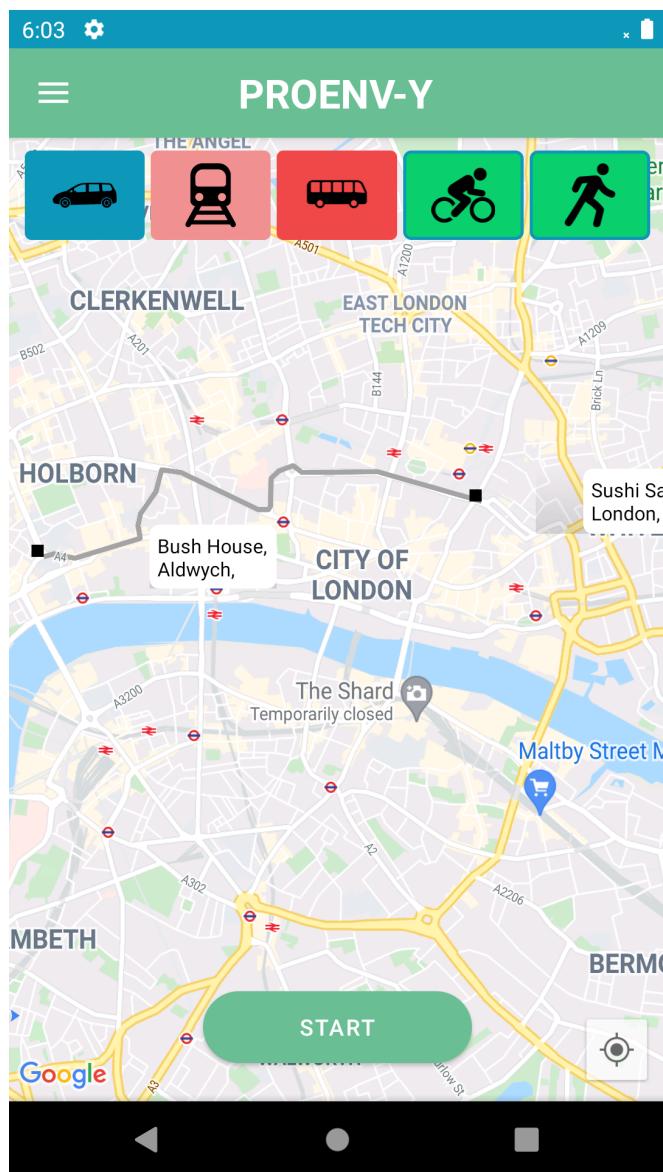
The user must allow the application to use their location. The app detects the user's current location as the default start location. In case the client wishes to choose another pick up location, the user needs to click on the current location text box on the screen and will be redirected to another page in which they can choose their start location. The procedure to choose the start location remains the same as the destination, but instead of redirecting the application to the home screen, the application will be redirected to the set destination page again.

PROENV- Y

Set Pick Up Location

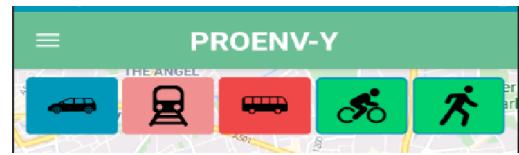
Set Location on Map

Pick Location (Start point)



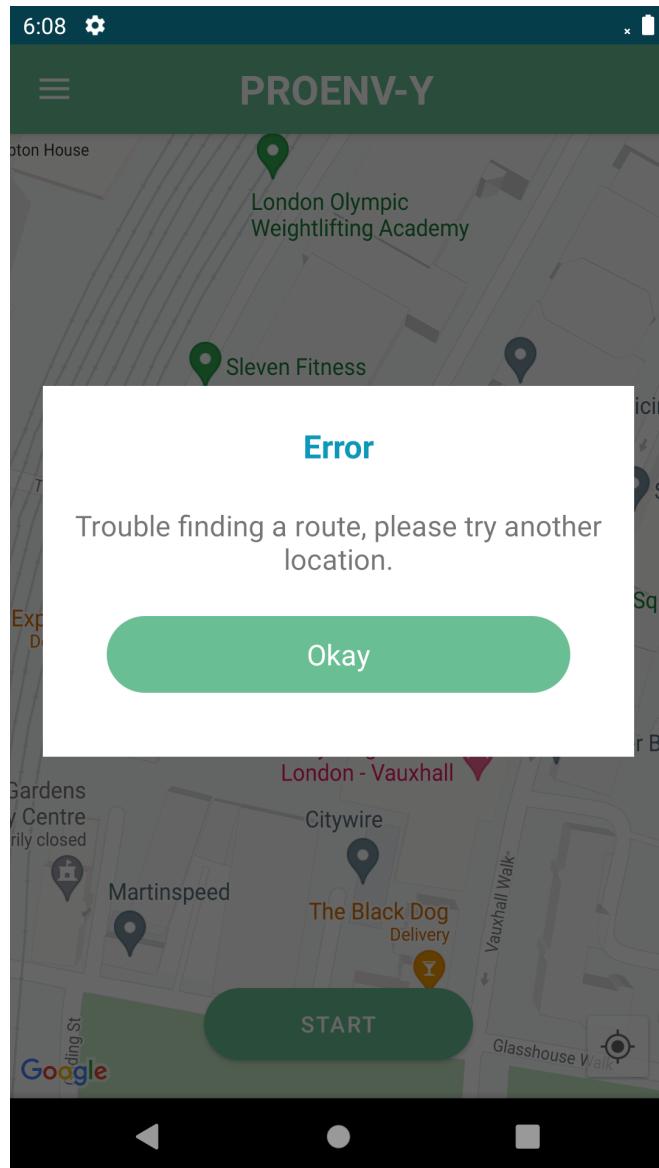
Home Screen with route

Once the user chooses the start and end location of the trip, the application redirects to the home screen where a route, in grey and black, between the Start and End points are drawn. Other options to choose the vehicle are made available on top of the screen and the user may choose the vehicle of their choice. The eco-friendly modes of transport are shown in green and the rest of the vehicles except cars are shown in increasing shade of red. The darker the shade of red is, the more the co2 emissions. Thus it is advisable for the user to opt for options with less or no co2 emissions.



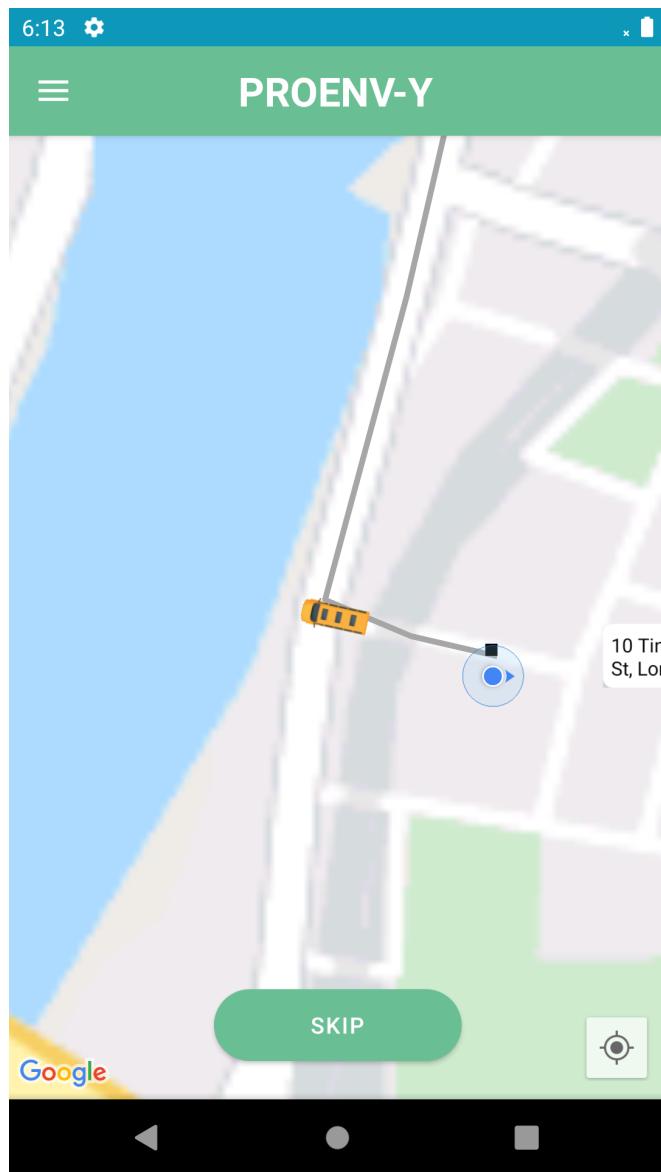
Vehicle Selection

At times the user may be prompted with the message saying, route not found. This happens when the application cannot find the route between the destinations. The google maps only allow the use of driving or walking routes, so in case a driving route cannot be found between the 2 locations, the message will be displayed.

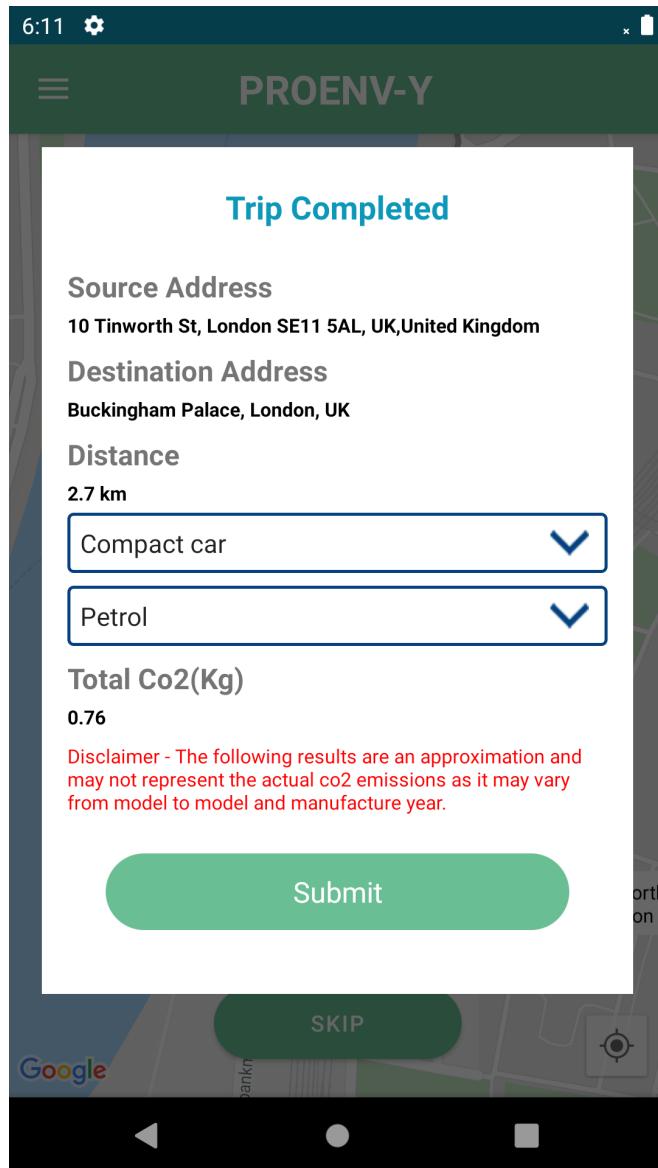


No route found dialog

As soon as the user chooses the vehicle, and clicks on the start button, the app shows a virtual vehicle in the former's shape travelling along the route indicated by the black line. If the client clicks on "skip", the animation ends. The user may note that the start button may take some time to be visible on the screen. This is because the data in background for map is being set and for large distances this may take a little longer than usual.

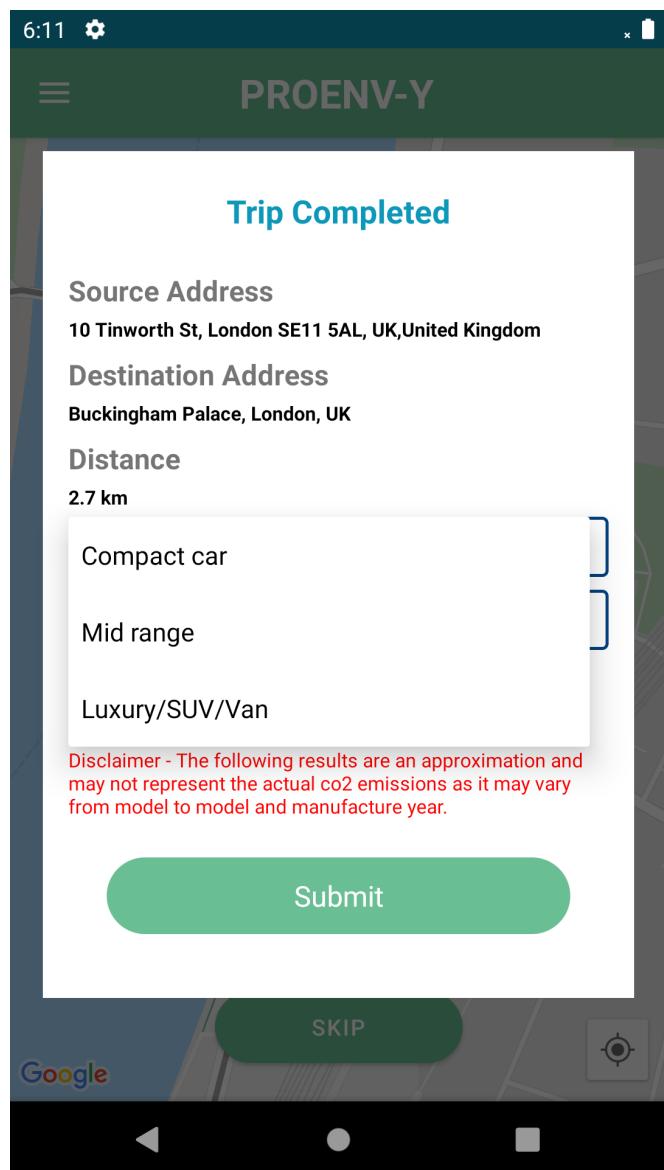


Map Animation

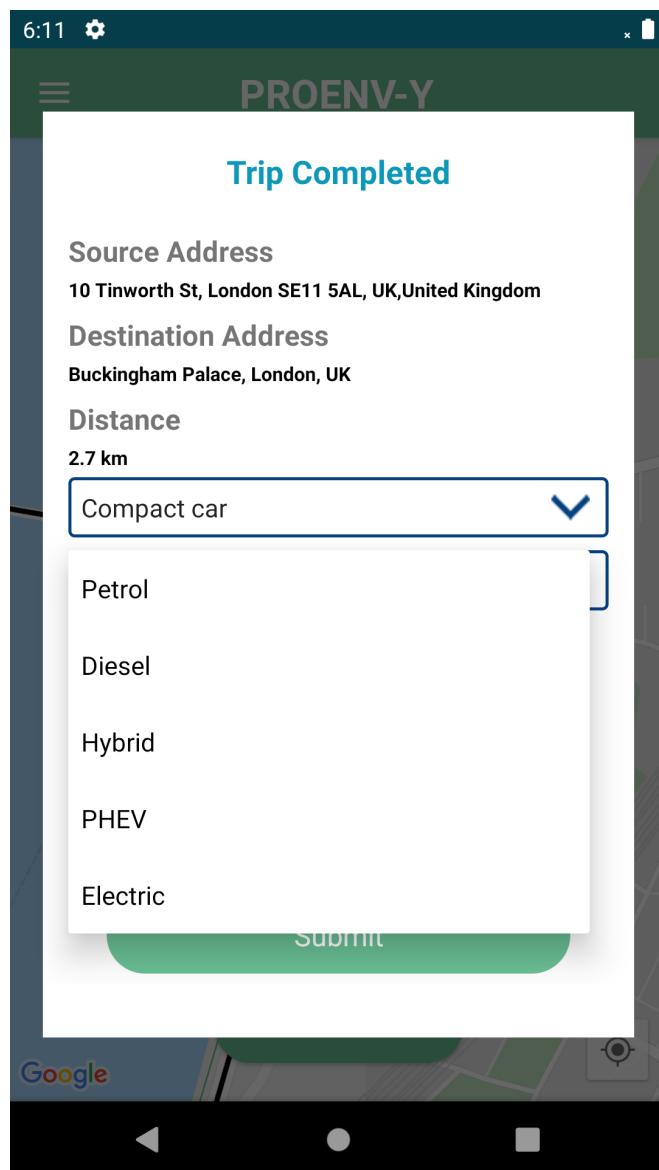


Trip Dialog

Trip Dialog- As soon as the user completes their trip, a pop up box appears on the app showing the source and destination address. It shows the distance covered by the vehicle as well as the total CO₂ emitted by the vehicle in unit kg. Under this, a disclaimer is given in red mentioning that the results calculated by the app is an approximation and may not represent the exact CO₂ emission as this may vary according to different models of the vehicle as well as the manufacturing year. Post this, the user clicks on submit, the trip summary gets saved. If the chosen vehicle by user is the car, 2 scrollable options are made available on the map through which the user can select the car type as well as the fuel.



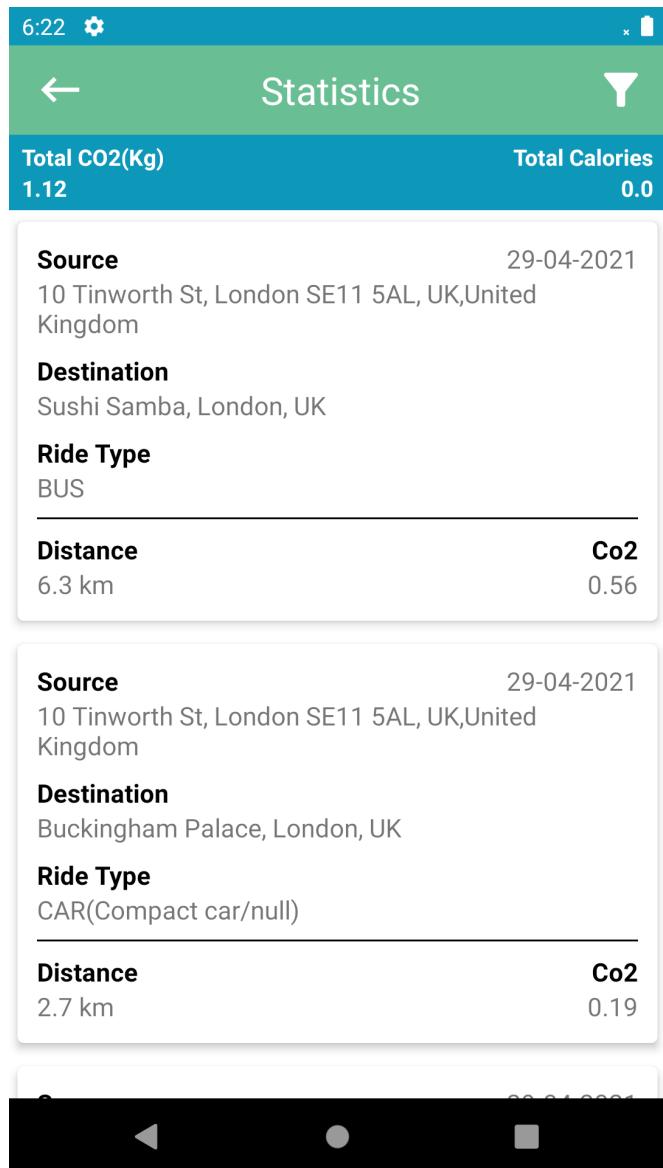
Trip Dialog



Trip Dialog

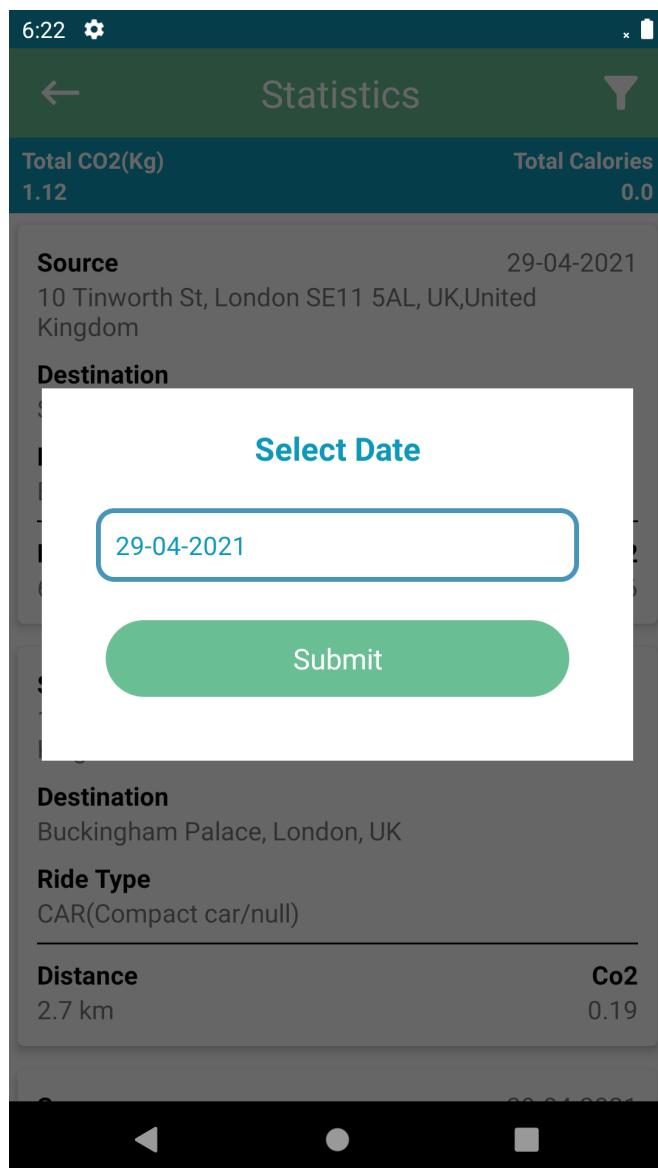
In case the fuel type chosen by the user is electric or PHEV, another disclaimer is displayed on the screen. The user now needs to click on the submit button.

B.6 Statistic Screen



Statistic Home Screen

On the top left of the screen of the home page, a 3 line option is provided containing options such as home, statistic, profile, change password and logout. By clicking on statistics, the client can view all their past trips till date including their riding type, distance covered after every journey, individual co2 burnt after every trip as well as the total co2 emitted or calories burnt. On top of the screen, on the right side the total CO2 counter of the day is displayed in green and to the left, Total calories counter. There is also a filter option on the top right of the screen from which the user can choose a date of their wish and get all the information from that particular date only.



Filter Date dialog

If the date chosen by user has no trip history, then the same would be displayed on the screen.



No trip history available

Filter Date dialog

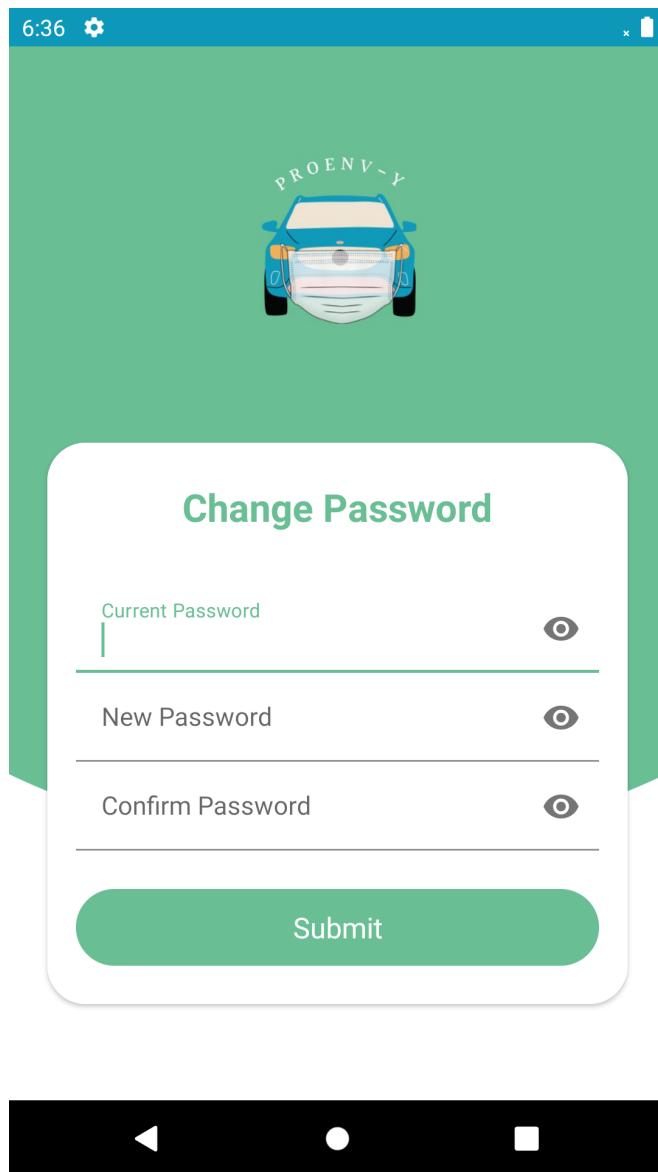
B.7 Profile Screen



Update Profile Screen

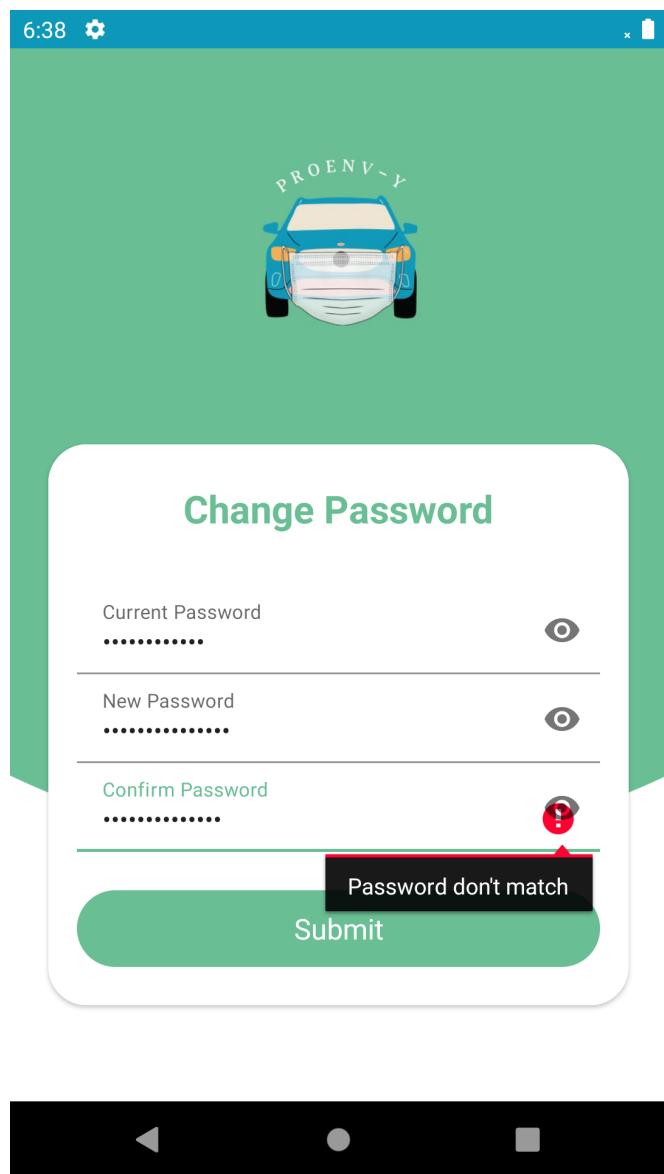
Another interesting feature of the app, is the profile option from the 3 line drop down which contains the information the client put during registration. All of that information can be changed in the app except the email provided by the user during registration.

B.8 Change Password

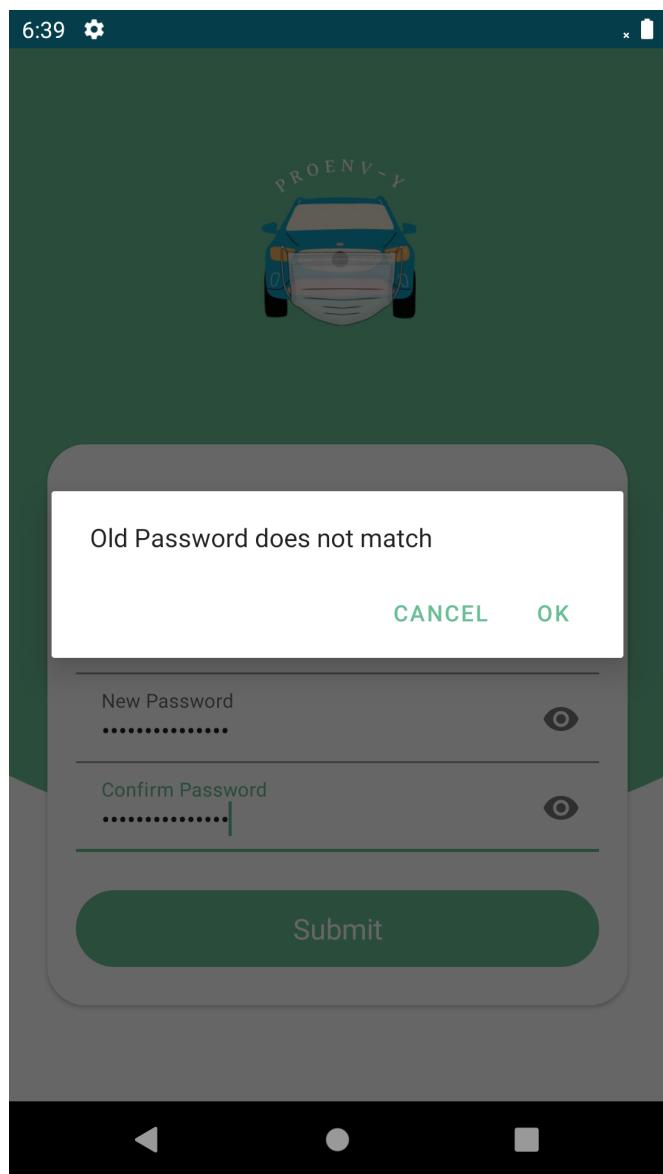


Update Profile Screen

The option of change password is also provided to the user through which they can change their password by providing their current password and then their new password. They will have to confirm their new password and if it fails, the application will prompt with the respective errors.

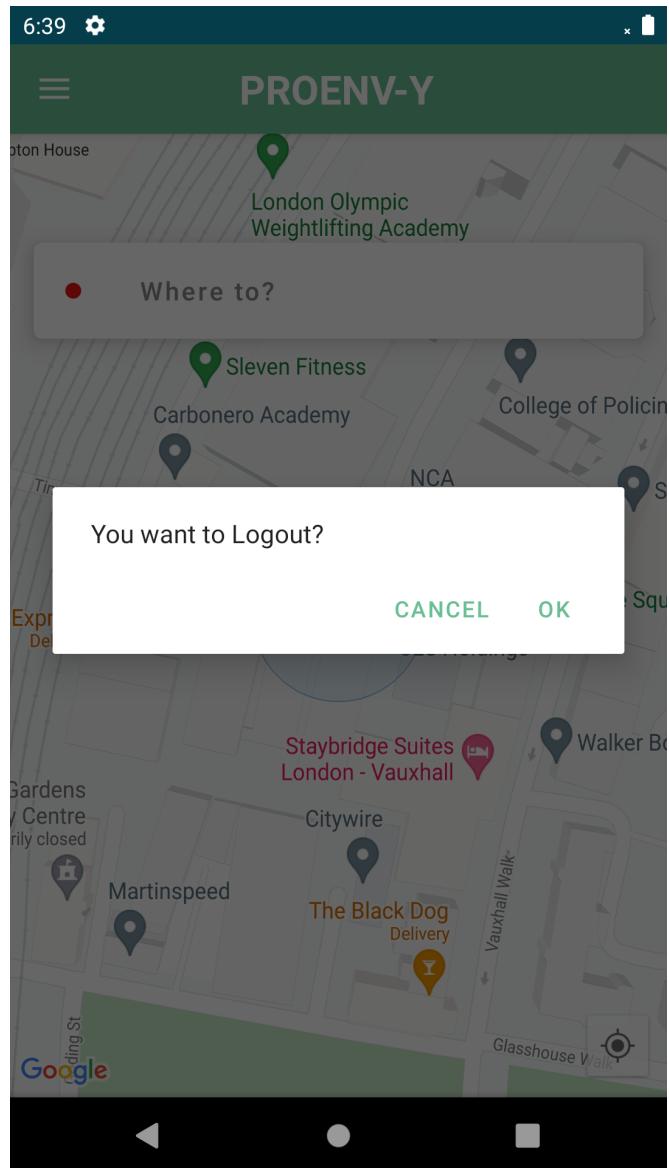


When the confirm password does not match



When the current password does not match with the old one.

B.9 Log out



Log out dialog

When the user chooses the log out option, they can leave the application and will have to sign in when they want to use the app again.

Appendix C

Source Code

I verify that I am the sole author of the programs contained in this folder, except where explicitly stated to the contrary.

Shitin Kamalia

Date: 30 April,2021

Source Code Listing

Activities:

1. Changepasswordactivity
2. HomeActivity
3. LoginActivity
4. PickLocation
5. RegisterActivity
6. SearchActivity
7. SplashActivity
8. StatisticsActivity
9. UpdateProfileActivity

Adapter:

1. PlaceAutocompleteAdapter
2. Statisticadapter

Utils

1. AnimationUtils
2. App
3. Constant
4. CustomDirection
5. CustomMapClusterRenderer
6. CustomModel
7. CustomPickUp
8. CustomProgress
9. DirectionListener
10. DirectionJSONParser
11. DirectionUtils
12. MapUtils
13. MyItem
14. Place
15. PlaceListener
16. ProgressListener

anim

1. slide_down.xml
2. slide_down_animation.xml
3. slide_out_left.xml
4. slide_up_animation.xml

drawable

1. assball.xml

2. avd_line.xml
3. bg_blu_btn.xml
4. bg_bottom_circle.xml
5. bg_btn.xml
6. bg_green_btn.xml
7. bg_maroon_btn.xml
8. bg_pink_btn.xml
9. bg_red_btn.xml
10. bg_white_btn.xml
11. btn_bg_selection.xml
12. btn_bg_selection_bus.xml
13. btn_bg_selection_flight.xml
14. btn_bg_selection_train.xml
15. btn_green_bg_selection.xml
16. btn_text_selection.xml
17. destball.xml
18. edt_bacground.xml
19. greenball.xml
20. ic_back.xml
21. ic_back_white.xml
22. ic_baseline_history_24.xml
23. ic_bicycle.xml
24. ic_bus.xml
25. ic_car.xml
26. ic_filter.xml
27. ic_flight.xml
28. ic_home.xml
29. ic_launcher_background.xml
30. ic_logout.xml
31. ic_profile.xml
32. ic_train.xml
33. ic_trophy.xml
34. ic_user_profile.xml
35. ic_walk.xml
36. shadow.xml
37. spinner_background.xml
38. vd_line.xml

layout

1. activity_change_password.xml
2. activity_home.xml
3. activity_login.xml
4. activity_pick.xml
5. activity_register.xml

6. activity_search.xml
7. activity_splash.xml
8. activity_statistics.xml
9. activity_update_profile.xml
10. app_bar_map.xml
11. custom_marker_infowindow_left.xml
12. custom_marker_infowindow_right.xml
13. dialog_forgot_password.xml
14. dialog_no_route.xml
15. dialog_progress.xml
16. dialog_statistics_filter.xml
17. dialog_trip_complate.xml
18. expandlistitem.xml
19. list_item.xml
20. nav_header_main.xml
21. setloc.xml

menu

1. activity_main_drawer.xml

mipmap

1. ic_launcher.xml
2. ic_launcher_round.xml

transition

1. changebounds.xml

values:

1. colors.xml
2. dimens.xml
3. strings.xml
4. themes.xml

Gradle Scripts:

1. build.gradle (Project:Custom_Co2)
2. build.gradle(Module:Custom_CO2.app)
3. gradle-wrapper-properties
4. proguard-rules.pro
5. gradle.properties
6. settings.gradle

Manifest

```
1 package com.custom.co2.activity;
2
3 import android.app.Dialog;
4 import android.os.Bundle;
5 import android.util.Log;
6 import android.widget.EditText;
7 import android.widget.TextView;
8 import android.widget.Toast;
9
10 import androidx.annotation.NonNull;
11 import androidx.appcompat.app.AppCompatActivity;
12
13 import com.custom.co2.R;
14 import com.custom.co2.utils.Constant;
15 import com.google.android.gms.tasks.OnCompleteListener;
16 import com.google.android.gms.tasks.Task;
17 import com.google.firebase.firestore.DocumentSnapshot;
18 import com.google.firebase.firestore.FirebaseFirestore;
19 import com.google.firebase.firestore.QuerySnapshot;
20
21 import java.util.HashMap;
22 import java.util.Map;
23
24 import static com.custom.co2.utils.Constant.getShaedPref;
25 import static com.custom.co2.utils.Constant.showProgressDialog;
26 import static com.custom.co2.utils.Constant.spEmail;
27
28 public class changepasswordactivity extends AppCompatActivity {
29     FirebaseFirestore db;
30     String TAG = "changepasswordactivity";
31     TextView btnLogin;
32     EditText edt_old_password, edt_new_password, edt_confirm_password;
33     Dialog progressDialog;
34
35     /**
36      * OnCreate method for initialize method
37      */
38     @Override
39     protected void onCreate(Bundle savedInstanceState) {
40         super.onCreate(savedInstanceState);
41         getSupportActionBar().hide();
42         setContentView(R.layout.activity_change_password);
43
44         inits();
45         clicks();
46
47         db = FirebaseFirestore.getInstance();
48         progressDialog = showProgressDialog(changepasswordactivity.this);
49     }
50
51     /**
52      * Method for hancle clicks
53      */
54     private void clicks() {
55
56         btnLogin.setOnClickListener(view -> {
57
58             if (Valid()) {
59                 checkPassword(edt_old_password.getText().toString());
60             }
61         });
62
63     }
64
65     /**
66      * Method for check validations
67      */
```



```
155         }
156     });
157 }
158
159     //Checking if task contains any payload. If no, then update
160     if (task.getResult().size() > 0) {
161         try {
162             registerDataToCloud(edt_new_password.getText().toString());
163         } catch (NullPointerException e) {
164             Log.e(TAG, "NullPointerException: " + e.getMessage());
165         }
166     } else {
167         Constant.showAlertDailogBox(changepasswordactivity.this, "Old Password does not match ");
168     }
169     progressDialog.dismiss();
170 }
171
172
173
174
175 }
```

```
package com.custom.co2.activity;

import android.Manifest;
import android.animation.ValueAnimator;
import android.annotation.SuppressLint;
import android.app.AlertDialog;
import android.app.Dialog;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Point;
import android.location.Location;
import android.os.AsyncTask;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.os.SystemClock;
1 import android.util.DisplayMetrics;
2 import android.util.Log;
3 import android.view.MenuItem;
4 import android.view.View;
5 import android.view.ViewGroup;
6 import android.view.WindowManager;
7 import android.view.animation.Animation;
8 import android.widget.AdapterView;
9 import android.widget.Button;
10 import android.widget.FrameLayout;
11 import android.widget.ImageButton;
12 import android.widget.ImageView;
13 import android.widget.LinearLayout;
14 import android.widget.RelativeLayout;
15 import android.widget.Spinner;
16 import android.widget.TextView;
17
18 import androidx.annotation.NonNull;
19 import androidx.appcompat.app.ActionBarDrawerToggle;
20 import androidx.appcompat.app.AppCompatActivity;
21 import androidx.appcompat.widget.Toolbar;
22 import androidx.cardview.widget.CardView;
23 import androidx.core.app.ActivityCompat;
24 import androidx.core.app.ActivityOptionsCompat;
25 import androidx.core.content.ContextCompat;
26 import androidx.core.view.GravityCompat;
27 import androidx.drawerlayout.widget.DrawerLayout;
28
29 import com.custom.co2.R;
30 import com.custom.co2.utils.AnimationUtils;
31 import com.custom.co2.utils.CustomDirection;
32 import com.custom.co2.utils.CustomMapClusterRenderer;
33 import com.custom.co2.utils.CustomModel;
34 import com.custom.co2.utils.DirectionListener;
35 import com.custom.co2.utils.DirectionUtils;
36 import com.custom.co2.utils.DirectionsJSONParser;
37 import com.custom.co2.utils.MapUtils;
38 import com.custom.co2.utils.MyItem;
39 import com.custom.co2.utils.Place;
40 import com.custom.co2.utils.PlaceListener;
41 import com.google.android.gms.common.ConnectionResult;
42 import com.google.android.gms.common.api.GoogleApiClient;
43 import com.google.android.gms.location.LocationListener;
44 import com.google.android.gms.location.LocationRequest;
45 import com.google.android.gms.location.LocationServices;
46 import com.google.android.gms.maps.CameraUpdate;
47 import com.google.android.gms.maps.CameraUpdateFactory;
48 import com.google.android.gms.maps.GoogleMap;
49 import com.google.android.gms.maps.OnMapReadyCallback;
50 import com.google.android.gms.maps.model.BitmapDescriptor;
51 import com.google.android.gms.maps.model.BitmapDescriptorFactory;
52 import com.google.android.gms.maps.model.LatLng;
53 import com.google.android.gms.maps.model.LatLngBounds;
54 import com.google.android.gms.maps.model.Marker;
55 import com.google.android.gms.maps.model.MarkerOptions;
56 import com.google.android.gms.maps.model.LatLng;
57 import com.google.android.gms.maps.model.LatLngBounds;
58 import com.google.android.gms.maps.model.Marker;
59 import com.google.android.gms.maps.model.MarkerOptions;
60 import com.google.android.gms.maps.model.LatLng;
61 import com.google.android.gms.maps.model.LatLngBounds;
62 import com.google.android.gms.maps.model.Marker;
63 import com.google.android.gms.maps.model.MarkerOptions;
64 import com.google.android.material.navigation.NavigationView;
65 import com.google.firebase.firestore.FirebaseFirestore;
66 import com.google.maps.android.clustering.ClusterManager;
67 import com.google.maps.android.ui.IconGenerator;
68 import com.logicbeans.uberpolylineanimation.MapAnimator;
69 import com.sothree.slidinguppanel.SlidingUpPanelLayout;
70 import com.tmall.ultraviewpager.UltraViewPager;
71
72 import org.json.JSONObject;
73
74 import java.io.BufferedReader;
75 import java.io.IOException;
76 import java.io.InputStream;
77 import java.io.InputStreamReader;
78 import java.net.HttpURLConnection;
79 import java.net.URL;
80 import java.text.SimpleDateFormat;
81 import java.util.ArrayList;
82 import java.util.Date;
83 import java.util.HashMap;
84 import java.util.List;
85 import java.util.Locale;
86 import java.util.Map;
87 import java.util.Objects;
88
89 import de.hdodenhof.circleimageview.CircleImageView;
```

```
import static com.custom.co2.utils.Constant.clearShaedPref;
import static com.custom.co2.utils.Constant.getShaedPref;
import static com.custom.co2.utils.Constant.showProgressDialog;
import static com.custom.co2.utils.Constant.spEmail;
import static com.custom.co2.utils.Constant.spUserId;
import static com.custom.co2.utils.Constant.spUsername;
import static com.custom.co2.utils.MapUtils.getScreenWidth;

public class HomeActivity extends AppCompatActivity implements OnMapReadyCallback, LocationListener,
    PlaceListener, GoogleApiClient.ConnectionCallbacks, GoogleApiClient.OnConnectionFailedListener,
    DirectionListener, GoogleMap.OnCameraIdleListener,
    NavigationView.OnNavigationItemSelectedListener, GoogleMap.OnMarkerClickListener {

    private static final long INTERVAL = 1000 * 5; //1 minute
    private static final long defaultInterval = 1500;
    private long lastTimeClick = 0;
    private static final long FASTEST_INTERVAL = 1000 * 3;
    float screenRatio = 3.3f;
    List<LatLng> polylineList;
    List<String> instruct;
    LinearLayout linMode;
    String trpSource = "";
    String trpDestination = "";
    String trpDistance = "";

    ImageView btn_car, btn_Walking, btn_cycle, btn_Bus, btn_Flight, btn_Train;
    GoogleApiClient mGoogleApiClient;
    LocationRequest mLocationRequest;
    CardView destCard;
    ImageButton back;
    Button dest;
    SlidingUpPanelLayout sliding_layout;
    Button fetch, btn_proceed;
    double totalKM = 0.0;
    String originLatlong = "";
    String destinationLatlong = "";
    String originAddress = "";
    String destinationAddress = "", RideDistance = "";
    UltraViewPager mRecyclerView;
    ViewGroup.MarginLayoutParams marginLayoutParams;
    ViewGroup.MarginLayoutParams marginLayoutParams1;
    //LinearLayout delLin;
    ImageView img;
    TextView txt;
    LatLng latLngUpdate;
    LatLng currentLatlong;
    Dialog progressDialog;
    CircleImageView img_logo;
    TextView tvUsername, tvEmail;
    DirectionUtils util;
    Place placeGet;

    int FuelSelectedPos = 0;
    int VehicleSelectedPos = 0;

    Animation.AnimationListener uptop = new Animation.AnimationListener() {
        @Override
        public void onAnimationStart(Animation animation) {
            destCard.setVisibility(View.GONE);
            back.setVisibility(View.GONE);
        }

        @Override
        public void onAnimationEnd(Animation animation) {
            //googleMap.setPadding(0,0,0,0);
            //googleMap.setPadding(0, 90, 0, getScreenHeight()/3);
        }

        @Override
        public void onAnimationRepeat(Animation animation) {
        }
    };

    Animation.AnimationListener bottomup = new Animation.AnimationListener() {
        @Override
        public void onAnimationStart(Animation animation) {
            dest.setEnabled(false);
            destCard.setVisibility(View.GONE);
        }

        @Override
        public void onAnimationEnd(Animation animation) {
            destCard.setVisibility(View.GONE);
            destCard.clearAnimation();
            MapUtils.createTopDownAnimation(HomeActivity.this, back, uptop);
        }

        @Override
        public void onAnimationRepeat(Animation animation) {
        }
    };

    Animation.AnimationListener backdest = new Animation.AnimationListener() {
        @Override
        public void onAnimationStart(Animation animation) {
            dest.setEnabled(true);
            destCard.setVisibility(View.VISIBLE);
            linMode.setVisibility(View.GONE);
        }

        @Override
        public void onAnimationEnd(Animation animation) {
        }
    };
}
```

```

199
200     @Override
201     public void onAnimationRepeat(Animation animation) {
202     }
203 }
204 Animation.AnimationListener backtop = new Animation.AnimationListener() {
205     @Override
206     public void onAnimationStart(Animation animation) {
207     }
208
209     @Override
210     public void onAnimationEnd(Animation animation) {
211         back.setVisibility(View.GONE);
212         MapUtils.createTopDownAnimation(HomeActivity.this, destCard, backdest);
213     }
214 }
215
216     @Override
217     public void onAnimationRepeat(Animation animation) {
218     }
219 };
220 CameraPosition position;
221 TextView tv_timing_right;
222
223 private GoogleMap googleMap;
224 private SupportMapFragment mapFragment;
225 private Handler handler;
226 private Marker carMarker, carMarker1, desMarker, carMarker2, carMarker3, desMarker2, desMarker3;
227 private String TAG = "HomeActivity";
228
229 private Location mLastLoc;
230
231 private ClusterManager<MyItem> mClusterManager;
232 private Marker originMarker;
233 private Marker destinationMarker;
234 private Marker movingCabMarker = null;
235 private LatLng previousLatLng = null;
236 private LatLng currentLatLng = null;
237 private Runnable runnable;
238 private int index = 0;
239 private String markerFlag = "BUS";
240
241 /**
242 * Method for manage location change requwst
243 */
244 protected void createLocationRequest() {
245     mLocationRequest = new LocationRequest();
246     mLocationRequest.setInterval(INTERVAL);
247     mLocationRequest.setFastestInterval(FASTEAST_INTERVAL);
248     mLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
249 }
250
251 /**
252 * Method for destroy activity
253 */
254 @Override
255 protected void onDestroy() {
256     super.onDestroy();
257 }
258
259 /**
260 * Method for initialize layout
261 */
262 @Override
263 protected void onCreate(Bundle savedInstanceState) {
264     super.onCreate(savedInstanceState);
265     setContentView(R.layout.activity_home);
266     Toolbar toolbar = findViewById(R.id.toolbar);
267     setSupportActionBar(toolbar);
268     DrawerLayout drawer = findViewById(R.id.drawer_layout1);
269     NavigationView navigationView = findViewById(R.id.nav_view1);
270     ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
271             this, drawer, toolbar, R.string.navigation_drawer_open, R.string.navigation_drawer_close);
272     drawer.addDrawerListener(toggle);
273     toggle.syncState();
274     navigationView.setNavigationItemSelectedListener(this);
275
276     View hView = navigationView.getHeaderView(0);
277     img_logo = hView.findViewById(R.id.img_logo);
278     tvUsername = hView.findViewById(R.id.tvUsername);
279     tvEmail = hView.findViewById(R.id.tvEmail);
280     progressDialog = showProgressDialog(HomeActivity.this);
281
282     tvUsername.setText(getShaedPref(HomeActivity.this, spUsername));
283     tvEmail.setText(getShaedPref(HomeActivity.this, spEmail));
284
285     CustomModel.getInstance().setListener(this);
286     CustomDirection.getInstance().setListener(this);
287     mapFragment = (SupportMapFragment) getSupportFragmentManager()
288             .findFragmentById(R.id.map);
289     if (mapFragment != null) {
290         mapFragment.getMapAsync(this);
291     }
292     handler = new Handler();
293     dest = findViewById(R.id.destBtn);
294     destCard = findViewById(R.id.destCard);
295     img = findViewById(R.id.img);
296     txt = findViewById(R.id.txt);
297     fetch = findViewById(R.id.fetch);
298     btn_proceed = findViewById(R.id.btn_proceed);
299     btn_car = findViewById(R.id.btn_car);
300     btn_Walking = findViewById(R.id.btn_Walking);
301     btn_cycle = findViewById(R.id.btn_cycle);
302     btn_Bus = findViewById(R.id.btn_bus);
303     btn_Flight = findViewById(R.id.btn_flight);
304 }
```

```

306
307     btn_Train = findViewById(R.id.btn_train);
308     btn_car.setSelected(true);
309     markerFlag = "CAR";
310     linMode = findViewById(R.id.lin_mode);
311     sliding_layout = findViewById(R.id.sliding_layout);
312     sliding_layout.setPanelState(SlidingUpPanelLayout.PanelState.HIDDEN);
313     MapUtils.createTopDownAnimation(this, destCard, null);
314     back = findViewById(R.id.back);
315     mRecyclerView = findViewById(R.id.recycle);
316     marginLayoutParams = (ViewGroup.MarginLayoutParams) mRecyclerView.getLayoutParams();
317     mRecyclerView.setScrollMode(UltraViewPager.ScrollMode.HORIZONTAL);
318     DisplayMetrics metrics = new DisplayMetrics();
319
320     getWindowManager().getDefaultDisplay().getMetrics(metrics);
321     int usableHeight = metrics.heightPixels;
322     back.setOnClickListener(v -> {
323         if (googleMap != null) {
324             googleMap.clear();
325             polylineList = null;
326             instruct = null;
327         }
328         sliding_layout.setPanelHeight(0);
329         sliding_layout.setPanelState(SlidingUpPanelLayout.PanelState.HIDDEN);
330         MapUtils.createBottomUpAnimation(HomeActivity.this, back, backtop);
331         position = new CameraPosition.Builder()
332             .target(new LatLng(mLastLoc.getLatitude(), mLastLoc.getLongitude()))
333             .zoom(16f)
334             .build();
335
336         CameraUpdate update = CameraUpdateFactory
337             .newCameraPosition
338                 (position);
339         googleMap.animateCamera(update);
340         if (carMarker != null) {
341             carMarker.remove();
342         }
343
344         carMarker = googleMap.addMarker(new MarkerOptions().position(new LatLng(mLastLoc.getLatitude(), mLastLoc.getLongitude())).anchor(0.5f, 0.5f).
345             flat(true).icon(BitmapDescriptorFactory.fromResource(R.mipmap.destination)));
346     });
347     createLocationRequest();
348     dest.setOnClickListener(new View.OnClickListener() {
349         @Override
350         public void onClick(View v) {
351             if (mLastLoc != null) {
352                 Intent intent = new Intent(HomeActivity.this, SearchActivity.class);
353                 intent.putExtra("lat", mLastLoc.getLatitude());
354                 intent.putExtra("lon", mLastLoc.getLongitude());
355                 if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
356                     ActivityOptionsCompat optionsCompat = ActivityOptionsCompat.makeSceneTransitionAnimation(HomeActivity.this, destCard, destCard.getTransitionName());
357                     startActivity(intent, optionsCompat.toBundle());
358                 } else
359                     startActivity(intent);
360             }
361         }
362     });
363     btn_Bus.setOnClickListener(view -> {
364         markerFlag = "BUS";
365         btn_Bus.setSelected(true);
366         btn_car.setSelected(false);
367         btn_Train.setSelected(false);
368         btn_Flight.setSelected(false);
369         btn_Walking.setSelected(false);
370         btn_cycle.setSelected(false);
371         if (placeGet.getSrcLatLng() != null) {
372             String url = getDirectionsUrl(placeGet.getSrcLatLng(), placeGet.getDesrLatLng(), "driving");
373             DownloadTask downloadTask = new DownloadTask();
374             downloadTask.execute(url);
375         }
376     });
377     btn_car.setOnClickListener(view -> {
378         markerFlag = "CAR";
379         btn_Bus.setSelected(false);
380         btn_car.setSelected(true);
381         btn_Train.setSelected(false);
382         btn_Flight.setSelected(false);
383         btn_Walking.setSelected(false);
384         btn_cycle.setSelected(false);
385
386         if (placeGet.getSrcLatLng() != null) {
387             String url = getDirectionsUrl(placeGet.getSrcLatLng(), placeGet.getDesrLatLng(), "driving");
388             DownloadTask downloadTask = new DownloadTask();
389             downloadTask.execute(url);
390         }
391     });
392     btn_Train.setOnClickListener(view -> {
393         markerFlag = "TRAIN";
394         btn_Bus.setSelected(false);
395         btn_car.setSelected(false);
396         btn_Train.setSelected(true);
397         btn_Flight.setSelected(false);
398         btn_Walking.setSelected(false);
399         btn_cycle.setSelected(false);
400
401         if (placeGet.getSrcLatLng() != null) {
402             String url = getDirectionsUrl(placeGet.getSrcLatLng(), placeGet.getDesrLatLng(), "driving");
403             DownloadTask downloadTask = new DownloadTask();
404             downloadTask.execute(url);
405         }
406     });
407     btn_Flight.setOnClickListener(view -> {
408         markerFlag = "FLIGHT";
409         btn_Bus.setSelected(false);
410         btn_car.setSelected(false);
411         btn_Train.setSelected(false);
412         btn_Flight.setSelected(true);

```

```

413     btn_Walking.setSelected(false);
414     btn_cycle.setSelected(false);
415     if (placeGet.getSrcLatLng() != null) {
416         String url = getDirectionsUrl(placeGet.getSrcLatLng(), placeGet.getDesrLatLng(), "driving");
417         DownloadTask downloadTask = new DownloadTask();
418         downloadTask.execute(url);
419     }
420 });
421 btn_Walking.setOnClickListener(view -> {
422     markerFlag = "WALK";
423     btn_Bus.setSelected(false);
424     btn_car.setSelected(false);
425     btn_Train.setSelected(false);
426     btn_Flight.setSelected(false);
427     btn_Walking.setSelected(true);
428     btn_cycle.setSelected(false);
429     if (placeGet.getSrcLatLng() != null) {
430         String url = getDirectionsUrl(placeGet.getSrcLatLng(), placeGet.getDesrLatLng(), "walking");
431         DownloadTask downloadTask = new DownloadTask();
432         downloadTask.execute(url);
433     }
434 });
435 btn_cycle.setOnClickListener(view -> {
436     markerFlag = "CYCLE";
437     btn_Bus.setSelected(false);
438     btn_car.setSelected(false);
439     btn_Train.setSelected(false);
440     btn_Flight.setSelected(false);
441     btn_Walking.setSelected(false);
442     btn_cycle.setSelected(true);
443     if (placeGet.getSrcLatLng() != null) {
444         String url = getDirectionsUrl(placeGet.getSrcLatLng(), placeGet.getDesrLatLng(), "walking");
445         DownloadTask downloadTask = new DownloadTask();
446         downloadTask.execute(url);
447     }
448 });
449 });
450
451
452 FrameLayout frameLayout = findViewById(R.id.sliderLin);
453 marginLayoutParams1 = (ViewGroup.MarginLayoutParams) frameLayout.getLayoutParams();
454
455 marginLayoutParams1.height = (int) (usableHeight / screenRatio);
456 frameLayout.setLayoutParams(marginLayoutParams1);
457
458 }
459
460 /**
461 * Override method for get place data
462 */
463 @Override
464 public void onPlace(Place place) {
465     this.placeGet = place;
466     MapUtils.createBottomUpAnimation(this, destCard, bottomup);
467
468     DisplayMetrics metrics = new DisplayMetrics();
469     getWindowManager().getDefaultDisplay().getMetrics(metrics);
470
471     btn_proceed.setVisibility(View.VISIBLE);
472
473     btn_proceed.setOnClickListener(view -> {
474         if(SystemClock.elapsedRealtime() - lastTimeClick < defaultInterval){
475             return;
476         }
477         lastTimeClick = SystemClock.elapsedRealtime();
478         if (btn_proceed.getText().toString().equals("Start")) {
479             new Handler().postDelayed(() -> showMovingCab(polyLineList), 100);
480             linMode.setVisibility(View.GONE);
481             btn_proceed.setText("Skip");
482         } else {
483             TripDialog();
484         }
485     });
486
487     if (place.getSrcLatLng() != null) {
488         String url = getDirectionsUrl(place.getSrcLatLng(), place.getDesrLatLng(), "driving");
489         DownloadTask downloadTask = new DownloadTask();
490         downloadTask.execute(url);
491     }
492 }
493
494 /**
495 * Override Method for get direction data
496 */
497 @Override
498 public void onDirectionsData(DirectionUtils utils) {
499     this.util = utils;
500     Log.e("distac", util.getDistance());
501 }
502
503 /**
504 * Method called when activity resumed
505 */
506 @Override
507 public void onResume() {
508     super.onResume();
509
510     if (ContextCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED) {
511         if (mGoogleApiClient != null && mGoogleApiClient.isConnected()) {
512             LocationServices.FusedLocationApi.requestLocationUpdates(mGoogleApiClient, mLocationRequest, this);
513         }
514     }
515
516     if (tvUsername != null && tvEmail != null) {
517         tvUsername.setText(getShaedPref(HomeActivity.this, spUsername));
518         tvEmail.setText(getShaedPref(HomeActivity.this, spEmail));
519     }

```



```

627         googleMap.setOnCameraIdleListener(this);
628     }
629   } else {
630     makeLocationPermissionRequest();
631   }
632 }
633 }
634 }
635 }
636 /**
637 * Method for prepare direction url
638 */
639 private String getDirectionsUrl(LatLng origin, LatLng dest, String mapType) {
640   String url = "";
641   String strOrigin = "origin=" + origin.latitude + "," + origin.longitude;
642   String strDest = "destination=" + dest.latitude + "," + dest.longitude;
643   String strMode = "";
644   String strKey = "key=" + getResources().getString(R.string.Map_Api_Key);
645
646   if (mapType.equals("driving")) {
647     strMode = "mode=driving";
648     url = "https://maps.googleapis.com/maps/api/directions/json?" + strOrigin + "&" + strDest + "&" + strMode + "&" + strKey;
649   } else {
650     strMode = "mode=walking";
651     url = "https://maps.googleapis.com/maps/api/directions/json?" + strOrigin + "&" + strDest + "&" + strMode + "&" + strKey;
652   }
653 }
654 return url;
655 }
656
657 /**
658 * Async Method for grab data from endpoint from url
659 */
660 private String downloadUrl(String strUrl) throws IOException {
661   String data = "";
662   InputStream iStream = null;
663   HttpURLConnection urlConnection = null;
664   try {
665     URL url = new URL(strUrl);
666     // Creating an http connection to communicate with url
667     urlConnection = (HttpURLConnection) url.openConnection();
668     // Connecting to url
669     urlConnection.connect();
670     // Reading data from url
671     iStream = urlConnection.getInputStream();
672     BufferedReader br = new BufferedReader(new InputStreamReader(iStream));
673     StringBuilder sb = new StringBuilder();
674     String line = "";
675     while ((line = br.readLine()) != null) {
676       sb.append(line);
677     }
678     data = sb.toString();
679     br.close();
680   } catch (Exception e) {
681     Log.d("Exceptiondownloading", e.toString());
682   } finally {
683     if (iStream != null) {
684       iStream.close();
685     }
686     if (urlConnection != null) {
687       urlConnection.disconnect();
688     }
689   }
690 }
691 return data;
692 }
693
694 @Override
695 public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {
696 }
697
698 /**
699 * Method for callback when side drawer item clicked
700 */
701 @Override
702 public boolean onNavigationItemSelected(MenuItem menuItem) {
703   int id = menuItem.getItemId();
704
705   if (id == R.id.nav_home) {
706     startActivity(new Intent(getApplicationContext(), HomeActivity.class).setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK));
707   } else if (id == R.id.nav_Profile) {
708     startActivity(new Intent(getApplicationContext(), UpdateProfileActivity.class));
709   } else if (id == R.id.nav_Score_Board) {
710     startActivity(new Intent(getApplicationContext(), StatisticsActivity.class));
711   } else if (id == R.id.nav_Change_Password) {
712     startActivity(new Intent(getApplicationContext(), changepasswordactivity.class));
713   } else if (id == R.id.nav_Logout) {
714     new AlertDialog.Builder(HomeActivity.this)
715       .setMessage("You want to Logout?")
716       .setCancelable(false)
717       .setPositiveButton(android.R.string.yes, (dialog, which) -> {
718         clearShaedPref(HomeActivity.this);
719
720         startActivity(new Intent(getApplicationContext(), LoginActivity.class).setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK));
721       })
722       .setNegativeButton(android.R.string.no, null)
723       .show();
724   }
725
726   DrawerLayout drawer = findViewById(R.id.drawer_layout1);
727   drawer.closeDrawer(GravityCompat.START);
728   return true;
729 }
730
731 /**
732 */

```

```

734 * Method for callback when market clicked
735 */
736 @Override
737 public boolean onMarkerClick(Marker marker) {
738     return false;
739 }
740 /**
741 * Method for start getting polyline
742 */
743 void staticPolyLine() {
744     googleMap.clear();
745     Log.e("Size", String.valueOf(polyLineList.get(0)));
746     final int MAP_BOUND_PADDING = 180;
747     LatLngBounds.Builder builder = new LatLngBounds.Builder();
748     for (LatLng latLng : polyLineList) {
749         builder.include(latLng);
750     }
751     final LatLngBounds bounds = builder.build();
752
753     CameraUpdate mCameraUpdate = CameraUpdateFactory.newLatLngBounds(bounds, MAP_BOUND_PADDING);
754     googleMap.animateCamera(mCameraUpdate);
755
756     MapAnimator.getInstance().animateRoute(googleMap, polyLineList);
757     startCarAnimation(polyLineList.get(0).latitude, polyLineList.get(0).longitude);
758 }
759 /**
760 * Override method called when map is ready
761 */
762 @SuppressLint("ResourceType")
763 @Override
764 public void onMapReady(GoogleMap googleMap) {
765     this.googleMap = googleMap;
766     this.googleMap.getUiSettings().setCompassEnabled(false);
767     this.googleMap.getUiSettings().setMapToolbarEnabled(false);
768
769     if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION)
770         != PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
771         return;
772     }
773     this.googleMap.setMyLocationEnabled(true);
774
775     View mapView = mapFragment.getView();
776     if (mapView != null && mapView.findViewById(1) != null) {
777         View locationButton = ((View) mapView.findViewById(1).getParent()).findViewById(2);
778         RelativeLayout.LayoutParams layoutParams = (RelativeLayout.LayoutParams)
779             locationButton.getLayoutParams();
780         layoutParams.addRule(RelativeLayout.ALIGN_PARENT_TOP, 0);
781         layoutParams.addRule(RelativeLayout.ALIGN_PARENT_BOTTOM, RelativeLayout.TRUE);
782         layoutParams.setMargins(0, 0, 30, 30);
783     }
784
785     if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
786         int locationPermission = ContextCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION);
787         int locationPermission2 = ContextCompat.checkSelfPermission(this, Manifest.permission.ACCESS_COARSE_LOCATION);
788         if (locationPermission != PackageManager.PERMISSION_GRANTED && locationPermission2 != PackageManager.PERMISSION_GRANTED) {
789             makeLocationPermissionRequest();
790         } else {
791             if (mGoogleApiClient.isConnected()) {
792                 LocationServices.FusedLocationApi.requestLocationUpdates(mGoogleApiClient, mLocationRequest, this);
793             }
794             googleMap.setOnCameraIdleListener(this);
795         }
796     }
797
798     setUpClusterer();
799 }
800
801 /**
802 * Method for callback when location changed
803 */
804 @Override
805 public void onLocationChanged(final Location location) {
806     if (polyLineList == null || polyLineList.size() == 0) {
807         if (mLastLoc == null) {
808             position = new CameraPosition.Builder()
809                 .target(new LatLng(location.getLatitude(), location.getLongitude()))
810                 .zoom(18f)
811                 .build();
812
813             CameraUpdate update = CameraUpdateFactory
814                 .newCameraPosition(position);
815
816             googleMap.animateCamera(update);
817         }
818         mLastLoc = location;
819     }
820 }
821
822 /**
823 * Method for set up map cluster
824 */
825 private void setUpClusterer() {
826     mClusterManager = new ClusterManager<>(this, googleMap);
827     googleMap.setOnCameraIdleListener(mClusterManager);
828     googleMap.setOnMarkerClickListener(mClusterManager);
829     mClusterManager.setAnimation(false);
830     mClusterManager.setRenderer(new CustomMapClusterRenderer<>(this, googleMap, mClusterManager));
831 }
832
833 /**
834 * Method for prepare polyline markers and ready for start car animation
835 */

```

```

842     @SuppressLint("SetTextI18n")
843     private void startCarAnimation(Double latitude, Double longitude) {
844         LatLng latLng = new LatLng(latitude, longitude);
845         currentLatlong = latLng;
846         if (carMarker != null) {
847             carMarker.remove();
848         }
849
850         originMarker = addOriginDestinationMarkerAndGet(latLng);
851         originMarker.setAnchor(1f, 1f);
852         destinationMarker = addOriginDestinationMarkerAndGet(polyLineList.get(polyLineList.size() - 1));
853         destinationMarker.setAnchor(1f, 1f);
854
855         IconGenerator iconFactory_left = new IconGenerator(this);
856         iconFactory_left.setBackground(null);
857         View view_left = View.inflate(this, R.layout.custom_marker_infowindow_left, null);
858         iconFactory_left.setContentView(view_left);
859         LinearLayout ll_marker_left = view_left.findViewById(R.id.ll_marker_right);
860         TextView tv_address_left = view_left.findViewById(R.id.tv_address_right);
861         tv_address_left.setText(placeGet.getSrcAddress());
862
863
864         originAddress = tv_address_left.getText().toString();
865         ll_marker_left.setPadding((int) (getScreenWidth() / 1.9), 0, 0, 0);
866         latLngUpdate = latLng;
867         carMarker2 = googleMap.addMarker(new MarkerOptions().position(latLng)
868             .anchor(0.5f, 0.5f)
869             .icon(BitmapDescriptorFactory.fromBitmap(iconFactory_left.makeIcon("current"))));
870
871         IconGenerator iconFactory_left2 = new IconGenerator(this);
872         iconFactory_left2.setBackground(null);
873         View view_left2 = View.inflate(this, R.layout.custom_marker_infowindow_left, null);
874         iconFactory_left2.setContentView(view_left2);
875         LinearLayout ll_marker_left2 = view_left2.findViewById(R.id.ll_marker_right);
876         TextView tv_timing_left2 = view_left2.findViewById(R.id.tv_timing_right);
877         tv_timing_left2.setText("5 mins");
878         TextView tv_address_left2 = view_left2.findViewById(R.id.tv_address_right);
879         tv_address_left2.setText(placeGet.getSrcAddress());
880         ll_marker_left2.setPadding(0, 0, (int) (getScreenWidth() / 1.9), 0);
881         carMarker3 = googleMap.addMarker(new MarkerOptions().position(latLng)
882             .anchor(0.5f, 0.5f)
883             .icon(BitmapDescriptorFactory.fromBitmap(iconFactory_left2.makeIcon("current2"))));
884         carMarker3.setVisible(false);
885
886
887         IconGenerator iconFactory_right = new IconGenerator(this);
888         iconFactory_right.setBackground(null);
889         View view_right = View.inflate(this, R.layout.custom_marker_infowindow_right, null);
890         iconFactory_right.setContentView(view_right);
891         tv_timing_right = view_right.findViewById(R.id.tv_timing_right);
892         if (util != null) {
893             tv_timing_right.setText(util.getDuration());
894
895             RideDistance = util.getDuration();
896             trpDistance = util.getDistance();
897             String[] separated = util.getDistance().replace(" ", "").split(" ");
898             totalKM = Double.parseDouble(separated[0]);
899             originLatlong = placeGet.getSrcLat() + " " + placeGet.getSrcLon();
900             destinationLatlong = placeGet.getDestLat() + " " + placeGet.getDestLon();
901             Log.e("ddddddd", originLatlong + ":" + destinationLatlong);
902         }
903         TextView tv_address_right = view_right.findViewById(R.id.tv_address_right);
904         tv_address_right.setText(placeGet.getDestAddress());
905         destinationAddress = tv_address_right.getText().toString();
906         Log.e("lat_lng", originAddress + "\n\n" + destinationAddress);
907
908
909         LinearLayout ll_marker_right = view_right.findViewById(R.id.ll_marker_right);
910         ll_marker_right.setPadding((int) (getScreenWidth() / 1.9), 0, 0, 0);
911         desMarker2 = googleMap.addMarker(new MarkerOptions().position(polyLineList.get(polyLineList.size() - 1))
912             .anchor(0.5f, 0.5f)
913             .icon(BitmapDescriptorFactory.fromBitmap(iconFactory_right.makeIcon("destination"))));
914
915         IconGenerator iconFactory_right2 = new IconGenerator(this);
916         iconFactory_right2.setBackground(null);
917         View view_right2 = View.inflate(this, R.layout.custom_marker_infowindow_right, null);
918         iconFactory_right2.setContentView(view_right2);
919         tv_timing_right = view_right2.findViewById(R.id.tv_timing_right);
920         if (util != null) {
921             tv_timing_right.setText(util.getDuration());
922
923             RideDistance = util.getDistance();
924             trpDistance = util.getDistance();
925         }
926         TextView tv_address_right2 = view_right2.findViewById(R.id.tv_address_right);
927         tv_address_right2.setText(placeGet.getDestAddress());
928         trpSource = placeGet.getSrcAddress().toString();
929         trpDestination = placeGet.getDestAddress().toString();
930         trpDistance = "";
931         LinearLayout ll_marker_right2 = view_right2.findViewById(R.id.ll_marker_right);
932         ll_marker_right2.setPadding(0, 0, (int) (getScreenWidth() / 1.9), 0);
933         desMarker3 = googleMap.addMarker(new MarkerOptions().position(polyLineList.get(polyLineList.size() - 1))
934             .anchor(0.5f, 0.5f)
935             .icon(BitmapDescriptorFactory.fromBitmap(iconFactory_right2.makeIcon("destination2"))));
936         desMarker3.setVisible(false);
937
938         double tempDistance = 0.0;
939         tempDistance = Double.parseDouble(RideDistance.replace(" km", "").replace(" m", "").replace(" ", ""));
940
941         if (tempDistance > 500) {
942             btn_Flight.setVisibility(View.VISIBLE);
943             btn_Walking.setVisibility(View.GONE);
944             btn_cycle.setVisibility(View.GONE);
945         } else if (tempDistance > 50 && tempDistance <= 500) {
946             btn_Flight.setVisibility(View.GONE);
947             btn_Walking.setVisibility(View.GONE);
948             btn_cycle.setVisibility(View.GONE);

```

```

949 } else {
950     btn_Flight.setVisibility(View.GONE);
951     btn_Walking.setVisibility(View.VISIBLE);
952     btn_cycle.setVisibility(View.VISIBLE);
953 }
954
955 linMode.setVisibility(View.VISIBLE);
956 }
957
958 /**
959 * Method for callback when camera is Idle
960 */
961 @Override
962 public void onCameralidle() {
963     Projection projection = googleMap.getProjection();
964     LatLng markerLocation = null;
965     LatLng dropMarkerLocation = null;
966     try {
967         if (carMarker != null) {
968             markerLocation = carMarker.getPosition();
969         } else {
970             markerLocation = new LatLng(0.0f, 0.0f);
971         }
972         if (desMarker != null) {
973             dropMarkerLocation = desMarker.getPosition();
974         } else {
975             dropMarkerLocation = new LatLng(0.0f, 0.0f);
976         }
977         Point screenPosition = projection.toScreenLocation(markerLocation);
978         Point screenPositiondrop = projection.toScreenLocation(dropMarkerLocation);
979
980         final View mapview = mapFragment.getView();
981         float maxX = 0;
982         if (mapview != null) {
983             maxX = mapview.getMeasuredWidth();
984         }
985     }
986
987     if (carMarker2 != null) {
988
989         if (screenPosition.x > 100 && screenPosition.x > maxX / 2) {
990             Log.e("Side", "CarLeft");
991             carMarker3.setVisible(true);
992             carMarker2.setVisible(false);
993         } else {
994             carMarker2.setVisible(true);
995             carMarker3.setVisible(false);
996             Log.e("Side", "CarRight");
997         }
998     }
999     if (desMarker != null) {
1000         if (screenPositiondrop.x > 100 && screenPositiondrop.x > maxX / 2) {
1001             desMarker3.setVisible(true);
1002             desMarker2.setVisible(false);
1003         } else {
1004             desMarker2.setVisible(true);
1005             desMarker3.setVisible(false);
1006             Log.e("Side", "DesRight");
1007         }
1008     }
1009 }
1010
1011 } catch (Exception e) {
1012     e.printStackTrace();
1013 }
1014 }
1015
1016 /**
1017 * Method for activity lifecycle when activity is in background
1018 */
1019 @Override
1020 protected void onPause() {
1021     super.onPause();
1022 }
1023
1024 /**
1025 * Method for show car animation & moving cab
1026 */
1027 private void showMovingCab(final List<LatLng> cabLatLngList) {
1028     handler = new Handler();
1029     runnable = () -> {
1030         if (cabLatLngList != null && index < cabLatLngList.size()) {
1031             updateCarLocation(cabLatLngList.get(index));
1032             handler.postDelayed(runnable, 900);
1033             ++index;
1034         } else {
1035             handler.removeCallbacks(runnable);
1036             TripDialog();
1037         }
1038     };
1039     handler.postDelayed(runnable, 1500);
1040 }
1041
1042 /**
1043 * Method for update pinpoint location to show animation
1044 */
1045 private void updateCarLocation(LatLng latLng) {
1046     if (movingCabMarker == null) {
1047         movingCabMarker = addCarMarkerAndGet(latLng);
1048     }
1049     if (previousLatLng == null) {
1050         currentLatLng = latLng;
1051         previousLatLng = currentLatLng;
1052         movingCabMarker.setPosition(currentLatLng);
1053         movingCabMarker.setAnchor(0.5f, 0.5f);
1054     } else {
1055         previousLatLng = currentLatLng;

```

```

1056
1057     currentLatLng = latLng;
1058     ValueAnimator valueAnimator = AnimationUtils.carAnimator();
1059     valueAnimator.addUpdateListener(animation -> {
1060         if (currentLatLng != null && previousLatLng != null) {
1061             float multiplier = animation.getAnimatedFraction();
1062             LatLng nextLocation = new LatLng(
1063                 multiplier * currentLatLng.latitude + (1 - multiplier) * previousLatLng.latitude,
1064                 multiplier * currentLatLng.longitude + (1 - multiplier) * previousLatLng.longitude);
1065             movingCabMarker.setPosition(nextLocation);
1066             float rotation = MapUtils.getRotation(previousLatLng, nextLocation);
1067             movingCabMarker.setRotation(rotation);
1068             movingCabMarker.setAnchor(0.5f, 0.5f);
1069         }
1070     });
1071     valueAnimator.start();
1072 }
1073 animateCamera(latLng);
1074
1075 /**
1076 * Method for show dialog when no route is found
1077 */
1078 private void NoRouteDialog() {
1079     final Dialog dialog = new Dialog(HomeActivity.this);
1080     dialog.setContentView(R.layout.dialog_no_route);
1081     Objects.requireNonNull(dialog.getWindow()).setBackgroundDrawableResource(android.R.color.transparent);
1082     dialog.setCancelable(false);
1083     DisplayMetrics displaymetrics = new DisplayMetrics();
1084     this.getWindowManager().getDefaultDisplay().getMetrics(displaymetrics);
1085     WindowManager.LayoutParams lp = new WindowManager.LayoutParams();
1086     lp.copyFrom(dialog.getWindow().getAttributes());
1087     lp.width = (int) (displaymetrics.widthPixels * 0.90);
1088     dialog.getWindow().setAttributes(lp);
1089     TextView btnSubmit = dialog.findViewById(R.id.btn_submit);
1090     btnSubmit.setOnClickListener(new View.OnClickListener() {
1091         @Override
1092         public void onClick(View view) {
1093             dialog.dismiss();
1094             startActivity(new Intent(getApplicationContext(), HomeActivity.class).setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK));
1095         }
1096     });
1097     dialog.show();
1098 }
1099
1100 /**
1101 * Method for show trip dialog
1102 */
1103 private void TripDialog() {
1104     Boolean isCo2 = false;
1105
1106     final Dialog dialog = new Dialog(HomeActivity.this);
1107     dialog.setContentView(R.layout.dialog_trip_complate);
1108     Objects.requireNonNull(dialog.getWindow()).setBackgroundDrawableResource(android.R.color.transparent);
1109     dialog.setCancelable(false);
1110     DisplayMetrics displaymetrics = new DisplayMetrics();
1111     this.getWindowManager().getDefaultDisplay().getMetrics(displaymetrics);
1112     WindowManager.LayoutParams lp = new WindowManager.LayoutParams();
1113     lp.copyFrom(dialog.getWindow().getAttributes());
1114     lp.width = (int) (displaymetrics.widthPixels * 0.90);
1115     dialog.getWindow().setAttributes(lp);
1116     TextView txtSource = dialog.findViewById(R.id.txtsourcAddress);
1117     TextView txtDestination = dialog.findViewById(R.id.txtDestinationAddress);
1118     TextView txtDistance = dialog.findViewById(R.id.txtDestence);
1119     TextView txtCo2 = dialog.findViewById(R.id.txtCO2);
1120     TextView txtCo2Title = dialog.findViewById(R.id.txtCO2Title);
1121
1122     TextView txtdec1 = dialog.findViewById(R.id.txtdec1);
1123     TextView txtdec2 = dialog.findViewById(R.id.txtdec2);
1124
1125     Spinner spFuelType = dialog.findViewById(R.id.spFuelType);
1126     Spinner spVehicleType = dialog.findViewById(R.id.spVehicleType);
1127
1128     if (markerFlag.equals("CAR")) {
1129         isCo2 = true;
1130         spFuelType.setVisibility(View.VISIBLE);
1131         spVehicleType.setVisibility(View.VISIBLE);
1132     } else if (markerFlag.equals("BUS") || markerFlag.equals("TRAIN") || markerFlag.equals("FLIGHT")) {
1133         isCo2 = true;
1134         spFuelType.setVisibility(View.GONE);
1135         spVehicleType.setVisibility(View.GONE);
1136         double BasicCo2 = 0.0;
1137         double co2value;
1138
1139
1140         if (markerFlag.equals("BUS")) {
1141             BasicCo2 = 0.089;
1142         } else if (markerFlag.equals("TRAIN")) {
1143             BasicCo2 = 0.06;
1144         } else {
1145             BasicCo2 = 0.1753;
1146         }
1147
1148
1149         if (RideDistance.contains(" km")) {
1150             co2value = BasicCo2 * Double.parseDouble(RideDistance.replace(" km", "").replace(",", ""));
1151         } else {
1152             co2value = BasicCo2 * (Double.parseDouble(RideDistance.replace(" m", "").replace(",", "")) / 1000);
1153         }
1154         txtCo2.setText(String.format("%.2f", co2value));
1155     } else {
1156         isCo2 = false;
1157         spFuelType.setVisibility(View.GONE);
1158         spVehicleType.setVisibility(View.GONE);
1159         txtdec2.setVisibility(View.GONE);
1160         txtCo2Title.setText("Total Calories");
1161         double basiccalories;
1162         double co2value;
1163     }

```

```

1153
1154     if (markerFlag.equals("WALK")) {
1155         basiccalories = 47;
1156     } else {
1157         basiccalories = 25;
1158     }
1159
1160     if (RideDistance.contains(" km")) {
1161         co2value = basiccalories * Double.parseDouble(RideDistance.replace(" km", ""));
1162     } else {
1163         co2value = basiccalories * (Double.parseDouble(RideDistance.replace(" m", "")) / 1000);
1164     }
1165     txtCo2.setText(String.format("%.2f", co2value));
1166
1167
1168     txtSource.setText(trpSource);
1169     txtDestination.setText(trpDestination);
1170     txtDistance.setText(RideDistance);
1171
1172     spFuelType.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
1173         public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
1174             if (position == 3 || position == 4) {
1175                 txtdec1.setVisibility(View.VISIBLE);
1176             } else {
1177                 txtdec1.setVisibility(View.GONE);
1178             }
1179             FuelSelectedPos = position;
1180             txtCo2.setText(String.format("%.2f",
1181                     calculateCO2(RideDistance)));
1182         }
1183
1184         public void onNothingSelected(AdapterView<?> parent) {
1185     });
1186
1187     spVehicleType.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
1188         public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
1189
1190             VehicleSelectedPos = position;
1191             txtCo2.setText(String.format("%.2f", calculateCO2(RideDistance)));
1192         }
1193
1194         public void onNothingSelected(AdapterView<?> parent) {
1195     });
1196
1197     btnSubmit.setOnClickListener(view -> {
1198         progressDialog.show();
1199         AddTripRecord(txtSource.getText().toString(),
1200             txtDestination.getText().toString(),
1201             txtDistance.getText().toString(),
1202             txtCo2.getText().toString(),
1203             dialog,
1204             finalIsCo,
1205             spFuelType.getSelectedItem().toString(),
1206             spVehicleType.getSelectedItem().toString(), markerFlag
1207         );
1208
1209         Boolean finalIsCo = isCo2;
1210         btnSubmit.setOnClickListener(view -> {
1211             progressDialog.show();
1212             AddTripRecord(txtSource.getText().toString(),
1213                 txtDestination.getText().toString(),
1214                 txtDistance.getText().toString(),
1215                 txtCo2.getText().toString(),
1216                 dialog,
1217                 finalIsCo,
1218                 spFuelType.getSelectedItem().toString(),
1219                 spVehicleType.getSelectedItem().toString(), markerFlag
1220             );
1221
1222         });
1223     });
1224     if (!this.isFinishing())
1225         dialog.show();
1226 }
1227
1228 /**
1229  * Method for calculate how much co2 consumed
1230 */
1231 private double calculateCO2(String distance) {
1232
1233     double basicValue = 0;
1234     double co2value = 0;
1235     if (FuelSelectedPos == 0 && VehicleSelectedPos == 0) {
1236         basicValue = 0.28;
1237     } else if (FuelSelectedPos == 0 && VehicleSelectedPos == 1) {
1238         basicValue = 0.34;
1239     } else if (FuelSelectedPos == 0 && VehicleSelectedPos == 2) {
1240         basicValue = 0.41;
1241     } else if (FuelSelectedPos == 1 && VehicleSelectedPos == 0) {
1242         basicValue = 0.24;
1243     } else if (FuelSelectedPos == 1 && VehicleSelectedPos == 1) {
1244         basicValue = 0.31;
1245     } else if (FuelSelectedPos == 1 && VehicleSelectedPos == 2) {
1246         basicValue = 0.39;
1247     } else if (FuelSelectedPos == 2 && VehicleSelectedPos == 0) {
1248         basicValue = 0.21;
1249     } else if (FuelSelectedPos == 2 && VehicleSelectedPos == 1) {
1250         basicValue = 0.26;
1251     } else if (FuelSelectedPos == 2 && VehicleSelectedPos == 2) {
1252         basicValue = 0.30;
1253     } else if (FuelSelectedPos == 3 && VehicleSelectedPos == 0) {
1254         basicValue = 0.19;
1255     } else if (FuelSelectedPos == 3 && VehicleSelectedPos == 1) {
1256         basicValue = 0.24;
1257     } else if (FuelSelectedPos == 3 && VehicleSelectedPos == 2) {
1258         basicValue = 0.28;
1259     } else if (FuelSelectedPos == 4 && VehicleSelectedPos == 0) {
1260         basicValue = 0.07;
1261     } else if (FuelSelectedPos == 4 && VehicleSelectedPos == 1) {
1262         basicValue = 0.08;
1263     } else if (FuelSelectedPos == 4 && VehicleSelectedPos == 2) {
1264         basicValue = 0.10;
1265     }
1266
1267     if (distance.contains(" km")) {
1268         co2value = basicValue * Double.parseDouble(distance.replace(" km", ""));
1269     }
1270

```

```

1270 } else {
1271     co2value = basicValue * (Double.parseDouble(distance.replace(" m", "")) / 1000);
1272 }
1273 return co2value;
1274 }
1275 /**
1276 * Method for add record in firebase
1277 */
1278 private void AddTripRecord(String Source, String Destination, String Distance, String Co2, Dialog dialog, Boolean isCo2,
1279     String FuelType, String VehicleType, String rideType) {
1280
1281     String date = new SimpleDateFormat("dd-MM-yyyy", Locale.getDefault()).format(new Date());
1282     Map<String, Object> user = new HashMap<>();
1283     user.put("UserId", getShaedPref(HomeActivity.this, spUserId));
1284     user.put("Source", Source);
1285     user.put("Destination", Destination);
1286     user.put("Distance", Distance);
1287     user.put("Co2Type", isCo2);
1288     user.put("Co2", Co2);
1289     user.put("RideType", rideType);
1290     if (markerFlag.equals("CAR")) {
1291         user.put("VehicleType", VehicleType);
1292         user.put("FuelType", FuelType);
1293     } else {
1294         user.put("VehicleType", "");
1295         user.put("FuelType", "");
1296     }
1297     user.put("Date", date);
1298     FirebaseFirestore db = FirebaseFirestore.getInstance();
1299     db.collection("Co2-Report")
1300         .document("Co2-Report" + System.currentTimeMillis() / 1000).set(user)
1301         .addOnSuccessListener(aVoid -> {
1302             dialog.dismiss();
1303             startActivity(new Intent(getApplicationContext(), HomeActivity.class).setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK));
1304
1305         }).addOnFailureListener(e -> {
1306             Log.e(TAG, "registerDataToCloud: " + e.getMessage());
1307             progressDialog.dismiss();
1308         });
1309     });
1310 }
1311 /**
1312 * Method for animation camera on map
1313 */
1314 private void animateCamera(LatLng latLng) {
1315     CameraPosition cameraPosition = CameraPosition.builder().target(latLng).zoom(16.5f).build();
1316     googleMap.animateCamera(CameraUpdateFactory.newCameraPosition(cameraPosition));
1317 }
1318 /**
1319 * Method for get marker on basis of selection
1320 */
1321 private Marker addCarMarkerAndGet(LatLng latLng) {
1322
1323     if (markerFlag.equals("CAR")) {
1324         return googleMap.addMarker(new MarkerOptions().position(latLng).anchor(0.5f, 0.5f).flat(true).icon(BitmapDescriptorFactory.fromResource(R.mipmap.new_car_small)));
1325     } else if (markerFlag.equals("BUS")) {
1326         return googleMap.addMarker(new MarkerOptions().position(latLng).anchor(0.5f, 0.5f).flat(true).icon(BitmapDescriptorFactory.fromResource(R.mipmap.ic_bus)));
1327     } else if (markerFlag.equals("TRAIN")) {
1328         return googleMap.addMarker(new MarkerOptions().position(latLng).anchor(0.5f, 0.5f).flat(true).icon(BitmapDescriptorFactory.fromResource(R.mipmap.ic_train)));
1329     } else if (markerFlag.equals("FLIGHT")) {
1330         return googleMap.addMarker(new MarkerOptions().position(latLng).anchor(0.5f, 0.5f).flat(true).icon(BitmapDescriptorFactory.fromResource(R.mipmap.ic_air_plan)));
1331     } else if (markerFlag.equals("WALK")) {
1332         return googleMap.addMarker(new MarkerOptions().position(latLng).anchor(0.5f, 0.5f).flat(true).icon(BitmapDescriptorFactory.fromResource(R.mipmap.ic_man)));
1333     } else
1334         return googleMap.addMarker(new MarkerOptions().position(latLng).anchor(0.5f, 0.5f).flat(true).icon(BitmapDescriptorFactory.fromResource(R.mipmap.ic_bike)));
1335 }
1336 /**
1337 * Method for add source and destination marker
1338 */
1339 private Marker addOriginDestinationMarkerAndGet(LatLng latLng) {
1340     BitmapDescriptor bitmapDescriptor =
1341         BitmapDescriptorFactory.fromBitmap(MapUtils.getOriginDestinationMarkerBitmap());
1342     return googleMap.addMarker(
1343         new MarkerOptions().position(latLng).flat(true).icon(bitmapDescriptor));
1344 }
1345 /**
1346 * Method for Aynctask to get direction data
1347 */
1348 @SuppressLint("StaticFieldLeak")
1349 private class DownloadTask extends AsyncTask<String, Void, String> {
1350     @Override
1351     protected String doInBackground(String... url) {
1352         String data = "";
1353         try {
1354             data = downloadUrl(url[0]);
1355         } catch (Exception e) {
1356             Log.d("Background Task", e.toString());
1357         }
1358         return data;
1359     }
1360     @Override
1361     protected void onPostExecute(String result) {
1362         super.onPostExecute(result);
1363         ParserTask parserTask = new ParserTask();
1364         parserTask.execute(result);
1365     }
1366 }
1367 /**
1368 * Method for AsyncTask to parse direction data
1369 */
1370 private class ParserTask extends AsyncTask<String, Void, List<Direction>> {
1371     @Override
1372     protected List<Direction> doInBackground(String... url) {
1373         List<Direction> directionList = new ArrayList<>();
1374         try {
1375             JSONObject jsonObject = new JSONObject(result);
1376             JSONArray jsonArray = jsonObject.getJSONArray("routes");
1377             for (int i = 0; i < jsonArray.length(); i++) {
1378                 JSONObject jsonObject1 = jsonArray.getJSONObject(i);
1379                 JSONArray jsonArray1 = jsonObject1.getJSONArray("legs");
1380                 for (int j = 0; j < jsonArray1.length(); j++) {
1381                     JSONObject jsonObject2 = jsonArray1.getJSONObject(j);
1382                     JSONArray jsonArray2 = jsonObject2.getJSONArray("steps");
1383                     for (int k = 0; k < jsonArray2.length(); k++) {
1384                         JSONObject jsonObject3 = jsonArray2.getJSONObject(k);
1385                         Direction direction = new Direction();
1386                         direction.setStartAddress(jsonObject3.getString("start_address"));
1387                         direction.setEndAddress(jsonObject3.getString("end_address"));
1388                         direction.setDistance(jsonObject3.getString("distance"));
1389                         direction.setDuration(jsonObject3.getString("duration"));
1390                         direction.setStartLat(jsonObject3.getDouble("start_lat"));
1391                         direction.setStartLon(jsonObject3.getDouble("start_lon"));
1392                         direction.setEndLat(jsonObject3.getDouble("end_lat"));
1393                         direction.setEndLon(jsonObject3.getDouble("end_lon"));
1394                         direction.setHeadings(jsonObject3.getJSONArray("headings"));
1395                         direction.setPolyline(jsonObject3.getJSONObject("polyline"));
1396                         directionList.add(direction);
1397                     }
1398                 }
1399             }
1400         } catch (Exception e) {
1401             e.printStackTrace();
1402         }
1403         return directionList;
1404     }
1405     @Override
1406     protected void onPostExecute(List<Direction> directionList) {
1407         if (directionList != null) {
1408             for (Direction direction : directionList) {
1409                 LatLng startLatLng = new LatLng(direction.getStartLat(), direction.getStartLon());
1410                 LatLng endLatLng = new LatLng(direction.getEndLat(), direction.getEndLon());
1411                 PolylineOptions polylineOptions = new PolylineOptions()
1412                     .addAll(new ArrayList<LatLng>()
1413                         .add(startLatLng)
1414                         .add(endLatLng));
1415                 Polyline polyline = googleMap.addPolyline(polylineOptions);
1416             }
1417         }
1418     }
1419 }

```

```
1378 /**
1379 * Method for AsynTask to parse direction data
1380 */
1381 @SuppressLint("StaticFieldLeak")
1382 private class ParserTask extends AsyncTask<String, Integer, List<List<HashMap<String, String>>> {
1383     @Override
1384     protected List<List<HashMap<String, String>>> doInBackground(String... jsonData) {
1385         JSONObject jObject;
1386         List<List<HashMap<String, String>>> routes = null;
1387
1388         try {
1389             jObject = new JSONObject(jsonData[0]);
1390
1391             if (!jObject.getString("status").equals("ZERO_RESULTS")) {
1392                 DirectionsJSONParser parser = new DirectionsJSONParser();
1393                 Log.e("JsonObject", jObject.toString());
1394                 routes = parser.parse(jObject);
1395             } else {
1396                 new Thread() {
1397                     public void run() {
1398                         HomeActivity.this.runOnUiThread(new Runnable() {
1399                             public void run() {
1400                                 NoRouteDialog();
1401                             }
1402                         });
1403                     }.start();
1404                 }
1405             }
1406
1407         } catch (Exception e) {
1408             e.printStackTrace();
1409         }
1410         return routes;
1411     }
1412 }
1413
1414 @Override
1415 protected void onPostExecute(List<List<HashMap<String, String>>> result) {
1416     if (result != null && result.size() > 0) {
1417         polylineList = new ArrayList<>();
1418         instruct = new ArrayList<>();
1419         for (int i = 0; i < result.size(); i++) {
1420             List<HashMap<String, String>> path = result.get(i);
1421             for (int j = 0; j < path.size(); j++) {
1422                 HashMap<String, String> point = path.get(j);
1423                 double lat = Double.parseDouble(point.get("lat"));
1424                 double lng = Double.parseDouble(point.get("lng"));
1425                 LatLng position = new LatLng(lat, lng);
1426                 polylineList.add(position);
1427                 instruct.add(point.get("instructions"));
1428             }
1429         }
1430         staticPolyLine();
1431     }
1432 }
1433 }
```

```
package com.custom.co2.activity;

import android.Manifest;
import android.annotation.SuppressLint;
import android.content.Context;
import android.content.pm.PackageManager;
import android.content.res.Resources;
1 import android.graphics.Color;
2 import android.location.Address;
3 import android.location.Geocoder;
4 import android.location.Location;
5 import android.os.Build;
6 import android.os.Bundle;
7 import android.os.Handler;
8 import android.text.Editable;
9 import android.text.InputType;
10 import android.text.TextWatcher;
11 import android.transition.TransitionInflater;
12 import android.util.Log;
13 import android.view.View;
14 import android.view.animation.Animation;
15 import android.view.animation.AnimationUtils;
16 import android.view.inputmethod.InputMethodManager;
17 import android.widget.Button;
18 import android.widget.EditText;
19 import android.widget.FrameLayout;
20 import android.widget.ImageView;
21 import android.widget.LinearLayout;
22 import android.widget.Toast;
23
24 import androidx.annotation.NonNull;
25 import androidx.appcompat.app.AppCompatActivity;
26 import androidx.core.content.ContextCompat;
27 import androidx.recyclerview.widget.LinearLayoutManager;
28 import androidx.recyclerview.widget.RecyclerView;
29 import androidx.vectordrawable.graphics.drawable.AnimatedVectorDrawableCompat;
30
31 import com.custom.co2.R;
32 import com.custom.co2.adapter.PlaceAutocompleteAdapter;
33 import com.custom.co2.utils.CustomPickUp;
34 import com.custom.co2.utils.CustomProgress;
35 import com.custom.co2.utils.MapUtils;
36 import com.custom.co2.utils.Place;
37 import com.custom.co2.utils.ProgressListener;
38 import com.google.android.gms.common.ConnectionResult;
39 import com.google.android.gms.common.api.GoogleApiClient;
40 import com.google.android.gms.common.api.PendingResult;
41 import com.google.android.gms.common.api.ResultCallback;
42 import com.google.android.gms.location.LocationListener;
43 import com.google.android.gms.location.LocationRequest;
44 import com.google.android.gms.location.LocationServices;
45 import com.google.android.gms.location.places.AutocompleteFilter;
46 import com.google.android.gms.location.places.PlaceBuffer;
47 import com.google.android.gms.location.places.Places;
48 import com.google.android.gms.maps.CameraUpdateFactory;
49 import com.google.android.gms.maps.GoogleMap;
50 import com.google.android.gms.maps.OnMapReadyCallback;
51 import com.google.android.gms.maps.SupportMapFragment;
52 import com.google.android.gms.maps.model.BitmapDescriptorFactory;
53 import com.google.android.gms.maps.model.LatLng;
54 import com.google.android.gms.maps.model.LatLngBounds;
55 import com.google.android.gms.maps.model.MapStyleOptions;
56
57 import com.google.android.gms.maps.model.Marker;
58 import com.google.android.gms.maps.model.MarkerOptions;
59
60 import java.io.IOException;
61 import java.util.ArrayList;
62 import java.util.List;
63
64 public class PickLocation extends AppCompatActivity implements GoogleApiClient.OnConnectionFailedListener,
65 PlaceAutocompleteAdapter.PlaceAutoCompleteInterface, OnMapReadyCallback, LocationListener, ProgressListener {
66     private RecyclerView mRecyclerView;
67     private PlaceAutocompleteAdapter mAdapter;
68     private static final LatLngBounds BOUNDS_UK = new LatLngBounds(
69         new LatLng(6.4626999, 68.1097),
70         new LatLng(35.513327, 97.3953586999999));
71     EditText mSearchEditText;
72     LocationRequest mLocationRequest;
73     private static final long INTERVAL = 1000 * 5; //1 minute
74     private static final long FASTEST_INTERVAL = 1000 * 3;
75     private GoogleApiClient mClient;
76     private GoogleMap googleMap;
77     LatLng loc;
78     private Marker carMarker;
79     ImageView pin;
80     FrameLayout frame;
81     Button done;
82     ImageView iv_line;
83     LinearLayout linSetLocation;
84
85     private AnimatedVectorDrawableCompat avd;
86
87     /**
88      * Method for start text field animation
89     */
90
91
92
93
94
```

```

94
95     private void startAnimation(boolean isStart) {
96         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
97             if (isStart) {
98                 repeatAnimation();
99             } else {
100                 handler.postDelayed(new Runnable() {
101                     @Override
102                     public void run() {
103                         iv_line.removeCallbacks(action);
104                         avd.stop();
105                         iv_line.setVisibility(View.GONE);
106                         iv_line.invalidate();
107                     }
108                 }, 800);
109             }
110         }
111     }
112
113     private Runnable action = () -> repeatAnimation();
114     Handler handler = new Handler();
115
116     /**
117      * Method for repeat text field animation
118     */
119     private void repeatAnimation() {
120         if (avd != null) {
121             new Thread(() -> {
122                 handler.post(() -> {
123                     iv_line.setVisibility(View.VISIBLE);
124                     avd.start();
125                     iv_line.postDelayed(action, 1000);
126
127                     iv_line.invalidate();
128                 });
129             }).start();
130         }
131     }
132
133     /**
134      * Method for destroy activity
135     */
136     @Override
137     protected void onDestroy() {
138         super.onDestroy();
139     }
140
141     /**
142      * Method for initialize layout
143     */
144     @Override
145     protected void onCreate(Bundle savedInstanceState) {
146         super.onCreate(savedInstanceState);
147         getWindow().getDecorView().setSystemUiVisibility(
148             View.SYSTEM_UI_FLAG_LAYOUT_STABLE
149             | View.SYSTEM_UI_FLAG_LAYOUT_FULLSCREEN);
150         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
151             getWindow().setStatusBarColor(Color.TRANSPARENT);
152             getWindow().setSharedElementEnterTransition(TransitionInflater.from(this)
153                 .inflateTransition(R.transition.changebounds));
154         }
155         Bundle bundle = getIntent().getExtras();
156         if (bundle != null) {
157             loc = new LatLng(bundle.getDouble("lat"), bundle.getDouble("lon"));
158         }
159     }
160
161     mClient = new GoogleApiClient
162         .Builder(this)
163         .addApi(Places.GEO_DATA_API)
164         .addApi(Places.PLACE_DETECTION_API)
165         .addApi(LocationServices.API)
166         .build();
167     setContentView(R.layout.activity_pick);
168     findViewById(R.id.close).setOnClickListener(new View.OnClickListener() {
169         @Override
170         public void onClick(View v) {
171             onBackPressed();
172         }
173     });
174     CustomProgress.getInstance().setListener(this);
175     Animation animation = AnimationUtils.loadAnimation(this, android.R.anim.slide_in_left);
176     findViewById(R.id.close).startAnimation(animation);
177     pin = findViewById(R.id.pin);
178     linSetLocation = findViewById(R.id.lin_set_location);
179     frame = findViewById(R.id.frame);
180     done = findViewById(R.id.done);
181     iv_line = findViewById(R.id.iv_line);
182     mSearchEdittext = findViewById(R.id.mSearchText);
183
184     mSearchEdittext.setEnabled(false);
185     mRecyclerView = findViewById(R.id.list_search);
186     mRecyclerView.setHasFixedSize(true);
187     LinearLayoutManager llm = new LinearLayoutManager(this);
188     mRecyclerView.setLayoutManager(llm);
189     avd = AnimatedVectorDrawableCompat.create(this, R.drawable.avd_line);
190     iv_line.setBackground(avd);
191     iv_line.setVisibility(View.GONE);
192     final AutocompleteFilter typeFilter = new AutocompleteFilter.Builder()
193         .setCountry("IN")

```

```
195 .onUi();
196 mAdapter = new PlaceAutocompleteAdapter(this, mClient, BOUNDS_UK, typeFilter);
197 mAdapter.setProgress(iv_line);
198 MapUtils.slideUp(this, mRecyclerView, new Animation.AnimationListener() {
199     @Override
200     public void onAnimationStart(Animation animation) {
201
202     }
203
204     @Override
205     public void onAnimationEnd(Animation animation) {
206         mRecyclerView.setAdapter(mAdapter);
207         mSearchEdittext.setEnabled(true);
208     }
209
210     @Override
211     public void onAnimationRepeat(Animation animation) {
212
213     }
214 });
215
216
217 mSearchEdittext.setOnClickListener(new View.OnClickListener() {
218     @Override
219     public void onClick(View v) {
220         linSetLocation.setVisibility(View.VISIBLE);
221         if (mRecyclerView.getVisibility() == View.GONE) {
222             MapUtils.slideUp(PickLocation.this, mRecyclerView, new Animation.AnimationListener() {
223                 @Override
224                 public void onAnimationStart(Animation animation) {
225                     MapUtils.hideKeyboard(PickLocation.this);
226                     mSearchEdittext.setText(null);
227                     mSearchEdittext.setInputType(InputType.TYPE_CLASS_TEXT);
228                     mRecyclerView.clearAnimation();
229                     mRecyclerView.setVisibility(View.VISIBLE);
230                     frame.setVisibility(View.GONE);
231                 }
232
233                 @Override
234                 public void onAnimationEnd(Animation animation) {
235
236                 }
237
238                 @Override
239                 public void onAnimationRepeat(Animation animation) {
240
241
242                 }
243             });
244         } else {
245             mSearchEdittext.setFocusable(true);
246             mSearchEdittext.setCursorVisible(true);
247             mSearchEdittext.requestFocus();
248             mSearchEdittext.postDelayed(new Runnable() {
249                 @Override
250                 public void run() {
251                     InputMethodManager inputMethodManager = (InputMethodManager) getSystemService(Context.INPUT_METHOD_SERVICE);
252                     if (inputMethodManager != null) {
253                         inputMethodManager.showSoftInput(mSearchEdittext, InputMethodManager.SHOW_IMPLICIT);
254                     }
255                     mSearchEdittext.setFocusable(true);
256                     mSearchEdittext.setCursorVisible(true);
257                     mSearchEdittext.requestFocus();
258                 }
259             }, 200);
260         }
261     }
262 });
263
264
265 mSearchEdittext.addTextChangedListener(new TextWatcher() {
266     @Override
267     public void beforeTextChanged(CharSequence s, int start, int count, int after) {
268
269     }
270
271     @Override
272     public void onTextChanged(CharSequence s, int start, int before, int count) {
273         if (count > 0) {
274             if (mAdapter != null) {
275                 mRecyclerView.setAdapter(mAdapter);
276             }
277         } else {
278             if (mAdapter != null) {
279                 mAdapter.clearList();
280                 mAdapter.notifyDataSetChanged();
281                 mRecyclerView.setAdapter(mAdapter);
282             }
283         }
284     }
285     if (!s.toString().equals("") && mClient.isConnected()) {
286         mAdapter.getFilter().filter(s.toString());
287     }
288 }
289
290     @Override
291     public void afterTextChanged(Editable s) {
292
293     }
294 });
295
296 }
```

```
297     linSetLocation.setOnClickListener(new View.OnClickListener() {
298         @Override
299         public void onClick(View view) {
300             linSetLocation.setVisibility(View.GONE);
301             MapUtils.hideKeyboard(PickLocation.this);
302             MapUtils.SlideToAbove(mRecyclerView, new Animation.AnimationListener() {
303                 @Override
304                 public void onAnimationStart(Animation animation) {
305
306                 }
307
308                 @Override
309                 public void onAnimationEnd(Animation animation) {
310                     mRecyclerView.clearAnimation();
311                     mRecyclerView.setVisibility(View.GONE);
312                     frame.setVisibility(View.VISIBLE);
313
314                     mSearchEdittext.setInputType(InputType.TYPE_NULL);
315                     final SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.map);
316                     mapFragment.getMapAsync(PickLocation.this);
317                 }
318
319                 @Override
320                 public void onAnimationRepeat(Animation animation) {
321
322                     }
323                 });
324             });
325         });
326     }
327 }
328 /**
329 * Method for hand back press
330 */
331 private void onBack() {
332     MapUtils.SlideToAbove(mRecyclerView, new Animation.AnimationListener() {
333         @Override
334         public void onAnimationStart(Animation animation) {
335
336             }
337
338             @Override
339             public void onAnimationEnd(Animation anim) {
340                 mRecyclerView.clearAnimation();
341                 mRecyclerView.setVisibility(View.GONE);
342                 Animation animation = AnimationUtils.loadAnimation(PickLocation.this, R.anim.slide_out_left);
343                 animation.setAnimationListener(new Animation.AnimationListener() {
344                     @Override
345                     public void onAnimationStart(Animation animation) {
346
347                         }
348
349                         @Override
350                         public void onAnimationEnd(Animation animation) {
351                             findViewById(R.id.close).setVisibility(View.GONE);
352                             PickLocation.super.onBackPressed();
353                         }
354
355                         @Override
356                         public void onAnimationRepeat(Animation animation) {
357
358                             }
359                         });
360                     findViewById(R.id.close).startAnimation(animation);
361                 }
362             }
363
364             @Override
365             public void onAnimationRepeat(Animation animation) {
366
367             }
368         });
369     }
370 }
371 /**
372 * Method for create location request
373 */
374 protected void createLocationRequest() {
375     mLocationRequest = new LocationRequest();
376     mLocationRequest.setInterval(INTERVAL);
377     mLocationRequest.setFastestInterval(FASTEST_INTERVAL);
378     mLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
379     if (ContextCompat.checkSelfPermission(this,
380             Manifest.permission.ACCESS_FINE_LOCATION)
381             == PackageManager.PERMISSION_GRANTED) {
382         if (mClient.isConnected()) {
383             LocationServices.FusedLocationApi.requestLocationUpdates(mClient, mLocationRequest, this);
384         }
385     }
386
387 }
388 }
389 /**
390 * Method for do stuffs while activity starts
391 */
392 @Override
393 protected void onStart() {
394     super.onStart();
395     mClient.connect();
396 }
```

```

398
399 /**
400 * Method for manage activity while in background for long time
401 */
402 @Override
403 protected void onStop() {
404     mClient.disconnect();
405     super.onStop();
406 }
407
408 /**
409 * Method for get callback when connection failed
410 */
411 @Override
412 public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {
413 }
414
415 /**
416 * Method for get place data when click on location in auto complete list
417 */
418 @Override
419 public void onPlaceClick(ArrayList<PlaceAutocompleteAdapter.PlaceAutocomplete> mResultList, int position) {
420     onProgressChanged(false);
421     if (mResultList != null) {
422         try {
423             final String placeld = String.valueOf(mResultList.get(position).placeld);
424             PendingResult<PlaceBuffer> placeResult = Places.GeoDataApi
425                 .getPlaceById(mClient, placeld);
426
427             placeResult.setResultCallback(new ResultCallback<PlaceBuffer>() {
428                 @Override
429                 public void onResult(@NonNull PlaceBuffer places) {
430                     Place place = new Place(getApplicationContext().PickLocation.this, String.valueOf(mResultList.get(position).description)), new LatLng(0, 0));
431                     mSearchEdittext.setText(String.valueOf(mResultList.get(position).description));
432                     srcAdd = String.valueOf(String.valueOf(mResultList.get(position).description));
433                     place.setSrcAddress(String.valueOf(mResultList.get(position).description));
434                     CustomPickUp.getInstance().changeState(place);
435                     onBack();
436
437                 }
438             });
439         } catch (Exception e) {
440             e.printStackTrace();
441         }
442     }
443 }
444
445 } else {
446     MapUtils.hideKeyboard(PickLocation.this);
447     MapUtils.SlideToAbove(mRecyclerView, new Animation.AnimationListener() {
448         @Override
449         public void onAnimationStart(Animation animation) {
450
451     }
452
453         @Override
454         public void onAnimationEnd(Animation animation) {
455             mRecyclerView.clearAnimation();
456             mRecyclerView.setVisibility(View.GONE);
457             frame.setVisibility(View.VISIBLE);
458             mSearchEdittext.setInputType(InputType.TYPE_NULL);
459             final SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
460                 .findFragmentById(R.id.map);
461             if (mapFragment != null) {
462                 mapFragment.getMapAsync(PickLocation.this);
463             }
464         }
465
466         @Override
467         public void onAnimationRepeat(Animation animation) {
468
469         }
470     });
471 }
472 }
473
474 /**
475 * Method for get lat lng from address
476 */
477 public LatLng getLocationFromAddress(Context context, String strAddress) {
478     Geocoder coder = new Geocoder(context);
479     List<Address> address;
480     LatLng p1 = null;
481     try {
482         address = coder.getFromLocationName(strAddress, 5);
483         if (address == null) {
484             return null;
485         }
486         Address location = address.get(0);
487         p1 = new LatLng(location.getLatitude(), location.getLongitude());
488     } catch (IOException ex) {
489         ex.printStackTrace();
490     }
491     return p1;
492 }
493
494
495 String srcAdd = "";
496
497 /**
498 * Method for get callback when map is ready
499 */

```

```

499 /**
500 * @Override
501 public void onMapReady(final GoogleMap googleMap) {
502     this.googleMap = googleMap;
503     this.googleMap.getUiSettings().setCompassEnabled(false);
504     this.googleMap.getUiSettings().setMapToolbarEnabled(false);
505     try {
506         boolean success = googleMap.setMapStyle(
507             MapStyleOptions.loadRawResourceStyle(
508                 this, R.raw.custom_map));
509     } catch (Resources.NotFoundException e) {
510         Log.e("error", "Can't find style. Error: ", e);
511     }
512 }
513 createLocationRequest();
514 googleMap.setOnCameraMoveListener(new GoogleMap.OnCameraMoveListener() {
515     @SuppressLint("SetTextI18n")
516     @Override
517     public void onCameraMove() {
518         done.setEnabled(false);
519         done.setText("Fetching...");
520     }
521 });
522 googleMap.setOnCameraIdleListener(new GoogleMap.OnCameraIdleListener() {
523     @SuppressLint("SetTextI18n")
524     @Override
525     public void onCameraIdle() {
526         LatLng ladw = googleMap.getCameraPosition().target;
527         try {
528             Geocoder selected_place_geocoder = new Geocoder(PickLocation.this);
529             List<Address> address = new ArrayList<>();
530             address = selected_place_geocoder.getFromLocation(ladw.latitude, ladw.longitude, 1);
531             Address location = address.get(0);
532             if (location != null) {
533                 if (location.getLocality() != null) {
534                     srcAdd = location.getAddressLine(0) + "," + location.getLocality() + "," + location.getCountryName();
535                     mSearchEdittext.setText(srcAdd);
536                 } else {
537                     srcAdd = location.getAddressLine(0) + "," + location.getCountryName();
538                     mSearchEdittext.setText(srcAdd);
539                 }
540             }
541         } catch (Exception e) {
542             e.printStackTrace();
543         }
544         done.setText("Done");
545         done.setEnabled(true);
546     }
547 });
548 done.setOnClickListener(v -> {
549     if (done.getText().toString().equalsIgnoreCase("Done")) {
550         Place place = new Place(new LatLng(googleMap.getCameraPosition().target.latitude, googleMap.getCameraPosition().target.longitude), googleMap.getCameraPosition().target);
551         place.setSrcAddress(srcAdd);
552         CustomPickUp.getInstance().changeState(place);
553         onBack();
554     } else {
555         Toast.makeText(getApplicationContext(), "Wait for Location to be fetched.", Toast.LENGTH_SHORT).show();
556     }
557 });
558 }
559 */
560 /**
561 * Method for get callback when location changed
562 */
563 @Override
564 public void onLocationChanged(Location location) {
565     if (carMarker == null) {
566         carMarker = googleMap.addMarker(new MarkerOptions().position(new LatLng(location.getLatitude(), location.getLongitude())));
567         flat(true).icon(BitmapDescriptorFactory.fromResource(R.mipmap.source));
568         googleMap.setOnMapLoadedCallback(() -> googleMap.animateCamera(CameraUpdateFactory.newLatLngZoom(carMarker.getPosition(), 18), new GoogleMap.CancelableCallback() {
569             @Override
570             public void onFinish() {
571                 pin.setVisibility(View.VISIBLE);
572                 done.setVisibility(View.VISIBLE);
573             }
574         });
575     }
576     @Override
577     public void onCancel() {
578         pin.setVisibility(View.VISIBLE);
579         done.setVisibility(View.VISIBLE);
580     }
581 });
582 } else {
583     carMarker.setPosition(new LatLng(location.getLatitude(), location.getLongitude()));
584 }
585 }
586 */
587 /**
588 * Method for get callback when progress changed
589 */
590 @Override
591 public void onProgressChanged(boolean state) {
592     startAnimation(state);
593 }
594 }

```

```
1 package com.custom.co2.activity;
2
3 import android.app.DatePickerDialog;
4 import android.app.Dialog;
5 import android.content.Context;
6 import android.content.Intent;
7 import android.net.ConnectivityManager;
8 import android.os.Bundle;
9 import android.util.Log;
10 import android.util.Patterns;
11 import android.widget.DatePicker;
12 import android.widget.EditText;
13 import android.widget.TextView;
14 import android.widget.Toast;
15
16 import androidx.annotation.NonNull;
17 import androidx.appcompat.app.AppCompatActivity;
18
19 import com.custom.co2.R;
20 import com.custom.co2.utils.Constant;
21 import com.google.android.gms.tasks.OnCompleteListener;
22 import com.google.android.gms.tasks.Task;
23 import com.google.firebase.firestore.DocumentSnapshot;
24 import com.google.firebase.firestore.FirebaseFirestore;
25 import com.google.firebase.firestore.QuerySnapshot;
26
27 import java.text.SimpleDateFormat;
28 import java.util.Calendar;
29 import java.util.HashMap;
30 import java.util.Locale;
31 import java.util.Map;
32
33 import static com.custom.co2.utils.Constant.showProgressDialog;
34
35 public class RegisterActivity extends AppCompatActivity {
36
37     TextView btnGotoSignUp;
38     FirebaseFirestore db;
39     String TAG = "RegisterActivity";
40     TextView btnLogin;
41     EditText edUsername, edEmail, edContact, edDob, edPassword, edConfirmPassword;
42     final Calendar myCalendar = Calendar.getInstance();
43     Dialog progressDialog;
44
45     /**
46      * Method for initailize layout
47      */
48     @Override
49     protected void onCreate(Bundle savedInstanceState) {
50         super.onCreate(savedInstanceState);
51         getSupportActionBar().hide();
52         setContentView(R.layout.activity_register);
53
54         inits();
55         clicks();
56
57         db = FirebaseFirestore.getInstance();
58         progressDialog = showProgressDialog(RegisterActivity.this);
59     }
60
61     /**
62      * Method for manage clicks
63      */
64     private void clicks() {
```

```
64 btnGoToSignUp.setOnClickListener(view -> {
65     startActivity(new Intent(getApplicationContext(), LoginActivity.class));
66     finish();
67 });
68 btnLogin.setOnClickListener(view -> {
69     if(!isNetworkConnected()){
70         Toast.makeText(this, "Internet is not connected", Toast.LENGTH_SHORT).show();
71         return;
72     }
73     if (Valid()) {
74         checkEmail(edEmail.getText().toString());
75     }
76 });
77 edDob.setOnClickListener(view -> {
78
79     myCalendar.add(Calendar.YEAR, -10);
80     int year = myCalendar.get(Calendar.YEAR);
81     int month = myCalendar.get(Calendar.MONTH);
82     int day = myCalendar.get(Calendar.DAY_OF_MONTH);
83
84     DatePickerDialog datePicker = new DatePickerDialog(RegisterActivity.this, date, year, month, day);
85
86     datePicker.getDatePicker().setMaxDate(myCalendar.getTimeInMillis());
87     datePicker.show();
88
89
90 });
91 }
92
93 /**
94 * Method for validations
95 */
96 private boolean Valid() {
97     if (edUsername.getText().toString().isEmpty()) {
98         edUsername.setError("Enter username");
99         edUsername.requestFocus();
100        return false;
101    }
102
103 } else if (edContact.getText().toString().isEmpty()) {
104     edContact.setError("Enter contact");
105     edContact.requestFocus();
106     return false;
107
108 } else if (edContact.getText().toString().length() != 10) {
109     edContact.setError("Enter valid contact");
110     edContact.requestFocus();
111     return false;
112
113 } else if (edDob.getText().toString().isEmpty()) {
114     edDob.setError("Enter date of birth");
115     edDob.requestFocus();
116     return false;
117
118 } else if (edEmail.getText().toString().isEmpty()) {
119     edEmail.setError("Enter email");
120     edEmail.requestFocus();
121     return false;
122
123 } else if (!Patterns.EMAIL_ADDRESS.matcher(edEmail.getText().toString()).matches()) {
124     edEmail.setError("Enter valid email");
125     edEmail.requestFocus();
126     return false;
127
128 } else if (edPassword.getText().toString().isEmpty()) {
129     edPassword.setError("Enter password");
130     edPassword.requestFocus();
```

```

131     return false;
132
133 }
134 else if (edPassword.getText().toString().length() < 6) {
135     edPassword.setError("Password length should be more then 5 letter");
136     edPassword.requestFocus();
137     return false;
138
139 }
140 else if (!edConfirmPassword.getText().toString().equals(edPassword.getText().toString())) {
141     edConfirmPassword.setError("Password don't match");
142     edConfirmPassword.requestFocus();
143     return false;
144
145 }
146
147
148     return true;
149 }
150
151 /**
152 * Method for initialize view and variables
153 */
154 private void init() {
155     btnGoSignUp = findViewById(R.id.tv_goto_login);
156     btnLogin = findViewById(R.id.tv_btn_login);
157     edUsername = findViewById(R.id.ed_username);
158     edConfirmPassword = findViewById(R.id.ed_confirm_password);
159     edEmail = findViewById(R.id.ed_email);
160     edPassword = findViewById(R.id.ed_password);
161     edDob = findViewById(R.id.ed_dob);
162     edContact = findViewById(R.id.ed_contact);
163
164 }
165
166 /**
167 * Method for get date picker callback when date selected
168 */
169 DatePickerDialog.OnDateSetListener date = (view, year, monthOfYear, dayOfMonth) -> {
170     myCalendar.set(Calendar.YEAR, year);
171     myCalendar.set(Calendar.MONTH, monthOfYear);
172     myCalendar.set(Calendar.DAY_OF_MONTH, dayOfMonth);
173     updateLabel();
174 };
175
176 /**
177 * Method for update date label
178 */
179 private void updateLabel() {
180     String myFormat = "dd/MM/yyyy";
181     SimpleDateFormat sdf = new SimpleDateFormat(myFormat, Locale.US);
182
183     edDob.setText(sdf.format(myCalendar.getTime()));
184     edDob.setError(null);
185 }
186
187 /**
188 * Method for register data to firebase
189 */
190 private void registerDataToCloud(String username, String email, String password, String dob, String contact) {
191     Map<String, Object> user = new HashMap<>();
192     user.put("Username", username);
193     user.put("Email", email);
194     user.put("Password", password);
195     user.put("DOB", dob);
196     user.put("Contact", contact);
197 }
```

```

198
199     db.collection("users")
200         .document("user" + System.currentTimeMillis() / 1000).set(user)
201         .addOnSuccessListener(aVoid -> {
202             Toast.makeText(RegisterActivity.this, "Registration successfully", Toast.LENGTH_SHORT).show();
203             progressDialog.dismiss();
204             startActivity(new Intent(getApplicationContext(), LoginActivity.class));
205             finish();
206
207         }) .addOnFailureListener(e -> {
208             Log.e(TAG, "registerDataToCloud: " + e.getMessage());
209             progressDialog.dismiss();
210         });
211     }
212
213 /**
214 * Method for check email existing
215 */
216 private void checkEmail(String email) {
217     progressDialog.show();
218
219     db.collection("users")
220         .whereEqualTo("Email", email)
221         .get()
222         .addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {
223             @Override
224             public void onComplete(@NonNull Task<QuerySnapshot> task) {
225                 Log.d(TAG, "checkingIfusernameExist: checking if username exists");
226
227                 if (task.isSuccessful()) {
228                     for (DocumentSnapshot ds : task.getResult()) {
229                         if (ds.getString("Email").equals(email)) {
230                             progressDialog.dismiss();
231                             Constant.showAlertDialog(RegisterActivity.this, "This email is already exist");
232
233                         }
234                     }
235                 }
236             //checking if task contains any payload. if no, then update
237             if (task.getResult().size() == 0) {
238                 try {
239                     registerDataToCloud(edUsername.getText().toString(), edEmail.getText().toString(),
240                         edPassword.getText().toString(), edDob.getText().toString(), edContact.getText().toString());
241                 } catch (NullPointerException e) {
242                     Log.e(TAG, "NullPointerException: " + e.getMessage());
243                 }
244             }
245         });
246     });
247 }
248
249 }
250
251 private boolean isNetworkConnected() {
252     ConnectivityManager cm = (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
253     return cm.getActiveNetworkInfo() != null && cm.getActiveNetworkInfo().isConnected();
254 }
255
}

```

```
package com.custom.co2.activity;

import android.Manifest;
import android.annotation.SuppressLint;
import android.content.Context;
import android.content.Intent;
import android.content.pm.PackageManager;
1 import android.content.res.Resources;
2 import android.graphics.Color;
3 import android.location.Address;
4 import android.location.Geocoder;
5 import android.location.Location;
6 import android.os.Build;
7 import android.os.Bundle;
8 import android.os.Handler;
9 import android.text.Editable;
10 import android.text.InputType;
11 import android.text.TextWatcher;
12 import android.transition.TransitionInflater;
13 import android.util.Log;
14 import android.view.View;
15 import android.view.animation.Animation;
16 import android.view.animation.AnimationUtils;
17 import android.view.inputmethod.InputMethodManager;
18 import android.widget.Button;
19 import android.widget.EditText;
20 import android.widget.FrameLayout;
21 import android.widget.ImageView;
22 import android.widget.LinearLayout;
23 import android.widget.RelativeLayout;
24 import android.widget.Toast;
25
26
27 import androidx.annotation.NonNull;
28 import androidx.appcompat.app.AppCompatActivity;
29 import androidx.core.app.ActivityCompat;
30 import androidx.core.app.ActivityOptionsCompat;
31 import androidx.core.content.ContextCompat;
32 import androidx.recyclerview.widget.LinearLayoutManager;
33 import androidx.recyclerview.widget.RecyclerView;
34 import androidx.vectordrawable.graphics.drawable.AnimatedVectorDrawableCompat;
35
36
37 import com.custom.co2.R;
38 import com.custom.co2.adapter.PlaceAutocompleteAdapter;
39 import com.custom.co2.utils.CustomModel;
40 import com.custom.co2.utils.CustomPickUp;
41 import com.custom.co2.utils.CustomProgress;
42 import com.custom.co2.utils.MapUtils;
43 import com.custom.co2.utils.Place;
44 import com.custom.co2.utils.PlaceListener;
45 import com.custom.co2.utils.ProgressListener;
46 import com.google.android.gms.common.ConnectionResult;
47 import com.google.android.gms.common.api.GoogleApiClient;
48 import com.google.android.gms.common.api.PendingResult;
49 import com.google.android.gms.common.api.ResultCallback;
50 import com.google.android.gms.location.LocationListener;
51 import com.google.android.gms.location.LocationRequest;
52 import com.google.android.gms.location.LocationServices;
53 import com.google.android.gms.location.places.AutocompleteFilter;
54 import com.google.android.gms.location.places.PlaceBuffer;
55 import com.google.android.gms.location.places.Places;
56 import com.google.android.gms.maps.CameraUpdateFactory;
57 import com.google.android.gms.maps.GoogleMap;
58
59
60 import com.google.android.gms.maps.OnMapReadyCallback;
61 import com.google.android.gms.maps.SupportMapFragment;
62 import com.google.android.gms.maps.model.BitmapDescriptorFactory;
63 import com.google.android.gms.maps.model.LatLng;
64 import com.google.android.gms.maps.model.LatLngBounds;
65 import com.google.android.gms.maps.model.MapStyleOptions;
66 import com.google.android.gms.maps.model.Marker;
67 import com.google.android.gms.maps.model.MarkerOptions;
68
69 import java.io.IOException;
70 import java.util.ArrayList;
71 import java.util.List;
72
73
74 public class SearchActivity extends AppCompatActivity implements GoogleApiClient.OnConnectionFailedListener,
75     PlaceAutocompleteAdapter.PlaceAutoCompleteInterface, OnMapReadyCallback, LocationListener, PlaceListener,
76     ProgressListener {
77     private RecyclerView mRecycler View;
78     PlaceAutocompleteAdapter mAdapter;
79     private static final LatLngBounds BOUNDS_INDIA = new LatLngBounds(
80         new LatLng(6.4626999, 68.1097),
81         new LatLng(35.513327, 97.3953586999999));
82     EditText mSearchEditText;
83     LocationRequest mLocationRequest;
84     private static final long INTERVAL = 1000 * 5; //1 minute
85     private static final long FASTEST_INTERVAL = 1000 * 3;
86     private GoogleApiClient mClient;
87     private GoogleMap googleMap;
88     LatLng loc;
89     private Marker carMarker;
90     ImageView pin;
91     FrameLayout frame;
92     Button done;
93     Place placeCust;
94     String srcAdd = "";
95     Button current;
96     ImageView iv_line;
97     LinearLayout linSetLocation;
98     SupportMapFragment mapFragment;
99 }
```

```
100 Handler handler = new Handler();
101 private AnimatedVectorDrawableCompat avd;
102
103 /**
104 * Method for callback of place selected
105 */
106 @Override
107 public void onPlace(Place place) {
108     this.placeCust = place;
109     if (current != null) {
110         current.setText(placeCust.getSrcAddress());
111     }
112 }
113
114 /**
115 * Method for callback when activity resumed
116 */
117 @Override
118 protected void onResume() {
119     super.onResume();
120 }
121
122 /**
123 * Method for start animation of view
124 */
125
126
127 private void startAnimation(boolean isStart) {
128     if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
129         if (isStart) {
130             repeatAnimation();
131         } else {
132             handler.postDelayed(new Runnable() {
133                 @Override
134                 public void run() {
135                     iv_line.removeCallbacks(action);
136                     avd.stop();
137                     iv_line.setVisibility(View.GONE);
138                     iv_line.invalidate();
139                 }
140             }, 800);
141         }
142     }
143 }
144
145
146 private Runnable action = () -> repeatAnimation();
147
148 /**
149 * Method for repeat animation
150 */
151 private void repeatAnimation() {
152     if (avd != null) {
153         new Thread(new Runnable() {
154             @Override
155             public void run() {
156                 handler.post(new Runnable() {
157                     @Override
158                     public void run() {
159                         iv_line.setVisibility(View.VISIBLE);
160                         avd.start();
161                         iv_line.postDelayed(action, 1000);
162                         iv_line.invalidate();
163                     }
164                 });
165             }
166         }).start();
167     }
168 }
169
170 /**
171 * Method for destroy activity
172 */
173
174 @Override
175 protected void onDestroy() {
176     super.onDestroy();
177 }
178
179 /**
180 * Method for initialize layout
181 */
182
183 @Override
184 protected void onCreate(Bundle savedInstanceState) {
185     super.onCreate(savedInstanceState);
186     getSupportActionBar().hide();
187     getWindow().getDecorView().setSystemUiVisibility(
188         View.SYSTEM_UI_FLAG_LAYOUT_STABLE
189         | View.SYSTEM_UI_FLAG_LAYOUT_FULLSCREEN);
190     if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
191         getWindow().setStatusBarColor(Color.TRANSPARENT);
192         getWindow().setSharedElementEnterTransition(TransitionInflater.from(this)
193             .inflateTransition(R.transition.changebounds));
194     }
195     Bundle bundle = getIntent().getExtras();
196     if (bundle != null) {
197         loc = new LatLng(bundle.getDouble("lat"), bundle.getDouble("lon"));
198     }
199     mClient = new GoogleApiClient
200         .Builder(this)
201         .addApi(Places.GEO_DATA_API)
202         .addApi(Places.PLACE_DETECTION_API)
203         .addApi(LocationServices.API)
204         .build();
205     setContentView(R.layout.activity_search);
206
207 }
```

```

208
209     CustomPickUp.getInstance().setListener(this);
210     CustomProgress.getInstance().setListener(this);
211     findViewById(R.id.close).setOnClickListener(new View.OnClickListener() {
212         @Override
213         public void onClick(View v) {
214             onBack();
215         }
216     });
217     current = findViewById(R.id.current);
218     linSetLocation = findViewById(R.id.lin_set_location);
219     iv_line = findViewById(R.id.iv_line);
220     current.setOnClickListener(new View.OnClickListener() {
221         @Override
222         public void onClick(View v) {
223             Intent intent = new Intent(SearchActivity.this, PickLocation.class);
224             intent.putExtra("lat", loc.latitude);
225             intent.putExtra("lon", loc.longitude);
226             if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
227                 current.setTransitionName("current");
228                 ActivityOptionsCompat optionsCompat = ActivityOptionsCompat.makeSceneTransitionAnimation(SearchActivity.this, current, current.getTransitionName());
229                 startActivity(intent, optionsCompat.toBundle());
230             } else
231                 startActivity(intent);
232         }
233     });
234     Animation animation = AnimationUtils.loadAnimation(this, android.R.anim.slide_in_left);
235     findViewById(R.id.close).startAnimation(animation);
236     pin = findViewById(R.id.pin);
237     frame = findViewById(R.id.frame);
238     done = findViewById(R.id.done);
239
240     mSearchEdittext = findViewById(R.id.mSearchText);
241
242     mSearchEdittext.setEnabled(false);
243     mRecyclerView = findViewById(R.id.list_search);
244     mRecyclerView.setHasFixedSize(true);
245     LinearLayoutManager llm = new LinearLayoutManager(this);
246     mRecyclerView.setLayoutManager(llm);
247     avd = AnimatedVectorDrawableCompat.create(this, R.drawable.avd_line);
248     iv_line.setBackground(avd);
249     iv_line.setVisibility(View.GONE);
250
251     mRecyclerView.setAdapter(mAdapter);
252     final AutocompleteFilter typeFilter = new AutocompleteFilter.Builder()
253         .setCountry("IN")
254         .build();
255     mAdapter = new PlaceAutocompleteAdapter(this, mClient, BOUNDS_INDIA, typeFilter);
256     mAdapter.setProgress(iv_line);
257
258
259
260     MapUtils.slideUp(this, mRecyclerView, new Animation.AnimationListener() {
261         @Override
262         public void onAnimationStart(Animation animation) {
263
264         }
265
266         @Override
267         public void onAnimationEnd(Animation animation) {
268             mRecyclerView.setAdapter(mAdapter);
269             mSearchEdittext.setEnabled(true);
270         }
271
272         @Override
273         public void onAnimationRepeat(Animation animation) {
274
275         }
276     });
277     mSearchEdittext.setOnClickListener(new View.OnClickListener() {
278         @Override
279         public void onClick(View v) {
280             linSetLocation.setVisibility(View.VISIBLE);
281             if (mRecyclerView.getVisibility() == View.GONE) {
282                 MapUtils.slideUp(SearchActivity.this, mRecyclerView, new Animation.AnimationListener() {
283                     @Override
284                     public void onAnimationStart(Animation animation) {
285                         MapUtils.hideKeyboard(SearchActivity.this);
286                         mSearchEdittext.setText(null);
287                         mSearchEdittext.setInputType(InputType.TYPE_CLASS_TEXT);
288                         mRecyclerView.clearAnimation();
289                         mRecyclerView.setVisibility(View.VISIBLE);
290                         frame.setVisibility(View.GONE);
291                     }
292
293                     @Override
294                     public void onAnimationEnd(Animation animation) {
295
296                     }
297
298                     @Override
299                     public void onAnimationRepeat(Animation animation) {
300
301                     }
302                 });
303             } else {
304                 mSearchEdittext.setFocusable(true);
305                 mSearchEdittext.setCursorVisible(true);
306                 mSearchEdittext.requestFocus();
307                 mSearchEdittext.postDelayed(new Runnable() {
308                     @Override
309                     public void run() {
310                         InputMethodManager inputMethodManager = (InputMethodManager) getSystemService(Context.INPUT_METHOD_SERVICE);
311                         if (inputMethodManager != null)
312                             inputMethodManager.showSoftInput(mSearchEdittext, InputMethodManager.SHOW_IMPLICIT);
313                     }
314                 });
315             }
316         }
317     });

```

```

315         mSearchEdittext.setFocusable(true);
316         mSearchEdittext.setCursorVisible(true);
317         mSearchEdittext.requestFocus();
318     }
319 }, 200);
320 }
321 });
322 });
323 LatLng ladw = new LatLng(loc.latitude, loc.longitude);
324 try {
325     Geocoder selected_place_geocoder = new Geocoder(SearchActivity.this);
326     List<Address> address;
327     address = selected_place_geocoder.getFromLocation(ladw.latitude, ladw.longitude, 1);
328     Address location = address.get(0);
329     if (location != null) {
330         if (location.getLocality() != null) {
331             srcAdd = location.getAddressLine(0) + "," + location.getLocality() + "," + location.getCountryName();
332         } else {
333             srcAdd = location.getAddressLine(0) + "," + location.getCountryName();
334         }
335     }
336 } catch (Exception e) {
337     e.printStackTrace();
338 }
339
340 mSearchEdittext.addTextChangedListener(new TextWatcher() {
341     @Override
342     public void beforeTextChanged(CharSequence s, int start, int count, int after) {
343     }
344
345     @Override
346     public void onTextChanged(CharSequence s, int start, int before, int count) {
347         if (count > 0) {
348             if (mAdapter != null) {
349                 mRecyclerView.setAdapter(mAdapter);
350                 Log.e("mAdapterSize", mAdapter.getItemCount() + "");
351             }
352         } else {
353             if (mAdapter != null) {
354                 mAdapter.clearList();
355                 mAdapter.notifyDataSetChanged();
356                 mRecyclerView.setAdapter(mAdapter);
357                 Log.e("mAdapterSize1", mAdapter.getItemCount() + "");
358             }
359         }
360         if (!s.toString().equals("") && mClient.isConnected()) {
361             mAdapter.getFilter().filter(s.toString());
362         }
363     }
364 }
365
366     @Override
367     public void afterTextChanged(Editable s) {
368
369     }
370 });
371
372 linSetLocation.setOnClickListener(new View.OnClickListener() {
373     @Override
374     public void onClick(View view) {
375         linSetLocation.setVisibility(View.GONE);
376         MapUtils.hideKeyboard(SearchActivity.this);
377         MapUtils.SlideToAbove(mRecyclerView, new Animation.AnimationListener() {
378             @Override
379             public void onAnimationStart(Animation animation) {
380
381
382             @Override
383             public void onAnimationEnd(Animation animation) {
384                 mRecyclerView.clearAnimation();
385                 mRecyclerView.setVisibility(View.VISIBLE);
386                 frame.setVisibility(View.VISIBLE);
387                 mSearchEdittext.setInputType(InputType.TYPE_NULL);
388                 mapFragment = (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.map);
389                 mapFragment.getMapAsync(SearchActivity.this);
390             }
391
392             @Override
393             public void onAnimationRepeat(Animation animation) {
394
395
396             });
397         });
398     }
399     com.google.android.libraries.places.api.Places.initialize(SearchActivity.this, getResources().getString(R.string.Map_Api_Key));
400 }
401
402 /**
403 * Method for handle on back press
404 */
405
406 private void onBack() {
407     MapUtils.SlideToAbove(mRecyclerView, new Animation.AnimationListener() {
408         @Override
409         public void onAnimationStart(Animation animation) {
410             MapUtils.hideKeyboard(SearchActivity.this);
411         }
412
413         @Override
414         public void onAnimationEnd(Animation anim) {
415             mRecyclerView.clearAnimation();
416             mRecyclerView.setVisibility(View.GONE);
417             Animation animation = AnimationUtils.loadAnimation(SearchActivity.this, R.anim.slide_out_left);
418             animation.setAnimationListener(new Animation.AnimationListener() {
419                 @Override
420                 public void onAnimationStart(Animation animation) {
421

```

```

422 }
423
424     @Override
425     public void onAnimationEnd(Animation animation) {
426         findViewById(R.id.close).setVisibility(View.GONE);
427         SearchActivity.super.onBackPressed();
428     }
429
430     @Override
431     public void onAnimationRepeat(Animation animation) {
432
433     }
434 });
435 findViewById(R.id.close).startAnimation(animation);
436
437 }
438
439     @Override
440     public void onAnimationRepeat(Animation animation) {
441
442     }
443 );
444
445 /**
446 * Method for create location request
447 */
448 protected void createLocationRequest() {
449     mLocationRequest = new LocationRequest();
450     mLocationRequest.setInterval(INTERVAL);
451     mLocationRequest.setFastestInterval(FAATEST_INTERVAL);
452     mLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
453     if (ContextCompat.checkSelfPermission(this,
454         Manifest.permission.ACCESS_FINE_LOCATION)
455             == PackageManager.PERMISSION_GRANTED) {
456         if (mClient.isConnected()) {
457             LocationServices.FusedLocationApi.requestLocationUpdates(mClient, mLocationRequest, this);
458         }
459     }
460 }
461
462 /**
463 * Method for callback when activity start
464 */
465 @Override
466 protected void onStart() {
467     super.onStart();
468     mClient.connect();
469 }
470
471 /**
472 * Method for callback when activity is in background for long time
473 */
474 @Override
475 protected void onStop() {
476     mClient.disconnect();
477     super.onStop();
478 }
479
480 /**
481 * Method for callback when google api client connection failed
482 */
483 @Override
484 public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {
485 }
486
487 /**
488 * Method for get callback when place clicked
489 */
490 @Override
491 public void onPlaceClick(ArrayList<PlaceAutocompleteAdapter.PlaceAutocomplete> mResultList, int position) {
492     onProgressChanged(false);
493     if (mResultList != null) {
494         try {
495             final String placeId = String.valueOf(mResultList.get(position).placeId);
496             Log.d("SelectPlaceId", placeId);
497
498             PendingResult<PlaceBuffer> placeResult = Places.GeoDataApi.getPlaceById(mClient, placeId);
499             placeResult.setResultCallback(new ResultCallback<PlaceBuffer>() {
500                 @Override
501                 public void onResult(@NonNull PlaceBuffer places) {
502                     try {
503                         if (placeCust != null) {
504                             Place place = new Place(new LatLng(placeCust.srcLat, placeCust.srcLon), getLocationFromAddress(SearchActivity.this, String.valueOf(mResultList.get(position).description)));
505                             place.setSrcAddress(placeCust.getSrcAddress());
506                             place.setDestAddress(String.valueOf(mResultList.get(position).description));
507                             CustomModel.getInstance().changeState(place);
508                         } else {
509                             Place place = new Place(new LatLng(loc.latitude, loc.longitude), getLocationFromAddress(SearchActivity.this, String.valueOf(mResultList.get(position).description)));
510                             place.setSrcAddress(srcAdd);
511                             place.setDestAddress(String.valueOf(mResultList.get(position).description));
512                             CustomModel.getInstance().changeState(place);
513                         }
514                     } catch (Exception e) {
515                         e.printStackTrace();
516                     }
517                     mSearchEdittext.setText(String.valueOf(mResultList.get(position).description));
518                     onBack();
519
520                 } catch (Exception e) {
521                     e.printStackTrace();
522                 }
523             });
524         } catch (Exception e) {
525             e.printStackTrace();
526         }
527     }
528 }
```

```

529 } else {
530     MapUtils.hideKeyboard(SearchActivity.this);
531     MapUtils.SlideToAbove(mRecyclerView, new Animation.AnimationListener() {
532         @Override
533         public void onAnimationStart(Animation animation) {
534             }
535
536         @Override
537         public void onAnimationEnd(Animation animation) {
538             mRecyclerView.clearAnimation();
539             mRecyclerView.setVisibility(View.GONE);
540             frame.setVisibility(View.VISIBLE);
541             mSearchEdittext.setInputType(InputType.TYPE_NULL);
542             final SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
543                 .findFragmentById(R.id.map);
544             mapFragment.getMapAsync(SearchActivity.this);
545         }
546     }
547
548     @Override
549     public void onAnimationRepeat(Animation animation) {
550
551     }
552 });
553 }
554 }
555 }
556 /**
557 * Method for get lat lng from address
558 */
559 public LatLng getLocationFromAddress(Context context, String strAddress) {
560     Geocoder coder = new Geocoder(context);
561     List<Address> address;
562     LatLng p1 = null;
563     try {
564         address = coder.getFromLocationName(strAddress, 5);
565         if (address == null) {
566             return null;
567         }
568         Address location = address.get(0);
569         p1 = new LatLng(location.getLatitude(), location.getLongitude());
570     } catch (IOException ex) {
571         ex.printStackTrace();
572     }
573
574     return p1;
575 }
576 /**
577 * Method for get callback when map is ready
578 */
579 @Override
580 public void onMapReady(final GoogleMap googleMap) {
581     this.googleMap = googleMap;
582     this.googleMap.getUiSettings().setCompassEnabled(false);
583     this.googleMap.getUiSettings().setMapToolbarEnabled(false);
584     try {
585         googleMap.setMapStyle(
586             MapStyleOptions.loadRawResourceStyle(
587                 this, R.raw.custom_map));
588     } catch (Resources.NotFoundException e) {
589         Log.e("error", "Can't find style. Error: ", e);
590     }
591     createLocationRequest();
592     if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this, Manife
593
594         return;
595     }
596     googleMap.setMyLocationEnabled(true);
597     googleMap.setOnCameraMoveListener(new GoogleMap.OnCameraMoveListener() {
598         @SuppressLint("SetTextI18n")
599         @Override
600         public void onCameraMove() {
601             done.setEnabled(false);
602             done.setText("Fetching...");
603         }
604     });
605
606     View mapView = mapFragment.getView();
607     if (mapView != null && mapView.findViewById(0) != null) {
608         View locationButton = ((View) mapView.findViewById(0).getParent()).findViewById(0);
609         RelativeLayout.LayoutParams layoutParams = (RelativeLayout.LayoutParams)
610             locationButton.getLayoutParams();
611         layoutParams.addRule(RelativeLayout.ALIGN_PARENT_TOP, 0);
612         layoutParams.addRule(RelativeLayout.ALIGN_PARENT_BOTTOM, RelativeLayout.TRUE);
613         layoutParams.setMargins(0, 0, 30, 30);
614     }
615     googleMap.setOnCameraIdleListener(new GoogleMap.OnCameraIdleListener() {
616         @SuppressLint("SetTextI18n")
617         @Override
618         public void onCameraIdle() {
619             LatLng ladw = googleMap.getCameraPosition().target;
620             try {
621                 Geocoder selected_place_geocoder = new Geocoder(SearchActivity.this);
622                 List<Address> address;
623                 address = selected_place_geocoder.getFromLocation(ladw.latitude, ladw.longitude, 1);
624                 Address location = address.get(0);
625                 if (location != null) {
626                     if (location.getLocality() != null) {
627                         String awq = location.getAddressLine(0) + "," + location.getLocality() + "," + location.getCountryName();
628                         mSearchEdittext.setText(awq);
629                     } else {
630                         String gsr = location.getAddressLine(0) + "," + location.getCountryName();
631                         mSearchEdittext.setText(gsr);
632                     }
633                 }
634             }
635         }
636     });
637 }

```

```

637 } catch (Exception e) {
638     e.printStackTrace();
639 }
640 done.setText("Done");
641 done.setEnabled(true);
642 }
643 });
644 done.setOnClickListener(new View.OnClickListener() {
645     @Override
646     public void onClick(View v) {
647         if (done.getText().toString().equalsIgnoreCase("Done")) {
648             if (placeCust != null) {
649                 Place place = new Place(new LatLng(placeCust.srcLat, placeCust.srcLon), googleMap.getCameraPosition().target);
650                 place.setSrcAddress(placeCust.getSrcAddress());
651                 place.setDestAddress(mSearchEdittext.getText().toString());
652                 CustomModel.getInstance().changeState(place);
653             } else {
654                 Place place = new Place(new LatLng(loc.latitude, loc.longitude), googleMap.getCameraPosition().target);
655                 place.setSrcAddress(srcAdd);
656                 place.setDestAddress(mSearchEdittext.getText().toString());
657                 CustomModel.getInstance().changeState(place);
658             }
659             onBack();
660         } else {
661             Toast.makeText(getApplicationContext(), "Wait for Location to be fetched.", Toast.LENGTH_SHORT).show();
662         }
663     }
664 });
665 }
666 /**
667 * Method for callback when location changed
668 */
669 @Override
670 public void onLocationChanged(Location location) {
671     if (carMarker == null) {
672         carMarker = googleMap.addMarker(new MarkerOptions().position(new LatLng(location.getLatitude(), location.getLongitude())).flat(true).icon(BitmapDescriptorFactory.fromResource(R.mipmap.so
673         carMarker.remove();
674         googleMap.setOnMapLoadedCallback(new GoogleMap.OnMapLoadedCallback() {
675             @Override
676             public void onMapLoaded() {
677                 googleMap.animateCamera(CameraUpdateFactory.newLatLngZoom(new LatLng(location.getLatitude(), location.getLongitude()), 18), new GoogleMap.CancelableCallback() {
678                     @Override
679                     public void onFinish() {
680                         pin.setVisibility(View.VISIBLE);
681                         done.setVisibility(View.VISIBLE);
682                     }
683                 });
684             }
685             @Override
686             public void onCancel() {
687                 pin.setVisibility(View.VISIBLE);
688                 done.setVisibility(View.VISIBLE);
689             }
690         });
691     } else {
692         carMarker.setPosition(new LatLng(location.getLatitude(), location.getLongitude()));
693     }
694 }
695 /**
696 * Method for get callback when progress changed
697 */
698 @Override
699 public void onProgressChanged(boolean state) {
700     startAnimation(state);
701 }

```

```

1 package com.custom.co2.activity;
2
3 import android.Manifest;
4 import android.app.Activity;
5 import android.content.Intent;
6 import android.content.IntentSender;
7 import android.content.pm.PackageManager;
8 import android.os.Bundle;
9 import android.os.Handler;
10 import android.view.WindowManager;
11 import android.widget.Toast;
12
13 import androidx.annotation.NonNull;
14 import androidx.annotation.Nullable;
15 import androidx.appcompat.app.AppCompatActivity;
16 import androidx.core.app.ActivityCompat;
17
18 import com.custom.co2.R;
19 import com.google.android.gms.common.ConnectionResult;
20 import com.google.android.gms.common.api.GoogleApiClient;
21 import com.google.android.gms.common.api.PendingResult;
22 import com.google.android.gms.common.api.ResultCallback;
23 import com.google.android.gms.common.api.Status;
24 import com.google.android.gms.location.LocationRequest;
25 import com.google.android.gms.location.LocationServices;
26 import com.google.android.gms.location.LocationSettingsRequest;
27 import com.google.android.gms.location.LocationSettingsResult;
28 import com.google.android.gms.location.LocationSettingsStates;
29 import com.google.android.gms.location.LocationSettingsStatusCodes;
30
31 import org.jetbrains.annotations.NotNull;
32
33 import static com.custom.co2.utils.Constant.getShaedPref;
34 import static com.custom.co2.utils.Constant.spEmail;
35
36 public class SplashActivity extends AppCompatActivity implements
37     GoogleApiClient.ConnectionCallbacks, GoogleApiClient.OnConnectionFailedListener{
38
39     GoogleApiClient mGoogleApiClient;
40
41     /**
42      * Method for initialize layout
43      */
44     @Override
45     protected void onCreate(Bundle savedInstanceState) {
46         super.onCreate(savedInstanceState);
47         this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);
48         getSupportActionBar().hide();
49         setContentView(R.layout.activity_splash);
50         locationPermissions();
51     }
52
53     private void locationPermissions() {
54         if (ActivityCompat.checkSelfPermission(SplashActivity.this, Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
55             ActivityCompat.checkSelfPermission(SplashActivity.this, Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
56             ActivityCompat.requestPermissions(SplashActivity.this, new String[]{Manifest.permission.ACCESS_FINE_LOCATION}, 1);
57             return;
58         } else {
59             turnGPSOn();
60         }
61     }
62
63     /**
64      * Method for check permission request
65      */
66     public void onRequestPermissionsResult(int requestCode, @NotNull String permissions[], @NotNull int[] grantResults) {
67         if (requestCode == 1) {
68             if (grantResults.length > 0
69                 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
70                 turnGPSOn();
71             }
72         }
73     }
74
75     /**
76      * Method for enable gps
77      */
78     private void turnGPSOn() {
79
80         if (mGoogleApiClient == null) {
81             mGoogleApiClient = new GoogleApiClient.Builder(SplashActivity.this)
82                 .addApi(LocationServices.API).addConnectionCallbacks(this)
83                 .addOnConnectionFailedListener(SplashActivity.this).build();
84             mGoogleApiClient.connect();
85             LocationRequest locationRequest = LocationRequest.create();
86             locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
87             locationRequest.setInterval(30 * 1000);
88             locationRequest.setFastestInterval(5 * 1000);
89

```

```

90 LocationSettingsRequest.Builder builder = new LocationSettingsRequest.Builder()
91     .addLocationRequest(locationRequest);
92 builder.setAlwaysShow(true); // this is the key ingredient
93 PendingResult<LocationSettingsResult> result = LocationServices.SettingsApi
94     .checkLocationSettings(mGoogleApiClient, builder.build());
95 result.setResultCallback(new ResultCallback<LocationSettingsResult>() {
96     @Override
97     public void onResult(LocationSettingsResult result) {
98         final Status status = result.getStatus();
99         final LocationSettingsStates state = result
100            .getLocationSettingsStates();
101         switch (status.getStatusCode()) {
102             case LocationSettingsStatusCodes.SUCCESS:
103                 SplashIntent();
104                 break;
105             case LocationSettingsStatusCodes.RESOLUTION_REQUIRED:
106                 try {
107                     status.startResolutionForResult(SplashActivity.this, 1000);
108                 } catch (IntentSender.SendIntentException e) {
109                 }
110                 break;
111             case LocationSettingsStatusCodes.SETTINGS_CHANGE_UNAVAILABLE:
112                 break;
113         }
114     }
115 });
116 });
117 }
118 }
119 }
120 /**
121 * Method for redirect to login or home
122 */
123 private void SplashIntent() {
124     new Handler().postDelayed(new Runnable() {
125         @Override
126         public void run() {
127             if (getShaedPref(SplashActivity.this, spEmail).equals("")) {
128                 startActivity(new Intent(getApplicationContext(), LoginActivity.class));
129                 finish();
130             } else {
131                 startActivity(new Intent(getApplicationContext(), HomeActivity.class).setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK));
132             }
133         }
134     }, 2500);
135 }
136 }
137 }
138 @Override
139 protected void onActivityResult(int requestCode, int resultCode, Intent data) {
140     super.onActivityResult(requestCode, resultCode, data);
141     switch (requestCode) {
142         case 1000:
143             if (resultCode == Activity.RESULT_OK) {
144                 SplashIntent();
145             } else if (resultCode == Activity.RESULT_CANCELED) {
146             }
147             break;
148         }
149     }
150 }
151 }
152 }
153 @Override
154 public void onConnected(@Nullable Bundle bundle) {
155 }
156 }
157 }
158 @Override
159 public void onConnectionSuspended(int i) {
160 }
161 }
162 }
163 @Override
164 public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {
165 }
}

```

```
1 package com.custom.co2.activity;
2
3 import android.app.DatePickerDialog;
4 import android.app.Dialog;
5 import android.os.Bundle;
6 import android.util.DisplayMetrics;
7 import android.util.Log;
8 import android.view.View;
9 import android.view.WindowManager;
10 import android.widget.ImageView;
11 import android.widget.TextView;
12
13 import androidx.appcompat.app.AppCompatActivity;
14 import androidx.constraintlayout.widget.ConstraintLayout;
15 import androidx.recyclerview.widget.LinearLayoutManager;
16 import androidx.recyclerview.widget.RecyclerView;
17
18 import com.custom.co2.R;
19 import com.custom.co2.adapter.Statisticdapter;
20 import com.custom.co2.utils.Constant;
21 import com.google.firebaseio.DocumentSnapshot;
22 import com.google.firebaseio.FirebaseFirestore;
23 import com.google.firebaseio.QuerySnapshot;
24
25 import java.text.SimpleDateFormat;
26 import java.util.ArrayList;
27 import java.util.Calendar;
28 import java.util.Date;
29 import java.util.Locale;
30 import java.util.Objects;
31
32 import static com.custom.co2.utils.Constant.getShaedPref;
33 import static com.custom.co2.utils.Constant.showProgressDialog;
34
35 public class StatisticsActivity extends AppCompatActivity {
36     FirebaseFirestore db;
37     String TAG = "StatisticsActivity";
38     Statisticdapter statisticdapter;
39     ImageView btnBack, filter;
40     ConstraintLayout crdCount;
41     Dialog progressDialog;
42     ArrayList<DocumentSnapshot> arrayList = new ArrayList<DocumentSnapshot>();
43     RecyclerView recStatistic;
44     TextView txtCO2, txtCalories;
45
46     double totalCO2 = 0.0, totalCalories = 0.0;
47
48     /**
49      * Method for initialize layout
50      */
51     @Override
52     protected void onCreate(Bundle savedInstanceState) {
53         super.onCreate(savedInstanceState);
54         getSupportActionBar().hide();
55         setContentView(R.layout.activity_statistics);
56         inits();
57
58         db = FirebaseFirestore.getInstance();
59         progressDialog = showProgressDialog(StatisticsActivity.this);
60         getStatisticsList(new SimpleDateFormat("dd-MM-yyyy", Locale.getDefault()).format(new Date()));
61     }
62
63     /**
64      * Method for init variable and widgets
65      */
66     private void inits() {
67
68         recStatistic = findViewById(R.id.recStatistic);
```

```

69     btnBack = findViewById(R.id.iv_back);
70     filter = findViewById(R.id.iv_filter);
71
72     crdCount = findViewById(R.id.crd_count);
73     txtCO2 = findViewById(R.id.txtCO2);
74     txtCalories = findViewById(R.id.txtCalories);
75
76     recStatistic.setLayoutManager(new LinearLayoutManager(this, LinearLayoutManager.VERTICAL, false));
77     statisticdapter = new Statisticdapter(this, arrayList);
78     recStatistic.setAdapter(statisticdapter);
79     btnBack.setOnClickListener(new View.OnClickListener() {
80         @Override
81         public void onClick(View view) {
82             finish();
83         }
84     });
85
86     filter.setOnClickListener(new View.OnClickListener() {
87         @Override
88         public void onClick(View view) {
89             showDateFilter();
90         }
91     });
92
93
94 }
95
96 /**
97 * Method for show date filter dialog
98 */
99 private void showDateFilter() {
100
101
102     final Dialog dialog = new Dialog(StatisticsActivity.this);
103     dialog.setContentView(R.layout.dialog_statistic_filter);
104     Objects.requireNonNull(dialog.getWindow()).setBackgroundDrawableResource(android.R.color.transparent);
105     dialog.setCancelable(false);
106     DisplayMetrics displaymetrics = new DisplayMetrics();
107     this.getWindowManager().getDefaultDisplay().getMetrics(displaymetrics);
108     WindowManager.LayoutParams lp = new WindowManager.LayoutParams();
109     lp.copyFrom(dialog.getWindow().getAttributes());
110     lp.width = (int) (displaymetrics.widthPixels * 0.90);
111     dialog.getWindow().setAttributes(lp);
112     TextView txtDate = dialog.findViewById(R.id.txtDate);
113
114     txtDate.setText(new SimpleDateFormat("dd-MM-yyyy", Locale.getDefault()).format(new Date()));
115
116     SimpleDateFormat dateFormatter = new SimpleDateFormat("dd-MM-yyyy", Locale.getDefault());
117     TextView btn_submit = dialog.findViewById(R.id.btn_submit);
118     txtDate.setOnClickListener(v -> {
119         final Calendar newCalendar = Calendar.getInstance();
120         final DatePickerDialog StartTime = new DatePickerDialog(this, (view, year, monthOfYear, dayOfMonth) -> {
121             Calendar newDate = Calendar.getInstance();
122             newDate.set(year, monthOfYear, dayOfMonth);
123             txtDate.setText(dateFormatter.format(newDate.getTime()));
124         }, newCalendar.get(Calendar.YEAR), newCalendar.get(Calendar.MONTH), newCalendar.get(Calendar.DAY_OF_MONTH));
125         StartTime.show();
126     });
127     btn_submit.setOnClickListener(v -> {
128         getStatisticsList(txtDate.getText().toString());
129         dialog.dismiss();
130     });
131
132     dialog.show();
133 }
134
135
136 /**
137 * Method for fetch statistic data from firebase
138 */
139 private void getStatisticsList(String Date) {
140

```

```

140     progressDialog.show();
141     arrayList.clear();
142     totalCO2 = 0.0;
143     totalCalories = 0.0;
144     db.collection("Co2-Report")
145         .whereEqualTo("Date", Date)
146         .whereEqualTo("UserID", getShaedPref(this, Constant.spUserId))
147         .get()
148         .addOnCompleteListener(task -> {
149
150             if (task.isSuccessful()) {
151                 QuerySnapshot document = task.getResult();
152                 if (!document.isEmpty()) {
153
154                     if (document.getDocuments().size() > 0) {
155                         arrayList.addAll(document.getDocuments());
156                         statisticdapter.notifyDataSetChanged();
157                         findViewById(R.id.txtNoData).setVisibility(View.GONE);
158                         crdCount.setVisibility(View.VISIBLE);
159                     } else {
160                         findViewById(R.id.txtNoData).setVisibility(View.VISIBLE);
161                         crdCount.setVisibility(View.GONE);
162                     }
163                     for (int i = 0; i < arrayList.size(); i++) {
164
165                         Log.d(TAG, "getStatisticsList: " + Double.parseDouble(arrayList.get(i).getString("Co2")));
166
167                         if (arrayList.get(i).getBoolean("Co2Type")) {
168                             totalCO2 = totalCO2 + Double.parseDouble(arrayList.get(i).getString("Co2"));
169                         } else {
170                             totalCalories = totalCalories + Double.parseDouble(arrayList.get(i).getString("Co2"));
171                         }
172
173                     }
174
175                     txtCO2.setText(String.valueOf(totalCO2));
176                     txtCalories.setText(String.valueOf(totalCalories));
177                 } else {
178                     if (arrayList.size() > 0) {
179                         findViewById(R.id.txtNoData).setVisibility(View.GONE);
180                         crdCount.setVisibility(View.VISIBLE);
181                     } else {
182                         findViewById(R.id.txtNoData).setVisibility(View.VISIBLE);
183                         crdCount.setVisibility(View.GONE);
184                     }
185                     statisticdapter.notifyDataSetChanged();
186                 }
187             }
188         } else {
189             Log.d(TAG, "get failed with ", task.getException());
190         }
191         progressDialog.dismiss();
192     });
193 }
}

```

```
1 package com.custom.co2.activity;
2
3 import android.app.DatePickerDialog;
4 import android.app.Dialog;
5 import android.os.Bundle;
6 import android.util.Log;
7 import android.util.Patterns;
8 import android.widget.DatePicker;
9 import android.widget.EditText;
10 import android.widget.TextView;
11 import android.widget.Toast;
12
13 import androidx.appcompat.app.AppCompatActivity;
14
15 import com.custom.co2.R;
16 import com.custom.co2.utils.Constant;
17 import com.google.firebase.firestore.FirebaseFirestore;
18
19 import java.text.SimpleDateFormat;
20 import java.util.Calendar;
21 import java.util.HashMap;
22 import java.util.Locale;
23 import java.util.Map;
24
25 import static com.custom.co2.utils.Constant.getShaedPref;
26 import static com.custom.co2.utils.Constant.setShaedPref;
27 import static com.custom.co2.utils.Constant.showProgressDialog;
28 import static com.custom.co2.utils.Constant.spConatact;
29 import static com.custom.co2.utils.Constant.spDob;
30 import static com.custom.co2.utils.Constant.spEmail;
31 import static com.custom.co2.utils.Constant.spUsername;
32
33 public class UpdateProfileActivity extends AppCompatActivity {
34     FirebaseFirestore db;
35     String TAG = "UpdateProfileActivity";
36     TextView btnLogin;
37     EditText edUsername, edEmail, edContact, edDob;
38     final Calendar myCalendar = Calendar.getInstance();
39     Dialog progressDialog;
40
41     /**
42      * Method for initialize layout
43      */
44     @Override
45     protected void onCreate(Bundle savedInstanceState) {
46         super.onCreate(savedInstanceState);
47         getSupportActionBar().hide();
48         setContentView(R.layout.activity_update_profile);
49
50         inits();
51         clicks();
52
53         db = FirebaseFirestore.getInstance();
54         progressDialog = showProgressDialog(UpdateProfileActivity.this);
55     }
56
57     /**
58      * Method for manage clicks
59      */
60     private void clicks() {
61
62         btnLogin.setOnClickListener(view -> {
63
64             if (Valid()) {
```

```
65     updateDataToCloud(edUsername.getText().toString(), edEmail.getText().toString(),
66                         edDob.getText().toString(), edContact.getText().toString());
67 }
68 });
69 edDob.setOnClickListener(view -> {
70
71     myCalendar.add(Calendar.YEAR, -10);
72     int year = myCalendar.get(Calendar.YEAR);
73     int month = myCalendar.get(Calendar.MONTH);
74     int day = myCalendar.get(Calendar.DAY_OF_MONTH);
75
76     DatePickerDialog datePicker = new DatePickerDialog(UpdateProfileActivity.this, date, year, month, day);
77
78     datePicker.getDatePicker().setMaxDate(myCalendar.getTimeInMillis());
79     datePicker.show();
80
81 });
82 }
83
84 /**
85 * Method for validations
86 */
87
88 private boolean Valid() {
89     if (edUsername.getText().toString().isEmpty()) {
90         edUsername.setError("Enter username");
91         edUsername.requestFocus();
92         return false;
93     }
94     else if (edContact.getText().toString().isEmpty()) {
95         edContact.setError("Enter contact");
96         edContact.requestFocus();
97         return false;
98     }
99     else if (edContact.getText().toString().length() != 10) {
100        edContact.setError("Enter valid contact");
101        edContact.requestFocus();
102        return false;
103    }
104    else if (edDob.getText().toString().isEmpty()) {
105        edDob.setError("Enter date of birth");
106        edDob.requestFocus();
107        return false;
108    }
109    else if (edEmail.getText().toString().isEmpty()) {
110        edEmail.setError("Enter email");
111        edEmail.requestFocus();
112        return false;
113    }
114    else if (!Patterns.EMAIL_ADDRESS.matcher(edEmail.getText().toString()).matches()) {
115        edEmail.setError("Enter valid email");
116        edEmail.requestFocus();
117        return false;
118    }
119 }
120 return true;
121 }
122 }
123
124 /**
125 * Method for init variable and widgets
126 */
127
128 private void inits() {
129     btnLogin = findViewById(R.id.tv_btn_login);
130     edUsername = findViewById(R.id.ed_username);
131     edEmail = findViewById(R.id.ed_email);
132     edDob = findViewById(R.id.ed_dob);
```

```
132     edContact = findViewById(R.id.ed_contact);
133
134     edUsername.setText(getShaedPref(this, spUsername));
135     edEmail.setText(getShaedPref(this, spEmail));
136     edDob.setText(getShaedPref(this, spDob));
137     edContact.setText(getShaedPref(this, spConatact));
138
139
140 }
141
142 /**
143 * Method for date listener
144 */
145 DatePickerDialog.OnDateSetListener date = new DatePickerDialog.OnDateSetListener() {
146
147     @Override
148     public void onDateSet(DatePicker view, int year, int monthOfYear, int dayOfMonth) {
149         // TODO Auto-generated method stub
150         myCalendar.set(Calendar.YEAR, year);
151         myCalendar.set(Calendar.MONTH, monthOfYear);
152         myCalendar.set(Calendar.DAY_OF_MONTH, dayOfMonth);
153         updateLabel();
154     }
155
156
157 };
158
159 /**
160 * Method for change date label
161 */
162 private void updateLabel() {
163     String myFormat = "dd/MM/yyyy";
164     SimpleDateFormat sdf = new SimpleDateFormat(myFormat, Locale.US);
165
166     edDob.setText(sdf.format(myCalendar.getTime()));
167     edDob.setError(null);
168 }
169
170 /**
171 * Method for update date to firebase
172 */
173 private void updateDataToCloud(String username, String email, String dob, String contact) {
174     progressDialog.show();
175     Map<String, Object> user = new HashMap<>();
176     user.put("Username", username);
177     user.put("DOB", dob);
178     user.put("Contact", contact);
179
180     db.collection("users").document(getShaedPref(this, Constant.spUserId))
181         .update(user)
182         .addOnSuccessListener(aVoid -> {
183
184             setShaedPref(UpdateProfileActivity.this, spUsername,username);
185             setShaedPref(UpdateProfileActivity.this, spConatact, contact);
186             setShaedPref(UpdateProfileActivity.this, spDob, dob);
187             Toast.makeText(UpdateProfileActivity.this, "Profile Updated", Toast.LENGTH_SHORT).show();
188             progressDialog.dismiss();
189
190         }).addOnFailureListener(e -> {
191             Log.e(TAG, "registerDataToCloud: " + e.getMessage());
192             progressDialog.dismiss();
193         });
194 }
}
```

```
package com.custom.co2.adapter;

import android.annotation.SuppressLint;
import android.content.Context;
import android.graphics.Typeface;
import android.text.style.CharacterStyle;
import android.text.style.StyleSpan;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Filter;
import android.widget.Filterable;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;

import com.custom.co2.R;
import com.custom.co2.utils.CustomProgress;
import com.google.android.gms.common.api.ApiException;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.location.places.AutocompleteFilter;
import com.google.android.gms.maps.model.LatLngBounds;
import com.google.android.libraries.places.api.model.AutocompleteSessionToken;
import com.google.android.libraries.places.api.net.FindAutocompletePredictionsRequest;
import com.google.android.libraries.places.api.net.PlacesClient;

import java.util.ArrayList;

public class PlaceAutocompleteAdapter extends RecyclerView.Adapter<PlaceAutocompleteAdapter.PlaceViewHolder> implements Filterable {

    private ImageView iv_line;

    public void setProgress(ImageView iv_line) {
        this.iv_line = iv_line;
    }

    public interface PlaceAutoCompleteInterface {
        public void onPlaceClick(ArrayList<PlaceAutocomplete> mResultList, int position);
    }

    Context mContext;
    PlaceAutoCompleteInterface mListener;
    private static final String TAG = "PlaceAutocompleteAdapter";
    private static final CharacterStyle STYLE_BOLD = new StyleSpan(Typeface.BOLD);
    private ArrayList<PlaceAutocomplete> mResultList = new ArrayList<>();
    private GoogleApiClient mGoogleApiClient;
    private LatLngBounds mBounds;
    private AutocompleteFilter mPlaceFilter;
    ArrayList<PlaceAutocomplete> resultList = new ArrayList<>();

    public PlaceAutocompleteAdapter(Context context, GoogleApiClient googleApiClient, LatLngBounds bounds, AutocompleteFilter filter) {
        this.mContext = context;
        mGoogleApiClient = googleApiClient;
        mBounds = bounds;
        mPlaceFilter = filter;
        this.mListener = (PlaceAutoCompleteInterface) mContext;
    }

    /*
     * Clear List items
     */
    public void clearList() {
        if (mResultList != null && mResultList.size() > 0) {
            mResultList.clear();
            mResultList = null;
        }
    }

    @Override
    public Filter getFilter() {
```

```

15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154

    return new Filter() {
        @Override
        protected FilterResults performFiltering(CharSequence constraint) {
            FilterResults results = new FilterResults();
            if (constraint != null) {
                CustomProgress.getInstance().changeState(true);
                mResultList = getAutocomplete(constraint);
                if (mResultList != null) {
                    results.values = mResultList;
                    results.count = mResultList.size();
                }
            }
            return results;
        }

        @Override
        protected void publishResults(CharSequence constraint, FilterResults results) {
            if (results != null && results.count > 0) {
                mResultList = (ArrayList<PlaceAutocomplete>) results.values;
                notifyDataSetChanged();
            } else {
                if (mResultList != null) {
                    mResultList.clear();
                    mResultList = null;
                }
                notifyItemRangeChanged(0, 0);
            }
            CustomProgress.getInstance().changeState(false);
        }
    };
}

private ArrayList<PlaceAutocomplete> getAutocomplete(CharSequence constraint) {
    if (mGoogleApiClient.isConnected()) {
        AutocompleteSessionToken token = AutocompleteSessionToken.newInstance();
        FindAutocompletePredictionsRequest request = FindAutocompletePredictionsRequest.builder()
            .setSessionToken(token)
            .setQuery(String.valueOf(constraint))
            .build();

        com.google.android.libraries.places.api.Places.initialize(mContext, mContext.getResources().getString(R.string.Map_Api_Key));

        PlacesClient placesClient = com.google.android.libraries.places.api.Places.createClient(mContext);

        placesClient.findAutocompletePredictions(request).addOnSuccessListener((response) -> {
            Log.d("ResultArray", String.valueOf(response.getAutocompletePredictions()));

            mResultList = new ArrayList<>();

            for (com.google.android.libraries.places.api.model.AutocompletePrediction prediction : response.getAutocompletePredictions()) {
                mResultList.add(new PlaceAutocomplete(prediction.getPlaceId(),
                    prediction.getFullText(STYLE_BOLD), prediction.getSecondaryText(STYLE_BOLD)));
            }
        }).addOnFailureListener((exception) -> {
            if (exception instanceof ApiException) {
                ApiException apiException = (ApiException) exception;
            }
        });
    }
    return mResultList;
}

@NonNull
@Override
public PlaceViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int viewType) {
    View v0;
    v0 = LayoutInflater.from(viewGroup.getContext()).inflate(R.layout.expandlistitem, viewGroup, false);
    return new PlaceViewHolder(v0);
}

@Override
public void onBindViewHolder(@NonNull PlaceViewHolder mPredictionHolder, @SuppressLint("RecyclerView") final int i) {
    switch (getItemViewType(i)) {
        case 0:
            mPredictionHolder.itemView.setOnClickListener(new View.OnClickListener() {
                ...
            });
    }
}

```

```
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
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
232
233
```

```
    @Override
    public void onClick(View v) {
        }
    });
    break;
    case 1:
        if (mResultList != null && mResultList.size() > 0) {
            if (mPredictionHolder.mParentLayout != null) {
                mPredictionHolder.mParentLayout.setVisibility(View.VISIBLE);
                mPredictionHolder.mParentLayout.setText(mResultList.get(i).description);
                mPredictionHolder.mAddress.setText(mResultList.get(i).second);
            } else {
                mPredictionHolder.mParentLayout.setVisibility(View.GONE);
            }
        } else {
            mPredictionHolder.mParentLayout.setVisibility(View.GONE);
        }
        mPredictionHolder.itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                mListener.onPlaceClick(mResultList, i);
            }
        });
    });
    break;
}
}

@Override
public int getItemViewType(int position) {
    if (mResultList != null && position < mResultList.size()) {
        if (mResultList != null && mResultList.get(position).placeId != null)
            return 1;
        else
            return 0;
    }
    return 0;
}

@Override
public int getItemCount() {
    if (mResultList != null)

        return mResultList.size();
    else
        return 1;
}

class PlaceViewHolder extends RecyclerView.ViewHolder {
    // CardView mCardView;
    TextView mParentLayout;
    TextView mAddress;

    PlaceViewHolder(View itemView) {
        super(itemView);
        mParentLayout = itemView.findViewById(R.id.text1);
        mAddress = itemView.findViewById(R.id.text2);
    }
}

/**
 * Holder for Places Geo Data Autocomplete API results.
 */
public class PlaceAutocomplete {

    public CharSequence placeId;
    public CharSequence description = "";
    public CharSequence second = "";

    PlaceAutocomplete(CharSequence placeId, CharSequence description, CharSequence secondaryText) {
        this.placeId = placeId;
        this.description = description;
        this.second = secondaryText;
    }

    @Override
    public String toString() {
        return description.toString();
    }
}
```



```

1 package com.custom.co2.adapter;
2
3 import android.content.Context;
4 import android.view.LayoutInflater;
5 import android.view.View;
6 import android.view.ViewGroup;
7 import android.widget.TextView;
8
9 import androidx.recyclerview.widget.RecyclerView;
10
11 import com.custom.co2.R;
12 import com.google.firebaseio.DocumentSnapshot;
13
14 import java.util.ArrayList;
15
16
17 public class StatisticAdapter extends RecyclerView.Adapter<StatisticAdapter.ViewHolder> {
18     ArrayList<DocumentSnapshot> arrayListData;
19     Context context;
20
21     public StatisticAdapter(Context context, ArrayList<DocumentSnapshot> arrayListData) {
22         this.arrayListData = arrayListData;
23         this.context = context;
24     }
25
26     @Override
27     public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
28         LayoutInflater inflater = LayoutInflater.from(parent.getContext());
29         View listItem = inflater.inflate(R.layout.list_item, parent, false);
30         ViewHolder viewHolder = new ViewHolder(listItem);
31         return viewHolder;
32     }
33
34     @Override
35     public void onBindViewHolder(ViewHolder holder, int position) {
36
37         holder.txtsource.setText(arrayListData.get(position).getString("Source"));
38         holder.txtDestination.setText(arrayListData.get(position).getString("Destination"));
39         holder.txtDistance.setText(arrayListData.get(position).getString("Distance"));
40         holder.txtCO2.setText(arrayListData.get(position).getString("Co2"));
41
42         if (arrayListData.get(position).getString("RideType").equals("CAR"))
43             holder.txtVehicle.setText(arrayListData.get(position).getString("RideType") + "(" + (arrayListData.get(position).getString("VehicleType")
44                 + "/" + arrayListData.get(position).getString("FuleType")) + ")");
45         else
46             holder.txtVehicle.setText(arrayListData.get(position).getString("RideType"));
47
48         holder.txtDate.setText(arrayListData.get(position).getString("Date"));
49         if (arrayListData.get(position).getBoolean("Co2Type"))
50             holder.txtCo2Title.setText("Co2");
51         else
52             holder.txtCo2Title.setText("Calories");
53     }
54
55
56     @Override
57     public int getItemCount() {
58         return arrayListData.size();
59     }
60
61     public static class ViewHolder extends RecyclerView.ViewHolder {
62
63         public TextView txtsource, txtDestination, txtDistance, txtCO2, txtCo2Title, txtVehicle, txtDate;
64
65
66         public ViewHolder(View itemView) {
67             super(itemView);
68             txtsource = (TextView) itemView.findViewById(R.id.txtsource);
69             txtDestination = (TextView) itemView.findViewById(R.id.txtDestination);
70             txtDistance = (TextView) itemView.findViewById(R.id.txtDistance);
71             txtCO2 = (TextView) itemView.findViewById(R.id.txtCO2);
72             txtCo2Title = (TextView) itemView.findViewById(R.id.txtCo2Title);
73             txtVehicle = (TextView) itemView.findViewById(R.id.txtVehicle);
74             txtDate = (TextView) itemView.findViewById(R.id.txtDate);

```

75 }
76 }
 }

```
1 package com.custom.co2.utils;
2
3
4 import android.animation.ValueAnimator;
5 import android.view.animation.LinearInterpolator;
6
7 public class AnimationUtils {
8     public static ValueAnimator carAnimator() {
9         ValueAnimator valueAnimator = ValueAnimator.ofFloat(0f, 1f);
10        valueAnimator.setDuration(900);
11        valueAnimator.setInterpolator(new LinearInterpolator());
12        return valueAnimator;
13    }
14
15 }
```

```
1 package com.custom.co2.utils;
2
3 import android.app.Application;
4
5 public class App extends Application {
6     @Override
7     public void onCreate() {
8         super.onCreate();
9     }
10 }
11 }
```

```
1 package com.custom.co2.utils;
2
3 public class CustomDirection {
4     private static CustomDirection mInstance;
5     private DirectionListener mListener;
6
7     private CustomDirection() {
8     }
9
10    public static CustomDirection getInstance() {
11        if (mInstance == null) {
12            mInstance = new CustomDirection();
13        }
14        return mInstance;
15    }
16
17    public void setListener(DirectionListener listener) {
18        mListener = listener;
19    }
20
21    public void changeState(DirectionUtils place) {
22        if (mListener != null) {
23            notifyStateChange(place);
24        }
25    }
26
27    private void notifyStateChange(DirectionUtils place) {
28        mListener.onDirectionsData(place);
29    }
30 }
```

```
1 package com.custom.co2.utils;
2
3 import android.app.Activity;
4 import android.content.Context;
5 import android.graphics.Bitmap;
6 import android.util.Log;
7 import android.view.View;
8
9 import com.custom.co2.R;
10 import com.google.android.gms.maps.GoogleMap;
11 import com.google.android.gms.maps.model.BitmapDescriptorFactory;
12 import com.google.android.gms.maps.model.MarkerOptions;
13 import com.google.maps.android.clustering.Cluster;
14 import com.google.maps.android.clustering.ClusterItem;
15 import com.google.maps.android.clustering.ClusterManager;
16 import com.google.maps.android.clustering.view.DefaultClusterRenderer;
17 import com.google.maps.android.ui.IconGenerator;
18
19
20 public class CustomMapClusterRenderer <T extends ClusterItem> extends DefaultClusterRenderer<T> {
21     private IconGenerator mClusterIconGenerator;
22
23     public CustomMapClusterRenderer(Context context, GoogleMap map, ClusterManager<T> clusterManager) {
24         super(context, map, clusterManager);
25         mClusterIconGenerator = new IconGenerator(
26             context);
27         Activity activity = (Activity) context;
28         View multiProfile = activity.getLayoutInflater().inflate(
29             R.layout.custom_marker_infowindow_left, null);
30         mClusterIconGenerator.setContentView(multiProfile);
31     }
32
33     @Override
34     protected void onBeforeClusterRendered(Cluster<T> cluster,
35                                         MarkerOptions markerOptions) {
36         try {
37             mClusterIconGenerator.setBackground(null);
38             Bitmap icon = mClusterIconGenerator.makeIcon(String.valueOf(cluster
39                 .getCount()));
40             markerOptions.icon(BitmapDescriptorFactory.fromBitmap(icon));
41         } catch (Exception e) {
42             e.printStackTrace();
43             Log.e("get_item_list_nir", "error 13.1 : " + e.toString());
44         }
45         Log.e("get_item_list_nir", "CallMap onBeforeClusterRendered 14");
46     }
47
48     @Override
49     protected void onBeforeClusterItemRendered(T item,
50                                         MarkerOptions markerOptions) {
51         markerOptions.icon(BitmapDescriptorFactory.fromResource(R.mipmap.new_car_small));
52         super.onBeforeClusterItemRendered(item, markerOptions);
53     }
54 }
```

```
package com.custom.co2.utils;

public class CustomModel {
    private static CustomModel mInstance;
    private PlaceListener mListener;
    private CustomModel() {}

    public static CustomModel getInstance() {
        if(mInstance == null) {
            mInstance = new CustomModel();
        }
        return mInstance;
    }

    public void setListener(PlaceListener listener) {
        mListener = listener;
    }

    public void changeState(Place place) {
        if(mListener != null) {
            notifyStateChange(place);
        }
    }

    private void notifyStateChange(Place place) {
        mListener.onPlace(place);
    }
}
```

```
package com.custom.co2.utils;

public class CustomPickUp {
    private static CustomPickUp mInstance;
    private PlaceListener mListener;
    private CustomPickUp() {}

    public static CustomPickUp getInstance() {
        if(mInstance == null) {
            mInstance = new CustomPickUp();
        }
        return mInstance;
    }

    public void setListener(PlaceListener listener) {
        mListener = listener;
    }

    public void changeState(Place place) {
        if(mListener != null) {
            notifyStateChange(place);
        }
    }

    private void notifyStateChange(Place place) {
        mListener.onPlace(place);
    }
}
```

```
package com.custom.co2.utils;

public interface DirectionListener {
    public void onDirectionsData(DirectionUtils utils);
}
```

```
package com.custom.co2.utils;

import android.text.Html;
import android.util.Log;

import com.google.android.gms.maps.model.LatLng;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

public class DirectionsJSONParser {

    String distance = "1.2";
    String duration = "1.3";
    String stAddress = "aaaa";
    String endAddress = "bbbb";

    public List<List<HashMap<String, String>>> parse(JSONObject jObject) {

        List<List<HashMap<String, String>>> routes = new ArrayList<List<HashMap<String, String>>>();
        JSONArray jRoutes = null;
        JSONArray jLegs = null;
        JSONArray jSteps = null;

        try {

            jRoutes = jObject.getJSONArray("routes");

            /** Traversing all routes */
            for (int i = 0; i < jRoutes.length(); i++) {
                jLegs = ((JSONObject) jRoutes.get(i)).getJSONArray("legs");
                List path = new ArrayList<HashMap<String, String>>();

                /** Traversing all legs */
                for (int j = 0; j < jLegs.length(); j++) {
                    JSONObject js = jLegs.getJSONObject(j);
                    JSONObject ss = js.getJSONObject("distance");
                    JSONObject mm = js.getJSONObject("duration");
                    String time = mm.getString("text");
                    Log.e("timing", time);
                    String secs = mm.getString("value");
                    distance = ss.getString("text");
                    duration = mm.getString("text");
                    endAddress = js.getString("end_address");
                    stAddress = js.getString("start_address");
                    DirectionUtils utils = new DirectionUtils();
                    utils.setDistance(distance);
                    utils.setDuration(duration);
                    utils.setEndAddress(endAddress);
                    utils.setStAddress(stAddress);
                    CustomDirection.getInstance().changeState(utils);
                    jSteps = ((JSONObject) jLegs.get(j)).getJSONArray("steps");

                    /** Traversing all steps */
                    for (int k = 0; k < jSteps.length(); k++) {
                        String polyline = "";
                        polyline = (String) ((JSONObject) ((JSONObject) jSteps.get(k)).get("polyline")).get("points");
                        List<LatLng> list = decodePoly(polyline);

                        /** Traversing all points */
                        for (int l = 0; l < list.size(); l++) {

```

```

        HashMap<String, String> hm = new HashMap<String, String>();
        hm.put("lat", Double.toString(((LatLng) list.get(l)).latitude));
        hm.put("lng", Double.toString(((LatLng) list.get(l)).longitude));
        String ins = ((JSONObject) jSteps.get(k)).getString("html_instructions");
        hm.put("instructions", String.valueOf(Html.fromHtml(ins)));
        hm.put("distance", distance);
        hm.put("duration", duration);
        hm.put("endAddress", endAddress);
        hm.put("stAddress", stAddress);
        path.add(hm);
    }
}
routes.add(path);
}
}

} catch (JSONException e) {
    e.printStackTrace();
} catch (Exception e) {
}

return routes;
}

private List<LatLng> decodePoly(String encoded) {

List<LatLng> poly = new ArrayList<LatLng>();
int index = 0, len = encoded.length();
int lat = 0, lng = 0;

while (index < len) {
    int b, shift = 0, result = 0;
    do {
        b = encoded.charAt(index++) - 63;
        result |= (b & 0x1f) << shift;
        shift += 5;
    } while (b >= 0x20);
    int dlat = ((result & 1) != 0 ? ~(result >> 1) : (result >> 1));
    lat += dlat;

    shift = 0;
    result = 0;
    do {
        b = encoded.charAt(index++) - 63;
        result |= (b & 0x1f) << shift;
        shift += 5;
    } while (b >= 0x20);
    int dlng = ((result & 1) != 0 ? ~(result >> 1) : (result >> 1));
    lng += dlng;

    LatLng p = new LatLng((((double) lat / 1E5)),
        (((double) lng / 1E5)));
    poly.add(p);
}

return poly;
}
}

```

```
package com.custom.co2.utils;

import java.io.Serializable;

public class DirectionUtils implements Serializable {
    public String getDistance() {
        return distance;
    }

    public void setDistance(String distance) {
        this.distance = distance;
    }

    public String getDuration() {
        return duration;
    }

    public void setDuration(String duration) {
        this.duration = duration;
    }

    public void setStAddress(String stAddress) {
        this.stAddress = stAddress;
    }

    public void setEndAddress(String endAddress) {
        this.endAddress = endAddress;
    }

    public String distance;
    public String duration;
    public String stAddress;
    public String endAddress;
}
```

```
package com.custom.co2.utils;

import android.app.Activity;
import android.content.Context;
import android.content.res.Resources;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.view.animation.TranslateAnimation;
import android.view.inputmethod.InputMethodManager;

import androidx.recyclerview.widget.RecyclerView;

import com.custom.co2.R;
import com.google.android.gms.maps.model.LatLng;

import static java.lang.Math.atan;

public class MapUtils {

    public static int getScreenWidth() {
        return Resources.getSystem().getDisplayMetrics().widthPixels;
    }

    public static void hideKeyboard(Activity activity) {
        InputMethodManager imm = (InputMethodManager) activity.getSystemService(Activity.INPUT_METHOD_SERVICE);
        View view = activity.getCurrentFocus();
        if (view == null) {
            view = new View(activity);
        }
        if (imm != null) {
            imm.hideSoftInputFromWindow(view.getWindowToken(), 0);
        }
    }

    public static void createBottomUpAnimation(Context context, View view, Animation.AnimationListener listener) {
        Animation slideUpAnimation = AnimationUtils.loadAnimation(context,
                R.anim.slide_up_animation);
        if (listener != null) {
            slideUpAnimation.setAnimationListener(listener);
        }
        view.startAnimation(slideUpAnimation);
    }

    public static void createTopDownAnimation(Context context, View view, Animation.AnimationListener listener) {
        Animation slideDownAnimation = AnimationUtils.loadAnimation(context,
                R.anim.slide_down_animation);
        if (listener != null) {
            slideDownAnimation.setAnimationListener(listener);
        }
        view.startAnimation(slideDownAnimation);
    }

    public static void SlideToAbove(final RecyclerView rl_footer, Animation.AnimationListener listener) {
        Animation slide = null;
        slide = new TranslateAnimation(Animation.RELATIVE_TO_SELF, 0.0f,
                Animation.RELATIVE_TO_SELF, 0.0f, Animation.RELATIVE_TO_SELF,
                0.0f, Animation.RELATIVE_TO_SELF, -5.0f);

        slide.setDuration(600);
        slide.setFillAfter(true);
    }
}
```

```

slide.setFillEnabled(true);
rl_footer.startAnimation(slide);
if (listener != null) {
    slide.setAnimationListener(listener);
}
}

public static void slideUp(Context context, View view, Animation.AnimationListener listener) {
    view.setVisibility(View.VISIBLE);
    Animation animate = AnimationUtils.loadAnimation(context, R.anim.slide_down);
    animate.setFillAfter(true);
    if (listener != null) {
        animate.setAnimationListener(listener);
    }
    view.startAnimation(animate);
}

public static Bitmap getOriginDestinationMarkerBitmap() {
    int height = 20;
    int width = 20;
    Bitmap bitmap = Bitmap.createBitmap(height, width, Bitmap.Config.RGB_565);
    Canvas canvas = new Canvas(bitmap);
    Paint paint = new Paint();
    paint.setColor(Color.BLACK);
    paint.setStyle(Paint.Style.FILL);
    paint.setAntiAlias(true);
    canvas.drawRect(0F, 0F, width, height, paint);
    return bitmap;
}

public static Float getRotation(LatLng start, LatLng end) {
    Double latDifference = Math.abs(start.latitude - end.latitude);
    Double lngDifference = Math.abs(start.longitude - end.longitude);
    float rotation = -1F;
    if(start.latitude < end.latitude && start.longitude < end.longitude) {
        rotation = (float) Math.toDegrees(Math.atan(lngDifference / latDifference));
    } else if(start.latitude >= end.latitude && start.longitude < end.longitude) {
        rotation = (float) (90 - Math.toDegrees(Math.atan(lngDifference / latDifference)) + 90);
    } else if(start.latitude >= end.latitude && start.longitude >= end.longitude) {
        rotation = (float) (Math.toDegrees(Math.atan(lngDifference / latDifference)) + 180);
    } else if(start.latitude < end.latitude && start.longitude >= end.longitude) {
        rotation = (float) (90 - Math.toDegrees(Math.atan(lngDifference / latDifference)) + 270);
    }
    return rotation;
}
}

```

```
package com.custom.co2.utils;

import com.google.android.gms.maps.model.LatLng;
import com.google.maps.android.clustering.ClusterItem;

public class MyItem implements ClusterItem {
    private final LatLng mPosition;
    private String mTitle = null;
    private String mSnippet = null;

    public MyItem(double lat, double lng) {
        mPosition = new LatLng(lat, lng);
    }

    public MyItem(double lat, double lng, String title, String snippet) {
        mPosition = new LatLng(lat, lng);
        mTitle = title;
        mSnippet = snippet;
    }

    @Override
    public LatLng getPosition() {
        return mPosition;
    }

    @Override
    public String getTitle() {
        return mTitle;
    }

    @Override
    public String getSnippet() {
        return mSnippet;
    }
}
```

```
package com.custom.co2.utils;

import com.google.android.gms.maps.model.LatLng;

public class Place {
    public Place()
    {}
    public Place(LatLng src, LatLng des)
    {
        this.srcLat = src.latitude;
        this.destLat = des.latitude;
        this.srcLon = src.longitude;
        this.destLon = des.longitude;
    }
    public double srcLat;

    public double getSrcLat() {
        return srcLat;
    }

    public double getSrcLon() {
        return srcLon;
    }

    public double getDestLat() {
        return destLat;
    }

    public double getDestLon() {
        return destLon;
    }

    public LatLng getSrcLatLng(){return new LatLng(srcLat,srcLon);}

    public LatLng getDesLatLng(){return new LatLng(destLat,destLon);}

    public double srcLon;
    public double destLat;
    public double destLon;

    public CharSequence getSrcAddress() {
        return srcaddress;
    }

    public void setSrcAddress(CharSequence address) {
        this.srcaddress = address;
    }

    public CharSequence srcaddress;

    public CharSequence getDestAddress() {
        return destaddress;
    }

    public void setDestAddress(CharSequence destaddress) {
        this.destaddress = destaddress;
    }

    public CharSequence destaddress;
}
```

```
package com.custom.co2.utils;
```

```
public interface Placelistener {  
    public void onPlace(Place place);  
}
```

```
1 package com.custom.co2.utils;  
2  
3 public interface ProgressListener {  
4     public void onProgressChanged(boolean state);  
5 }
```

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <translate>
        android:duration="1500"
        android:fromYDelta="5"
        android:toYDelta="50%" />
</set>
```

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android" android:interpolator="@android:anim/linear_interpolator">
    <scale>
        android:duration="800"
        android:fromXScale="1.0"
        android:fromYScale="0.0"
        android:toXScale="1.0"
        android:toYScale="1.0" />
</set>
```

```
<?xml version="1.0" encoding="utf-8"?>
<set>
    <translate xmlns:android="http://schemas.android.com/apk/res/android"
        android:fromXDelta="0"
        android:toXDelta="100%"
        android:interpolator="@android:anim/accelerate_interpolator"
        android:duration="@android:integer/config_mediumAnimTime"/>
</set>
```

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <translate>
        android:duration="1500"
        android:fromYDelta="90%"
        android:toYDelta="0" />
</set>
```

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item>
        <shape android:shape="oval">
            <solid android:color="#818181"/>
            <size android:width="15dp" android:height="15dp"/>
        </shape>
    </item>
</selector>
```

```
<?xml version="1.0" encoding="utf-8"?>
<animated-vector xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:aapt="http://schemas.android.com/aapt"
    android:drawable="@drawable/vd_line">
    <target android:name="path_line">
        <aapt:attr name="android:animation">
            <set android:ordering="sequentially">

                <set>
                    <objectAnimator
                        android:duration="800"
                        android:interpolator="@android:interpolator/fast_out_slow_in"
                        android:propertyName="trimPathOffset"
                        android:repeatCount="0"
                        android:valueFrom="0"
                        android:valueTo="0.8" />

                <set android:ordering="sequentially">
                    <objectAnimator
                        android:duration="400"
                        android:interpolator="@android:interpolator/fast_out_slow_in"
                        android:propertyName="trimPathEnd"
                        android:repeatCount="0"
                        android:valueFrom="0.01"
                        android:valueTo="0.2" />

                    <objectAnimator
                        android:duration="400"
                        android:interpolator="@android:interpolator/fast_out_slow_in"
                        android:propertyName="trimPathStart"
                        android:repeatCount="0"
                        android:valueFrom="0"
                        android:valueTo="0.19" />
                
```

```
</set>
</aapt:attr>
</target>
</animated-vector>
```

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item>
        <shape android:shape="rectangle">

            <corners android:radius="4dp" />

            <solid android:color="@color/colorPrimaryDark" />
        </shape>
    </item>
</selector>
```

```
<?xml version="1.0" encoding="utf-8"?>
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:gravity="bottom">
        <shape android:shape="rectangle" />
    </item>
    <item
        android:width="720dp"
        android:height="500dp"
        android:gravity="bottom|center_horizontal"
        android:top="-40dp">
        <shape android:shape="oval">
            <solid android:color="@color/colorPrimary" />
        </shape>
    </item>
    <item
        android:height="720dp"
        android:bottom="250dp"
        android:gravity="bottom">
        <shape android:shape="rectangle">
            <solid android:color="@color/colorPrimary" />
        </shape>
    </item>
</layer-list>
```

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">

    <item>
        <shape android:shape="rectangle">
            <corners android:radius="24dp" />
            <solid android:color="@color/colorPrimary" />
        </shape>
    </item>
</selector>
```

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item>
        <shape android:shape="rectangle">
            <corners android:radius="4dp" />
            <stroke android:width="2dp" android:color="@color/colorPrimaryDark" />
            <solid android:color="#09CF6D" />
        </shape>
    </item>
</selector>
```

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item>
        <shape android:shape="rectangle">

            <corners android:radius="4dp" />

            <solid android:color="#710c04" />
        </shape>
    </item>
</selector>
```

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item>
        <shape android:shape="rectangle">
            <corners android:radius="4dp" />
            <solid android:color="#f09090" />
        </shape>
    </item>
</selector>
```

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item>
        <shape android:shape="rectangle">
            <corners android:radius="4dp" />
            <solid android:color="#f04848" />
        </shape>
    </item>
</selector>
```

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item>
        <shape android:shape="rectangle">
            <corners android:radius="4dp" />
            <stroke android:width="2dp" android:color="@color/colorPrimaryDark" />
            <solid android:color="@color/white"/>
        </shape>
    </item>
</selector>
```

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@drawable/bg_blue_btn" android:state_pressed="true" /> <!-- pressed -->
    <item android:drawable="@drawable/bg_blue_btn" android:state_focused="true" /> <!-- focused -->
    <item android:drawable="@drawable/bg_blue_btn" android:state_selected="true" /> <!-- hovered -->
    <item android:drawable="@drawable/bg_red_btn" /> <!-- default -->
</selector>
```

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@drawable/bg_blue_btn" android:state_pressed="true" /> <!-- pressed -->
    <item android:drawable="@drawable/bg_blue_btn" android:state_focused="true" /> <!-- focused -->
    <item android:drawable="@drawable/bg_blue_btn" android:state_selected="true" /> <!-- hovered -->
    <item android:drawable="@drawable/bg_maroon_btn" /> <!-- default -->
</selector>
```

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@drawable/bg_blue_btn" android:state_pressed="true" /> <!-- pressed -->
    <item android:drawable="@drawable/bg_blue_btn" android:state_focused="true" /> <!-- focused -->
    <item android:drawable="@drawable/bg_blue_btn" android:state_selected="true" /> <!-- hovered -->
    <item android:drawable="@drawable/bg_pink_btn" /> <!-- default -->
</selector>
```

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@drawable/bg_blue_btn" android:state_pressed="true" /> <!-- pressed -->
    <item android:drawable="@drawable/bg_blue_btn" android:state_focused="true" /> <!-- focused -->
    <item android:drawable="@drawable/bg_blue_btn" android:state_selected="true" /> <!-- hovered -->
    <item android:drawable="@drawable/bg_white_btn" /> <!-- default -->
</selector>
```

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@drawable/bg_blue_btn" android:state_pressed="true" /> <!-- pressed -->
    <item android:drawable="@drawable/bg_blue_btn" android:state_focused="true" /> <!-- focused -->
    <item android:drawable="@drawable/bg_blue_btn" android:state_selected="true" /> <!-- hovered -->
    <item android:drawable="@drawable/bg_green_btn" /> <!-- default -->
</selector>
```

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:color="@color/white" android:state_pressed="true" /> <!-- pressed -->
    <item android:color="@color/white" android:state_focused="true" /> <!-- focused -->
    <item android:color="@color/white" android:state_selected="true" /> <!-- hovered -->
    <item android:color="@color/colorPrimaryDark" /> <!-- default -->
</selector>
```

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item>
        <shape android:shape="oval">
            <solid android:color="#e91313"/>
            <size android:width="15dp" android:height="15dp"/>
        </shape>
    </item>
</selector>
```

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle" >

    <corners android:radius="10dp" />
    <solid android:color="@color/colorWhite" />

    <stroke
        android:width="3dp"
        android:color="@color/colorButton" />

</shape>
```

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"  
    android:width="32dp"  
    android:height="32dp"  
    android:viewportWidth="24.0"  
    android:viewportHeight="24.0">  
    <path  
        android:fillColor="#FFFFFF"  
        android:pathData="M21,11H6.83l3.58,-3.59L9,6l-6,6 6,6 1.41,-1.41L6.83,13H21z"/>  
</vector>
```

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"  
    android:width="24dp"  
    android:height="24dp"  
    android:viewportWidth="24.0"  
    android:viewportHeight="24.0">  
    <path  
        android:fillColor="#FF000000"  
        android:pathData="M21,11H6.83l3.58,-3.59L9,6l-6,6 6,6 1.41,-1.41L6.83,13H21z"/>  
</vector>
```

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"  
    android:width="512dp"  
    android:height="512dp"  
    android:tint="?attr/colorControlNormal"  
    android:viewportWidth="60"  
    android:viewportHeight="60">  
    <path  
        android:fillColor="@android:color/white"  
        android:pathData="M13,3c-4.97,0 -9,4.03 -9,9L1,12l3.89,3.89 0.07,0.14L9,12L6,12c0,-3.87 3.13,-7 7,-7s7,3.13 7,7 -3.13,7  
-7,7c-1.93,0 -3.68,-0.79 -4.94,-2.06l-1.42,1.42C8.27,19.99 10.51,21 13,21c4.97,0 9,-4.03 9,-9s-4.03,-9 -9,-9zM12,8v5l4.28,2.54  
0.72,-1.21 -3.5,-2.08L13.5,8L12,8z" />  
</vector>
```

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"  
    android:width="32dp"  
    android:height="32dp"  
    android:viewportWidth="496.628"  
    android:viewportHeight="496.628">  
    <path  
        android:fillColor="#FF000000"  
        android:pathData="M141.871,417.141c-11.752,9.038 -26.278,14.631 -42.217,14.631c-38.411,0 -69.667,-31.256 -69.667,-  
69.667c0,-38.419 31.256,-69.676 69.667,-69.676c28.99,0 53.853,17.825 64.336,43.067l21.816,-22.626c-17.18,-29.986 -  
49.118,-50.427 -86.152,-50.427C44.608,262.443 0,307.061 0,362.105c0,55.036 44.608,99.654 99.654,99.654c25.898,0  
49.265,-10.122 66.991,-26.326c-6.803,-2.283 -13.256,-5.905 -18.77,-11.215C145.6,422.02 143.637,419.619 141.871,417.141z"  
/>  
    <path  
        android:fillColor="#FF000000"  
        android:pathData="M293.191,263.204l-52.769,-30.65c23.533,-14.729 14.807,-9.274 38.341,-24.004l-0.741,7.497c-  
1.27,12.759 8.133,24.435 19.971,26.834c0.283,0.058 92.635,15.755 96.003,15.755c11.997,0 22.589,-8.668 24.617,-  
20.899c2.265,-13.617 -6.939,-26.484 -20.556,-28.737l-67.989,-11.295l3.233,-32.445c3.259,-20.733 4.674,-27.234 -3.067,-  
39.603l-9.732,-15.559c-12.573,-20.08 -39.084,-26.22 -59.212,-13.618l-106.515,66.689c-26.161,16.38 -38.919,50.917 -  
17.561,76.323c8.344,9.928 89.47,59.781 89.47,59.781l-65.703,68.153c-11.499,11.918 -11.157,30.905  
0.771,42.394c5.818,5.611 13.315,8.403 20.811,8.403c7.859,0 15.707,-3.064 21.583,-9.176l95.564,-99.105C313.192,295.992  
310.03,272.984 293.191,263.204z" />  
    <path  
        android:fillColor="#FF000000"  
        android:pathData="M405.134,127.072c23.36,-14.632 30.437,-45.42 15.813,-68.769c-14.621,-23.369 -45.41,-30.446 -  
68.767,-15.823c-23.35,14.623 -30.427,45.4 -15.813,68.749C350.987,134.599 381.776,141.686 405.134,127.072z" />  
    <path  
        android:fillColor="#FF000000"  
        android:pathData="M396.973,262.443c-55.619,0 -99.654,45.888 -99.654,99.662c0,55.036 44.61,99.654  
99.654,99.654c55.034,0 99.655,-44.618 99.655,-99.654C496.628,307.061 452.007,262.443  
396.973,262.443zM396.973,431.772c-38.411,0 -69.667,-31.256 -69.667,-69.667c0,-38.419 31.257,-69.676 69.667,-  
69.676c38.412,0 69.668,31.257 69.668,69.676C466.642,400.516 435.385,431.772 396.973,431.772z" />  
</vector>
```

```
<vector android:height="128dp" android:viewportHeight="640"  
        android:viewportWidth="640" android:width="128dp" xmlns:android="http://schemas.android.com/apk/res/android">  
    <path android:fillColor="#FF000000" android:pathData="m615.5,277.332 l-30.898,-68.199c-11.141,-24.223 -35.344,-39.762 -  
62.004,-39.801h-443.297c-32.738,0.035 -59.266,26.563 -59.301,59.305v148.195c0,17.949 14.551,32.5  
32.5,32.5h48.5c4.734,23.309 25.219,40.051 49,40.051s44.266,-16.742 49,-40.051h242c4.734,23.309 25.219,40.051  
49,40.051s44.266,-16.742 49,-40.051h53.203c12.348,-0.004 23.219,-8.148 26.699,-20 0.723,-2.539 1.09,-5.16 1.098,-7.797v-  
83.5c-0.004,-7.145 -1.539,-14.203 -4.5,-20.703zM70,289.332c-5.523,0 -10,-4.477 -10,-10v-80c0,-5.52 4.477,-10 10,-  
10h70c5.523,0 10,4.48 10,10v80c0,5.523 -4.477,10 -10,10zM150,429.332c-16.566,0 -30,-13.43 -30,-30 0,-16.566 13.434,-30  
30,-30s30,13.434 30,30c-0.047,16.551 -13.453,29.953 -30,30zM260,279.332c0,5.523 -4.477,10 -10,10h-70c-5.523,0 -10,-4.477  
-10,-10v-80c0,-5.52 4.477,-10 10,-10h70c5.523,0 10,4.48 10,10zM370,279.332c0,5.523 -4.477,10 -10,10h-70c-5.523,0 -10,-4.477  
-10,-10v-80c0,-5.52 4.477,-10 10,-10h70c5.523,0 10,4.48 10,10v80c0,5.523 -4.477,10 -10,10zM490,429.332c-16.566,0 -30,-13.43 -30,-30 0,-16.566  
13.434,-30 30,-30s30,13.434 30,30c-0.047,16.551 -13.453,29.953 -30,30zM509.199,289.332c-5.184,-0.461 -9.168,-4.793 -  
9.199,-10v-80.086c0,-5.473 4.438,-9.914 9.914,-9.914h12.684c18.824,0.051 35.914,11.012  
43.805,28.102l30.898,68.199c1.613,3.555 2.5,7.398 2.602,11.297zM509.199,289.332"/>  
</vector>
```

```
<vector android:height="128.00128dp" android:viewportHeight="99.443"
    android:viewportWidth="99.442" android:width="128dp" xmlns:android="http://schemas.android.com/apk/res/android">
    <path android:fillColor="#FF000000" android:pathData="M19.097,54.071c-4.175,0 -7.561,3.383 -7.561,7.56c0,0.324
0.026,0.641 0.066,0.951c0.469,3.729 3.642,6.611 7.494,6.611c3.959,0 7.202,-3.042 7.53,-6.916c0.018,-0.214 0.033,-0.428
0.033,-0.646C26.66,57.454 23.275,54.071 19.097,54.071zM15.31,58.782l1.543,1.543c-0.121,0.206 -0.214,0.429 -0.274,0.665h-
2.174C14.518,60.165 14.833,59.415 15.31,58.782zM14.397,62.298h2.189c0.062,0.233 0.153,0.454
0.274,0.656L15.314,64.5C14.838,63.871 14.513,63.119 14.397,62.298zM18.443,66.328c-0.818,-0.112 -1.564,-0.434 -2.193,-
0.908l1.537,-1.538c0.202,0.118 0.424,0.205 0.656,0.266V66.328zM18.443,59.116c-0.238,0.062 -0.461,0.157 -0.668,0.279l-
1.541,-1.541c0.633,-0.48 1.385,-0.8 2.209,-0.913V59.116zM19.752,56.941c0.826,0.113 1.577,0.433 2.209,0.914l-1.54,1.54c-
0.207,-0.122 -0.43,-0.218 -0.669,-0.279V56.941zM19.752,66.328v-2.182c0.233,-0.061 0.454,-0.147 0.657,-
0.268l1.538,1.54C21.317,65.894 20.572,66.214 19.752,66.328zM22.885,64.504l-1.551,-1.551c0.12,-0.203 0.22,-0.42 0.282,-
0.655h2.172C23.673,63.119 23.364,63.875 22.885,64.504zM21.617,60.99c-0.06,-0.236 -0.153,-0.459 -0.274,-0.665l1.543,-
1.543c0.478,0.633 0.792,1.383 0.905,2.208H21.617z"/>
    <path android:fillColor="#FF000000" android:pathData="M83.965,54.071c-4.176,0 -7.561,3.383 -7.561,7.56c0,0.324
0.025,0.641 0.065,0.951c0.468,3.729 3.643,6.611 7.494,6.611c3.958,0 7.201,-3.042 7.53,-6.916c0.018,-0.214 0.031,-0.428
0.031,-0.646C91.526,57.454 88.142,54.071 83.965,54.071zM80.177,58.782l1.543,1.543c-0.12,0.206 -0.214,0.429 -
0.273,0.665h-2.175C79.385,60.165 79.7,59.415 80.177,58.782zM79.265,62.298h2.19c0.062,0.233 0.152,0.454
0.272,0.656L80.182,64.5C79.705,63.871 79.38,63.119 79.265,62.298zM83.31,66.328c-0.818,-0.112 -1.563,-0.434 -2.192,-
0.908l1.537,-1.538c0.201,0.118 0.424,0.205 0.655,0.266V66.328zM83.31,59.116c-0.237,0.062 -0.461,0.157 -
0.669,0.279L81.1,57.854c0.634,-0.48 1.385,-0.8 2.209,-0.913L83.31,59.116L83.31,59.116zM84.62,56.941c0.824,0.113
1.576,0.433 2.209,0.914l-1.541,1.54c-0.207,-0.122 -0.431,-0.218 -0.668,-0.279V56.941zM84.62,66.328v-2.182c0.231,-0.061
0.454,-0.147 0.655,-0.268l1.538,1.54C86.184,65.894 85.438,66.214 84.62,66.328zM87.752,64.504l-1.551,-1.551c0.12,-0.203
0.22,-0.42 0.281,-0.655h2.174C88.54,63.119 88.23,63.875 87.752,64.504zM86.483,60.99c-0.06,-0.236 -0.152,-0.459 -0.274,-
0.665l1.542,-1.543c0.478,0.633 0.792,1.383 0.906,2.208H86.483z"/>
    <path android:fillColor="#FF000000" android:pathData="M99.091,47.939c-0.056,-1.67 -0.516,-3.301 -1.339,-4.754l-5.43,-
9.581c-0.89,-1.569 -2.521,-2.573 -4.322,-2.664c-9.1,-0.456 -37.002,-1.618 -45.786,0.744C36.272,33.283 21.278,43.14
21.278,43.14S4.781,45.695 1.634,53.219c0,0 -1.358,0.793 -1.605,2.826c-0.127,1.046 0.183,2.634 0.525,3.965c0.375,1.456
1.582,2.552 3.067,2.783l7.16,1.122c-0.107,-0.391 -0.196,-0.788 -0.248,-1.198c-0.045,-0.354 -0.075,-0.716 -0.075,-1.087c0,-
4.763 3.875,-8.637 8.639,-8.637c4.765,0 8.64,3.874 8.64,8.637c0,0.249 -0.016,0.493 -0.036,0.735c-0.072,0.844 -0.268,1.651 -
0.567,2.408l0.842,0.045l47.568,-1.287c-0.061,-0.268 -0.109,-0.538 -0.145,-0.814c-0.046,-0.354 -0.074,-0.716 -0.074,-1.087c0,-
4.763 3.875,-8.637 8.638,-8.637c4.765,0 8.64,3.874 8.64,8.637c0,0.249 -0.016,0.493 -0.037,0.735c-0.013,0.16 -0.041,0.315 -
0.062,0.473L96.609,62c1.693,-0.346 2.891,-1.86 2.831,-3.589L99.091,47.939zM71.715,32.71l1.093,10.911l-
16.774,0.686V32.655C60.938,32.542 66.536,32.593 71.715,32.71zM29.387,45.087l-1.659,0.093c-0.451,0.025 -0.864,-0.249 -
1.016,-0.675c-0.152,-0.424 -0.005,-0.897 0.358,-1.164c0.975,-0.712 2.169,-1.563 3.499,-2.462v2.784C29.759,44.348
29.387,45.087 29.387,45.087zM33.498,42.533c-0.105,0.004 -0.191,0.03 -0.291,0.04V39.15c3.382,-2.144 7.215,-4.273 10.511,-
5.34c1.5,-0.485 4.236,-0.795 7.636,-0.98v11.668l-14.412,0.589C36.942,45.087 36.442,42.423 33.498,42.533zM91.768,41.475c-
0.503,0.874 -1.419,1.429 -2.426,1.471l77.49,43.43l-1.062,-10.594c4.898,0.149 8.99,0.333 11.063,0.445c0.959,0.051
1.824,0.604 2.271,1.454l2.057,3.903C92.29,39.531 92.27,40.603 91.768,41.475z"/>
</vector>
```

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"  
    android:width="32dp"  
    android:height="32dp"  
    android:tint="#FFFFFF"  
    android:viewportWidth="24"  
    android:viewportHeight="24">  
    <path  
        android:fillColor="@android:color/white"  
        android:pathData="M4.25,5.61C6.27,8.2 10,13 10,13v6c0,0.55 0.45,1 1,1h2c0.55,0 1,-0.45 1,-1v-6c0,0 3.72,-4.8 5.74,-  
7.39C20.25,4.95 19.78,4 18.95,4H5.04C4.21,4 3.74,4.95 4.25,5.61z" />  
</vector>
```

```
<vector android:height="128dp" android:viewportHeight="46.942"  
        android:viewportWidth="46.942" android:width="128dp" xmlns:android="http://schemas.android.com/apk/res/android">  
    <path android:fillColor="#030104" android:pathData="M25.789,24.854h0.239L43.8,30.71c0,0 1.299,-3.445 -0.414,-4.601c-  
1.713,-1.153 -5.233,-3.524 -5.233,-3.524l-5.23,-3.521l-6.895,-4.643h-0.07l-0.393,-10.91c0,0 0.089,-3.511 -2.497,-3.511c-2.588,0  
-2.87,3.08 -2.87,3.08l-0.521,11.791l-6.017,4.202l-5.24,3.666c0,0 -3.024,1.729 -4.825,3.368c-1.801,1.644 -0.415,5.19 -  
0.415,5.19l16.662,-6.18l1.48,15.146c0,0.356 0.012,0.7 0.045,1.024l15.95,44.99c0,0 -1.418,1.952 0,1.952c1.416,0 12.69,0  
13.828,0c1.136,0 0.008,-1.78 0.008,-1.78l-5.951,-3.901c0.034,-0.19 0.054,-0.386 0.054,-0.585L25.789,24.854z"/>  
</vector>
```

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"  
    android:width="512dp"  
    android:height="512dp"  
    android:viewportWidth="512"  
    android:viewportHeight="512">  
<path  
    android:fillColor="#FF000000"  
    android:pathData="M503.401,228.884l-43.253,-39.411V58.79c0,-8.315 -6.741,-15.057 -15.057,-15.057H340.976c-8.315,0 -15.057,6.741 -15.057,15.057v8.374l-52.236,-47.597c-10.083,-9.189 -25.288,-9.188 -35.367,-0.001L8.598,228.885c-8.076,7.36 -10.745,18.7 -6.799,28.889c3.947,10.189 13.557,16.772 24.484,16.772h36.689v209.721c0,8.315 6.741,15.057  
15.057,15.057h125.913c8.315,0 15.057,-6.741 15.057,-15.057V356.931h293v127.337c0,8.315 6.741,15.057  
15.057,15.057h125.908c8.315,0 15.057,-6.741 15.056,-15.057V274.547h36.697c10.926,0 20.537,-6.584 24.484,-  
16.772C514.147,247.585 511.479,236.245 503.401,228.884zM433.965,244.433c-8.315,0 -15.057,6.741 -  
15.057,15.057v209.721h-95.793V341.874c0,-8.315 -6.742,-15.057 -15.057,-15.057H203.942c-8.315,0 -15.057,6.741 -  
15.057,15.057v127.337h-95.8V259.49c0,-8.315 -6.741,-15.057 -15.057,-15.057H36.245l219.756,-  
200.24l74.836,68.191c4.408,4.016 10.771,5.051 16.224,2.644c5.454,-2.41 8.973,-7.812 8.973,-  
13.774V73.847h74.002v122.276c0,4.237 1.784,8.276 4.916,11.13l40.803,37.18H433.965" />  
</vector>
```

```
<?xml version="1.0" encoding="utf-8"?>
<vector android:height="108dp"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="108dp"
    android:viewportWidth="108"
    android:viewportHeight="108">
    <path
        android:fillColor="#3DDC84"
        android:pathData="M0,0h108v108h-108z" />
    <path
        android:fillColor="#00000000"
        android:pathData="M9,0L9,108"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M19,0L19,108"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M29,0L29,108"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M39,0L39,108"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M49,0L49,108"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M59,0L59,108"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M69,0L69,108"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M79,0L79,108"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M89,0L89,108"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M99,0L99,108"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M0,9L108,9"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
```

```
    android:fillColor="#00000000"
    android:pathData="M0,19L108,19"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M0,29L108,29"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M0,39L108,39"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M0,49L108,49"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M0,59L108,59"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M0,69L108,69"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M0,79L108,79"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M0,89L108,89"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M0,99L108,99"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M19,29L89,29"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M19,39L89,39"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M19,49L89,49"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M19,59L89,59"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
```

```
    android:fillColor="#00000000"
    android:pathData="M19,69L89,69"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M19,79L89,79"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M29,19L29,89"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M39,19L39,89"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M49,19L49,89"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M59,19L59,89"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M69,19L69,89"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M79,19L79,89"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
</vector>
```

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"  
    android:width="24dp"  
    android:height="24dp"  
    android:viewportWidth="24"  
    android:viewportHeight="24">  
    <path  
        android:fillColor="#000000"  
        android:pathData="M11,7L9.6,8.4l2.6,2.6H2v2h10.2l-2.6,2.6L11,17l5,-5L11,7zM20,19h-8v2h8c1.1,0 2,-0.9 2,-2V5c0,-1.1 -0.9,-2 -2,-2h-8v2h8V19z" />  
</vector>
```

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"  
    android:width="512dp"  
    android:height="512.001dp"  
    android:viewportWidth="512"  
    android:viewportHeight="512.001">  
<path  
    android:fillColor="#FF000000"  
    android:pathData="m252.352,246.633c33.883,0 63.219,-12.152 87.195,-36.129 23.969,-23.973 36.125,-53.305 36.125,-  
87.191 0,-33.875 -12.152,-63.211 -36.129,-87.191 -23.977,-23.969 -53.313,-36.121 -87.191,-36.121 -33.887,0 -63.219,12.152 -  
87.191,36.125s-36.129,53.309 -36.129,87.188c0,33.887 12.156,63.223 36.129,87.195 23.98,23.969 53.316,36.125  
87.191,36.125zM186.379,57.34c18.395,-18.395 39.973,-27.336 65.973,-27.336 25.996,0 47.578,8.941 65.977,27.336  
18.395,18.398 27.34,39.98 27.34,65.973 0,26 -8.945,47.578 -27.34,65.977 -18.398,18.398 -39.98,27.34 -65.977,27.34 -25.992,0  
-47.57,-8.945 -65.973,-27.34 -18.395,-18.395 -27.344,-39.977 -27.344,-65.977 0,-25.992 8.945,-47.574 27.344,-  
65.973zM186.379,57.34"/>  
<path  
    android:fillColor="#FF000000"  
    android:pathData="m468.129,393.703c-0.691,-9.977 -2.09,-20.859 -4.148,-32.352 -2.078,-11.578 -4.754,-22.523 -7.957,-  
32.527 -3.313,-10.34 -7.809,-20.551 -13.375,-30.336 -5.77,-10.156 -12.551,-19 -20.16,-26.277 -7.957,-7.613 -17.699,-13.734 -  
28.965,-18.199 -11.227,-4.441 -23.668,-6.691 -36.977,-6.691 -5.227,0 -10.281,2.145 -20.043,8.5 -6.008,3.918 -13.035,8.449 -  
20.879,13.461 -6.707,4.273 -15.793,8.277 -27.016,11.902 -10.949,3.543 -22.066,5.34 -33.043,5.34 -10.969,0 -22.086,-1.797 -  
33.043,-5.34 -11.211,-3.621 -20.301,-7.625 -26.996,-11.898 -7.77,-4.965 -14.801,-9.496 -20.898,-13.469 -9.754,-6.355 -14.809,-  
8.5 -20.035,-8.5 -13.313,0 -25.75,2.254 -36.973,6.699 -11.258,4.457 -21.004,10.578 -28.969,18.199 -7.609,7.281 -  
14.391,16.121 -20.156,26.273 -5.559,9.785 -10.059,19.992 -13.371,30.34 -3.199,10.004 -5.875,20.945 -7.953,32.523 -  
2.063,11.477 -3.457,22.363 -4.148,32.363 -0.68,9.777 -1.023,19.953 -1.023,30.234 0.26,7.277 8.496,48.363 25.25,64.32  
16.547,15.746 38.438,23.73 65.066,23.73h246.531c26.621,0 48.512,-7.984 65.063,-23.73 16.758,-15.945 25.254,-37.59  
25.254,-64.324 -0.004,-10.316 -0.352,-20.492 -1.035,-30.242zM423.223,466.531c-10.934,10.406 -25.449,15.465 -  
44.379,15.465h-246.527c-18.934,0 -33.449,-5.059 -44.379,-15.461 -10.723,-10.207 -15.934,-24.141 -15.934,-42.586 0,-9.594  
0.316,-19.066 0.949,-28.16 0.617,-8.922 1.879,-18.723 3.75,-29.137 1.848,-10.285 4.199,-19.938 6.996,-28.676 2.684,-8.379  
6.344,-16.676 10.883,-24.668 4.332,-7.617 9.316,-14.152 14.816,-19.418 5.145,-4.926 11.629,-8.957 19.27,-11.98 7.066,-2.797  
15.008,-4.328 23.629,-4.559 1.051,0.559 2.922,1.625 5.953,3.602 6.168,4.02 13.277,8.605 21.137,13.625 8.859,5.648  
20.273,10.75 33.91,15.152 13.941,4.508 28.16,6.797 42.273,6.797 14.113,0 28.336,-2.289 42.27,-6.793 13.648,-4.41 25.059,-  
9.508 33.93,-15.164 8.043,-5.141 14.953,-9.594 21.121,-13.617 3.031,-1.973 4.902,-3.043 5.953,-3.602 8.625,0.23 16.566,1.762  
23.637,4.559 7.637,3.023 14.121,7.059 19.266,11.98 5.5,5.262 10.484,11.797 14.816,19.422 4.543,7.988 8.207,16.289  
10.887,24.66 2.801,8.75 5.156,18.398 7.28.676 1.867,10.434 3.133,20.238 3.75,29.145v0.008c0.637,9.059 0.957,18.527  
0.961,28.148 -0.004,18.449 -5.215,32.379 -15.938,42.582zM423.223,466.531"/>  
</vector>
```

```
<vector android:height="128dp" android:viewportHeight="582.675"  
        android:viewportWidth="582.675" android:width="128dp" xmlns:android="http://schemas.android.com/apk/res/android">  
    <path android:fillColor="#FF000000" android:pathData="M518.288,545.7l-85.425,-155.55c24.225,-3.825 42.074,-25.5  
42.074,-49.726v-229.5C474.937,49.725 425.212,0 364.012,0h-145.35c-61.2,0 -112.2,49.725 -112.2,112.2v228.225c0,25.5  
17.85,45.9 42.075,49.726L64.387,545.7c-6.375,12.75 -2.55,28.05 10.2,34.425c3.825,2.55 7.65,2.55 11.475,2.55c8.925,0  
17.85,-5.1 21.675,-12.75l7.65,-12.75h351.9l7.649,12.75c5.101,8.925 12.75,12.75 21.676,12.75c3.824,0 7.649,-1.274 11.475,-  
2.55C520.837,573.75 524.663,558.45 518.288,545.7zM398.437,430.95h-214.2l21.675,-  
39.525h170.85L398.437,430.95zM394.613,345.525c-20.4,0 -36.976,-16.575 -36.976,-36.976c0,-20.4 16.575,-36.975 36.976,-  
36.975c20.399,0 36.975,16.575 36.975,36.975C431.587,327.675 415.012,345.525 394.613,345.525zM151.087,126.225c0,-  
39.525 31.875,-71.4 70.125,-71.4h138.975c38.25,0 71.4,31.875 71.4,71.4v61.2c0,14.025 -11.475,24.225 -25.5,24.225h-229.5c-  
14.025,0 -24.225,-11.475 -24.225,-24.225L151.087,126.225L151.087,126.225zM188.062,270.3c20.4,0 36.975,16.575  
36.975,36.976c0,20.399 -16.575,36.975 -36.975,36.975s-36.975,-16.575 -36.975,-36.975C151.087,286.875 167.663,270.3  
188.062,270.3zM135.788,518.925l28.05,-49.725h255l28.05,49.725H135.788z"/>  
</vector>
```

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"  
    android:width="512dp"  
    android:height="512dp"  
    android:viewportWidth="60"  
    android:viewportHeight="60">  
<path  
    android:pathData="m26,4h-1c-0.5523,0 -1,0.4477 -1,1s0.4477,1 1,1h1c0.5523,0 1,-0.4477 1,-1s-0.4477,-1 -1,-1z"  
    android:fillColor="#000000"  
    android:fillType="nonZero"/>  
<path  
    android:pathData="m30,6h5c0.5523,0 1,-0.4477 1,-1s-0.4477,-1 -1,-1h-5c-0.5523,0 -1,0.4477 -1,1s0.4477,1 1,1z"  
    android:fillColor="#000000"  
    android:fillType="nonZero"/>  
<path  
    android:pathData="m58,48h-3v-2c0,-1.1046 -0.8954,-2 -2,-2h-6v-2h3c1.1046,0 2,-0.8954 2,-2v-8c0,-1.1046 -0.8954,-2 -2,-  
2h-3v-2c0,-1.1046 -0.8954,-2 -2,-2h-8.35c-0.607,-1.7029 -1.9471,-3.043 -3.65,-3.65v-2.582c3.0842,-0.7088 5.5718,-2.9813  
6.556,-5.989 4.306,-1.182 5.24,-7.063 5.438,-9.633 0.0417,-0.5508 -0.1474,-1.0941 -0.522,-1.5 -0.3828,-0.4118 -0.9197,-0.6459  
-1.482,-0.646h-2.99c0,-1.1046 -0.8954,-2 -2,-2h-16c-1.1046,0 -2,0.8954 -2,2h-2.99c-0.5623,0.0001 -1.0992,0.2342 -1.482,0.646  
-0.3747,0.4058 -0.5638,0.9492 -0.522,1.5 0.2,2.571 1.132,8.452 5.438,9.634 0.9845,3.0073 3.472,5.2794 6.556,5.988v2.582c-  
1.7029,0.607 -3.043,1.9471 -3.65,3.65h-8.35c-1.1046,0 -2,0.8954 -2,2v2h-3c-1.1046,0 -2,0.8954 -2,2v8c0,1.1046 0.8954,2 2,2h8c1.1046,0 2,-0.8954 2,-2v-8c0,-  
1.1046 -0.8954,-2 -2,-2h-3v-2h14v2h-3c-1.1046,0 -2,0.8954 -2,2v8c0,1.1046 0.8954,2 2,2h8c1.1046,0 2,-0.8954 2,-2v-8c0,-  
1.1046 -0.8954,-2 -2,-2h-3v-2c0,-1.1046 -0.8954,-2 -2,-2h-6v-2h3c1.1046,0 2,-0.8954 2,-2v-8c0,-1.1046 -0.8954,-2 -2,-2h-3v-  
2h8c0,1.1046 0.8954,2 2,2h10c1.1046,0 2,-0.8954 2,-2h8v2h-3c-1.1046,0 -2,0.8954 -2,2v8c0,1.1046 0.8954,2 2,2h3v2h-6c-  
1.1046,0 -2,0.8954 -2,2v2h-3c-1.1046,0 -2,0.8954 -2,2v8c0,1.1046 0.8954,2 2,2h8c1.1046,0 2,-0.8954 2,-2v-8c0,-1.1046 -  
0.8954,-2 -2,-2h-3v-2h14v2h-3c-1.1046,0 -2,0.8954 -2,2v8c0,1.1046 0.8954,2 2,2h8c1.1046,0 2,-0.8954 2,-2v-8c0,-1.1046 -  
0.8954,-2 -2,-2zM43,3.992c-0.145,1.887 -0.73,5.933 -3.021,7.425 0.006,-0.139 0.021,-0.276 0.021,-0.417v-7zM10,58h-8v-  
8h8zM26,58h-8v-8h8zM18,40h-8v-8h8zM17.01,4h2.99v7c0,0.141 0.015,0.278 0.021,0.418 -2.291,-1.493 -2.874,-5.538 -3.011,-  
7.418zM22,11v-9h16v9c-0.0044,3.8642 -3.1358,6.9956 -7,7h-2c-3.8642,-0.0044 -6.9956,-3.1358 -7,-7zM29,20h2v2h-  
2zM25,28c0,-2.2091 1.7909,-4 4,-4h2c2.2078,0.0033 3.9967,1.7922 4,4zM42,32h8v8h-8zM42,58h-8v-8h8zM58,58h-8v-8h8z"  
    android:fillColor="#000000"  
    android:fillType="nonZero"/>  
</vector>
```

```
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="512dp"
    android:height="512dp"
    android:viewportWidth="512"
    android:viewportHeight="512">
<path
    android:pathData="m256.025,483.334 l101.429,-25.614c57.895,-48.074 94.771,-120.586 94.771,-201.719 0,-125.144 -87.711,-229.801 -205.012,-255.852 -137.316,4.631 -247.213,117.407 -247.213,255.851 0,71.112 29,135.446 75.812,181.836z"
    android:fillColor="#cbe2ff"/>
<path
    android:pathData="m446.914,256c0,83.915 -40.381,158.391 -102.765,205.079l92.031,-23.241c46.815,-46.39 75.82,-110.724 75.82,-181.838 0,-141.385 -114.615,-256 -256 -11.024,0 -21.886,0.698 -32.543,2.05 126.019,15.988 223.457,123.59 223.457,253.95z"
    android:fillColor="#bed8fb"/>
<path
    android:pathData="m319.621,96.952c0,-13.075 -10.599,-23.674 -23.674,-23.674h-81.582c-30.091,0 -54.485,24.394 -54.485,54.485v60.493h192.209v-59.635c0,-13.075 -10.599,-23.674 -23.674,-23.674h-0.798c-4.416,0 -7.996,-3.579 -7.996,-7.995z"
    android:fillColor="#365e7d"/>
<path
    android:pathData="m328.415,104.947h-0.798c-4.416,0 -7.996,-3.58 -7.996,-7.996 0,-13.075 -10.599,-23.674 -23.674,-23.674h-8.945v114.978h65.086v-59.635c0.001,-13.073 -10.599,-23.673 -23.673,-23.673z"
    android:fillColor="#2b4d66"/>
<path
    android:pathData="m425.045,372.355c-6.259,-6.182 -14.001,-10.963 -22.79,-13.745l-69.891,-22.128 -76.348,-2.683 -76.38,2.683 -69.891,22.128c-23.644,7.486 -39.713,29.428 -39.713,54.229v19.094c44.789,47.328 107.451,77.568 177.183,79.92 78.128,-17.353 143.129,-69.576 177.83,-139.498z"
    android:fillColor="#4a80aa"/>
<path
    android:pathData="m441.968,431.932v-19.094c0,-17.536 -8.04,-33.635 -21.105,-44.213 -37.111,75.626 -110.422,130.268 -197.346,141.317 10.492,1.329 21.178,2.038 32.026,2.057 10.423,-0.016 20.708,-0.62 30.824,-1.782 61.031,-7.212 115.485,-35.894 155.601,-78.285z"
    android:fillColor="#407093"/>
<path
    android:pathData="m261.796,508.168c15.489,-30.751 55.822,-118.067 44.321,-172.609l-50.101,-19.499 -50.148,19.5c11.856,56.225 31.37,147.277 45.681,175.29 3.442,-0.826 6.859,-1.721 10.247,-2.682z"
    android:fillColor="#e4f6ff"/>
<path
    android:pathData="m288.197,483.789 l-20.314,-79.917h-23.767l-20.264,79.699 25.058,27.897c6.361,-1.457 12.634,-3.146 18.81,-5.057z"
    android:fillColor="#e28086"/>
<path
    android:pathData="m249.302,511.905c2.075,0.054 4.154,0.091 6.241,0.095 2.415,-0.004 4.822,-0.046 7.222,-0.113l12.907,-14.259c-10.159,3.564 -20.61,6.506 -31.309,8.779z"
    android:fillColor="#dd636e"/>
<path
    android:pathData="m298.774,328.183v-45.066h-85.58v45.066c0,23.632 42.79,49.446 42.79,49.446s42.79,-25.814 42.79,-49.446z"
    android:fillColor="#ffddcce"/>
<path
    android:pathData="m352.089,180.318h-16.359c-9.098,0 -16.473,-7.375 -16.473,-16.473v-9.015c0,-11.851 -11.595,-20.23 -22.847,-16.511 -26.243,8.674 -54.579,8.676 -80.823,0.006l-0.031,-0.01c-11.252,-3.717 -22.845,4.662 -22.845,16.512v9.019c0,9.098 -7.375,16.473 -16.473,16.473h-16.358v26.938c0,6.883 5.58,12.464 12.464,12.464 2.172,0 3.939,1.701 4.076,3.869 2.628,41.668 37.235,74.654 79.565,74.654 42.33,0 76.937,-32.986 79.565,-74.654 0.137,-2.167 1.904,-3.869 4.076,-3.869 6.883,0 12.464,-5.58 12.464,-12.464v-26.939z"
    android:fillColor="#ffddcce"/>
<path
    android:pathData="m335.73,180.318c-9.098,0 -16.473,-7.375 -16.473,-16.473v-9.015c0,-11.851 -11.595,-20.23 -22.847,-16.511 -3.108,1.027 -6.247,1.923 -9.407,2.707v88.972c-0.438,28.948 -16.3,54.142 -39.725,67.758 2.861,0.311 5.763,0.486 8.706,0.486 42.33,0 76.937,-32.986 79.565,-74.654 0.137,-2.167 1.904,-3.869 4.076,-3.869 6.883,0 12.464,-5.58 12.464,-12.464v-26.938h-16.359z"
    android:fillColor="#ffccbbe"/>
<path
    android:pathData="m213.194,316.06 l-33.558,27.267 35.192,43.513c4.281,4.168 11.019,4.424 15.605,0.594l26.465,-22.107z"
    android:fillColor="#ffccbbe"/>
```

```
    android:fillColor="#f4fbff"/>
<path
    android:pathData="m298.79,316.06 l-41.892,49.267 24.874,21.268c4.557,3.896 11.327,3.7 15.651,-0.453l34.94,-42.815z"
    android:fillColor="#f4fbff"/>
<path
    android:pathData="m213.194,316.06 l-49.256,24.199c-3.75,1.842 -5.256,6.404 -3.341,10.117l9.65,18.71c2.501,4.848
1.578,10.756 -2.282,14.61 -1.987,1.983 -4.139,4.131 -6.004,5.993 -3.338,3.332 -4.537,8.255 -3.067,12.737 11.651,35.517
67.725,89.828 88.946,109.478 1.427,0.038 2.857,0.064 4.29,0.08 -15.389,-29.933 -69.922,-143.655 -38.936,-195.924z"
    android:fillColor="#365e7d"/>
<path
    android:pathData="m344.019,383.695c-3.861,-3.854 -4.783,-9.762 -2.282,-14.61l9.65,-18.71c1.915,-3.713 0.409,-8.275 -
3.341,-10.117l-49.256,-24.198c30.978,52.255 -23.517,165.929 -38.923,195.9 1.448,-0.025 2.893,-0.061 4.335,-0.109 21.265,-
19.695 77.248,-73.94 88.888,-109.424 1.47,-4.482 0.271,-9.405 -3.067,-12.737 -1.865,-1.863 -4.017,-4.012 -6.004,-5.995z"
    android:fillColor="#365e7d"/>
<path
    android:pathData="m256.898,365.327 l-26.06,21.764 13.278,16.781h23.767l13.279,-17.771z"
    android:fillColor="#dd636e"/>
</vector>
```

```
<vector android:height="128dp" android:viewportHeight="51.76"
    android:viewportWidth="51.76" android:width="128dp" xmlns:android="http://schemas.android.com/apk/res/android">
    <path android:fillColor="#010002" android:pathData="M36.3695,4.82m4.82,-0a4.82,4.82 0,1 0,-9.64 0a4.82,4.82 45,1 0,9.64
-0"/>
    <path android:fillColor="#010002" android:pathData="M6.0705,46.201c5.744,-8.573 12.312,-19.74 13.416,-21.625c0.1,-0.342
0.232,-0.689 0.414,-1.038c1.312,-2.557 2.684,-5.085 4.084,-7.596c-1.484,0.116 -2.972,0.231 -4.455,0.346c-1.02,2.01 -
2.035,4.021 -3.053,6.03c-1.639,3.237 -6.49,0.389 -4.855,-2.839c1.312,-2.593 2.867,-8.425 6.254,-8.688c3.604,-0.28 7.207,-0.56
10.81,-0.839c0.088,-0.007 0.163,0.005 0.247,0.005c2.724,-0.729 5.917,1.193 6.495,3.97c1.842,3.989 4.487,7.104
8.925,8.404c3.474,1.016 1.994,6.443 -1.495,5.423c-4.387,-1.284 -7.538,-3.798 -9.927,-7.107c-1.447,2.58 -2.86,5.176 -
4.214,7.808c2.977,2.053 5.578,4.536 7.857,7.401c0.636,0.799 0.745,1.621 0.531,2.344c0.008,0.27 -0.009,0.55 -0.086,0.854c-
0.877,3.472 -1.753,6.938 -2.631,10.406c-0.931,3.696 -6.782,2.646 -5.843,-1.074c0.835,-3.305 1.671,-6.61 2.506,-9.917c-2.094,
2.469 -4.563,-4.617 -7.384,-6.221c-0.664,-0.378 -1.082,-0.899 -1.314,-1.473c-3.859,5.914 -7.719,14.096 -
11.576,19.055C8.4225,52.856 3.7135,49.233 6.0705,46.201z"/>
</vector>
```

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <gradient
        android:startColor="#20000000"
        android:endColor="@android:color/transparent"
        android:angle="90" >
    </gradient>
</shape>
```

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item>
        <layer-list>
            <item>
                <shape>
                    <solid android:color="@color/colorWhite"></solid>

                    <corners android:radius="3dp" />
                    <padding android:right="@dimen/margin_12dp" />

                    <stroke android:width="2dp" android:color="@color/colorBlueLight" />
                </shape>
            </item>
            <item>
                <bitmap android:gravity="center_vertical|right"
                        android:src="@drawable/download" />
            </item>
        </layer-list>
    </item>
</selector>
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".activity.changepasswordactivity"
8     tools:ignore="ContentDescription,Autofill,LabelFor">
9
10 <ImageView
11     android:layout_width="match_parent"
12     android:layout_height="500dp"
13     android:background="@drawable/bg_bottom_circle"
14     app:layout_constraintEnd_toEndOf="parent"
15     app:layout_constraintHorizontal_bias="0.0"
16     app:layout_constraintStart_toStartOf="parent"
17     app:layout_constraintTop_toTopOf="parent" />
18
19 <ImageView
20     android:id="@+id/app_logo"
21     android:layout_width="128dp"
22     android:layout_height="128dp"
23     android:adjustViewBounds="true"
24     android:src="@drawable/app_icon"
25     app:layout_constraintBottom_toTopOf="@+id/crd_form"
26     app:layout_constraintEnd_toEndOf="parent"
27     app:layout_constraintStart_toStartOf="parent"
28     app:layout_constraintTop_toTopOf="parent" />
29
30 <androidx.cardview.widget.CardView
31     android:id="@+id/crd_form"
32     android:layout_width="match_parent"
33     android:layout_height="wrap_content"
34     android:layout_marginHorizontal="24dp"
35     app:cardCornerRadius="24dp"
36     app:layout_constraintBottom_toBottomOf="parent"
37     app:layout_constraintEnd_toEndOf="parent"
38     app:layout_constraintStart_toStartOf="parent"
39     app:layout_constraintTop_toBottomOf="@+id/app_logo">
40
41 <ScrollView
42     android:layout_width="match_parent"
43     android:layout_height="match_parent">
44
45     <LinearLayout
46         android:layout_width="match_parent"
47         android:layout_height="wrap_content"
48         android:background="@color/white"
49         android:orientation="vertical"
50         android:paddingHorizontal="18dp"
51         android:paddingVertical="24dp">
52
53         <TextView
54             android:layout_width="wrap_content"
55             android:layout_height="wrap_content"
56             android:layout_gravity="center_horizontal"
57             android:text="@string/change_password"
58             android:textColor="@color/colorPrimary"
59             android:textSize="24sp"
60             android:textStyle="bold" />
61
62         <com.google.android.material.textfield.TextInputLayout
63             android:layout_width="match_parent"
64             android:layout_height="wrap_content"
65             android:layout_marginTop="32dp"
66             android:background="@color/white"
```

```
67 android:hint="@string/current_password"
68 app:passwordToggleEnabled="true">
69
70 <EditText
71     android:id="@+id edt_old_password"
72     android:layout_width="match_parent"
73     android:layout_height="wrap_content"
74     android:background="@color/white"
75     android:inputType="textPassword"
76     android:singleLine="true" />
77 </com.google.android.material.textfield.TextInputLayout>
78
79
80 <com.google.android.material.textfield.TextInputLayout
81     android:layout_width="match_parent"
82     android:layout_height="wrap_content"
83     android:background="@color/white"
84     android:hint="@string/new_password"
85     app:passwordToggleEnabled="true">
86
87 <EditText
88     android:id="@+id edt_new_password"
89     android:layout_width="match_parent"
90     android:layout_height="wrap_content"
91     android:background="@color/white"
92     android:inputType="textPassword"
93     android:singleLine="true" />
94 </com.google.android.material.textfield.TextInputLayout>
95
96
97 <com.google.android.material.textfield.TextInputLayout
98     android:layout_width="match_parent"
99     android:layout_height="wrap_content"
100    android:background="@color/white"
101    android:hint="@string/confirm_password"
102    app:passwordToggleEnabled="true">
103
104 <EditText
105     android:id="@+id edt_confirm_password"
106     android:layout_width="match_parent"
107     android:layout_height="wrap_content"
108     android:background="@color/white"
109     android:inputType="textPassword"
110     android:singleLine="true" />
111 </com.google.android.material.textfield.TextInputLayout>
112
113
114 <TextView
115     android:id="@+id tv_btn_login"
116     android:layout_width="match_parent"
117     android:layout_height="48dp"
118     android:layout_gravity="center_horizontal"
119     android:layout_marginTop="24dp"
120     android:background="@drawable/bg_btn"
121     android:gravity="center"
122     android:text="@string/submit"
123     android:textColor="@color/white"
124     android:textSize="18sp" />
125
126 </LinearLayout>
127
128 </ScrollView>
129
130
131
132 </androidx.cardview.widget.CardView>
133
</androidx.constraintlayout.widget.ConstraintLayout>
```



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.drawerlayout.widget.DrawerLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/drawer_layout1"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:fitsSystemWindows="true"
9     tools:openDrawer="start">
10
11     <ImageButton
12         android:id="@+id/back"
13         android:layout_width="wrap_content"
14         android:layout_height="wrap_content"
15         android:layout_marginStart="15dp"
16         android:layout_marginTop="35dp"
17         android:background="@android:color/transparent"
18         android:elevation="10dp"
19         android:src="@drawable/ic_back"
20         android:visibility="gone"
21         android:layout_marginLeft="15dp" />
22
23     <include
24         layout="@layout/app_bar_map"
25         android:layout_width="match_parent"
26         android:layout_height="match_parent" />
27
28     <com.google.android.material.navigation.NavigationView
29         android:id="@+id/nav_view1"
30         android:layout_width="match_parent"
31         android:layout_height="match_parent"
32         android:layout_gravity="start"
33         android:fitsSystemWindows="true"
34         app:headerLayout="@layout/nav_header_main"
35         app:itemIconTint="@color/black"
36         app:menu="@menu/activity_main_drawer" />
37
38 </androidx.drawerlayout.widget.DrawerLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:ignore="ContentDescription"
    tools:context=".activity.LoginActivity">
```

<ImageView

```
    android:layout_width="match_parent"
    android:layout_height="500dp"
    android:background="@drawable/bg_bottom_circle"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
/>
```

<ImageView

```
    android:id="@+id/app_logo"
    android:layout_width="128dp"
    android:layout_height="128dp"
    android:layout_marginTop="56dp"
    android:src="@drawable/app_icon"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

<TextView

```
    android:id="@+id/tv_dont_have"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="@string/don_t_have_an_account"
    android:textColor="@color/black"
    android:textStyle="bold"
    app:layout_constraintBottom_toTopOf="@+id/tv_sign_up" />
```

<TextView

```
    android:id="@+id/tv_sign_up"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="56dp"
    android:gravity="center_horizontal"
    android:paddingHorizontal="4dp"
    android:paddingVertical="4dp"
    android:text="@string/register"
    android:textColor="@color/colorPrimaryDark"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
```

<androidx.cardview.widget.CardView

```
    android:id="@+id/crd_form"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="24dp"
    app:cardCornerRadius="24dp"
    app:layout_constraintBottom_toTopOf="@+id/tv_dont_have"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/app_logo">
```

<ScrollView

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
```

<LinearLayout

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/white"
    android:orientation="vertical"
    android:paddingHorizontal="18dp"
    android:paddingVertical="24dp">
```

<TextView

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:text="@string/login"
    android:textColor="@color/colorPrimary"
    android:textSize="24sp"
    android:textStyle="bold" />
```

<com.google.android.material.textfield.TextInputLayout

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:background="@color/white"
    android:hint="@string/e_email">
```

<com.google.android.material.textfield.TextInputEditText

```
    android:id="@+id/ed_login_email"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="24dp"
    android:background="@color/white"
    android:inputType="textEmailAddress" />
```

```
</com.google.android.material.textfield.TextInputLayout>
```

<com.google.android.material.textfield.TextInputLayout

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/white"
    android:hint="@string/password">
```

<com.google.android.material.textfield.TextInputEditText

```
    android:id="@+id/ed_login_password"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/white"
    android:inputType="textPassword" />
```

```
</com.google.android.material.textfield.TextInputLayout>
```

<TextView

```
    android:id="@+id/btn_forgot_password"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="right"
    android:layout_marginTop="4dp"
    android:paddingVertical="4dp"
    android:text="@string/forgot_password"
    android:textColor="@color/colorPrimaryDark"
    android:textStyle="bold" />
```

<TextView

```
        android:id="@+id/btn_login"
        android:layout_width="match_parent"
        android:layout_height="48dp"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="24dp"
        android:background="@drawable/bg_btn"
        android:gravity="center"
        android:text="@string/login"
        android:textColor="@color/white"
        android:textSize="18sp" />

    </LinearLayout>

</ScrollView>

</androidx.cardview.widget.CardView>

</androidx.constraintlayout.widget.ConstraintLayout>
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:background="@android:color/transparent"
8     android:orientation="vertical"
9     tools:ignore="ContentDescription,Autofill,LabelFor">
10
11 <androidx.cardview.widget.CardView
12     android:id="@+id/destCard"
13     android:layout_width="match_parent"
14     android:layout_height="140dp"
15     android:paddingStart="15dp"
16     android:paddingEnd="15dp"
17     android:transitionName="current"
18     app:cardBackgroundColor="@android:color/white"
19     app:cardElevation="2dp"
20     app:cardMaxElevation="2dp"
21     app:cardPreventCornerOverlap="true">
22
23 <LinearLayout
24     android:layout_width="match_parent"
25     android:layout_height="wrap_content"
26     android:layout_marginTop="28dp"
27     android:orientation="vertical">
28
29 <ImageButton
30     android:id="@+id/close"
31     android:layout_width="wrap_content"
32     android:layout_height="wrap_content"
33     android:layout_marginStart="15dp"
34     android:layout_marginEnd="15dp"
35     android:background="@android:color/transparent"
36     app:srcCompat="@drawable/ic_back"
37     tools:ignore="VectorDrawableCompat" />
38
39 <LinearLayout
40     android:layout_width="match_parent"
41     android:layout_height="match_parent"
42     android:layout_marginStart="15dp"
43     android:layout_marginTop="15dp"
44     android:layout_marginEnd="15dp">
45
46 <LinearLayout
47     android:layout_width="0dp"
48     android:layout_height="match_parent"
49     android:layout_weight="2"
50     android:orientation="vertical">
51
52 <EditText
53     android:id="@+id/mSearchText"
54     android:layout_width="match_parent"
55     android:layout_height="40dp"
56     android:layout_marginTop="15dp"
57     android:background="#e2e2e2"
58     android:ellipsize="end"
59     android:gravity="start|center"
60     android:hint="@string/set_pick_up"
61     android:imeOptions="actionDone"
62     android:paddingStart="5dp"
63     android:paddingEnd="5dp"
64     android:singleLine="true"
65     android:textAllCaps="false"
66     android:textColor="@color/colorPrimaryDark"
```

```
67     android:textColorHint="#b3b1b1"
68     android:textSize="14sp" />
69   </LinearLayout>
70   </LinearLayout>
71   </LinearLayout>
72 </androidx.cardview.widget.CardView>
73
74 <FrameLayout
75   android:layout_width="match_parent"
76   android:layout_height="wrap_content">
77
78   <TextView
79     android:layout_width="match_parent"
80     android:layout_height="8dp"
81     android:background="#dcdbdb" />
82
83   <ImageView
84     android:id="@+id/iv_line"
85     android:layout_width="match_parent"
86     android:layout_height="wrap_content"
87     android:layout_marginLeft="0dp"
88     android:layout_marginRight="0dp"
89     android:background="@drawable/avd_line" />
90
91 </FrameLayout>
92
93 <include layout="@layout/setloc" />
94
95 <androidx.recyclerview.widget.RecyclerView
96   android:id="@+id/list_search"
97   android:layout_width="match_parent"
98   android:layout_height="match_parent"
99   android:background="@android:color/white"
100  android:paddingStart="12dp"
101  android:paddingEnd="12dp"
102  android:scrollbars="none"
103  android:visibility="gone"/>
104
105 <FrameLayout
106   android:id="@+id/frame"
107   android:layout_width="match_parent"
108   android:layout_height="match_parent"
109   android:visibility="gone">
110
111   <fragment
112     android:id="@+id/map"
113     android:name="com.google.android.gms.maps.SupportMapFragment"
114     android:layout_width="match_parent"
115     android:layout_height="match_parent"
116     tools:context="logicbeansz.com.CheapRidedDrivernz.SearchActivity" />
117
118   <ImageView
119     android:id="@+id/pin"
120     android:layout_width="25dp"
121     android:layout_height="28dp"
122     android:layout_gravity="center"
123     android:layout_marginBottom="30dp"
124     android:elevation="15dp"
125     android:scaleType="fitCenter"
126     android:src="@drawable/drag_mark"
127     android:visibility="gone" />
128
129   <Button
130     android:id="@+id/done"
131     android:layout_width="wrap_content"
132     android:layout_height="wrap_content"
133     android:layout_gravity="bottom|center"
134     android:layout_marginBottom="25dp"
```

```
135 android:paddingStart="80dp"
136 android:paddingEnd="80dp"
137 android:text="@string/done"
138 android:textAllCaps="false"
139 android:textColor="@android:color/white"
140 android:textSize="15sp"
141 android:visibility="gone" />
142 </FrameLayout>
</LinearLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    1     xmlns:app="http://schemas.android.com/apk/res-auto"
    2     xmlns:tools="http://schemas.android.com/tools"
    3     android:layout_width="match_parent"
    4     android:layout_height="match_parent"
    5     tools:context=".activity.RegisterActivity"
    6     tools:ignore="ContentDescription,Autofill,LabelFor" >
    7
    8     <ImageView
    9         android:layout_width="match_parent"
   10        android:layout_height="500dp"
   11        android:background="@drawable/bg_bottom_circle"
   12        app:layout_constraintEnd_toEndOf="parent"
   13        app:layout_constraintHorizontal_bias="0.0"
   14        app:layout_constraintStart_toStartOf="parent"
   15        app:layout_constraintTop_toTopOf="parent" />
   16
   17     <ImageView
   18         android:id="@+id/app_logo"
   19         android:layout_width="128dp"
   20         android:layout_height="128dp"
   21         android:adjustViewBounds="true"
   22         android:src="@drawable/app_icon"
   23         app:layout_constraintBottom_toTopOf="@+id/crd_form"
   24         app:layout_constraintEnd_toEndOf="parent"
   25         app:layout_constraintStart_toStartOf="parent"
   26         app:layout_constraintTop_toTopOf="parent" />
   27
   28     <TextView
   29         android:id="@+id/tv_dont_have"
   30         android:layout_width="match_parent"
   31         android:layout_height="wrap_content"
   32         android:gravity="center"
   33         android:text="@string/already_have_an_account"
   34         android:textColor="@color/black"
   35         android:textStyle="bold"
   36         app:layout_constraintBottom_toTopOf="@+id/tv_goto_login" />
   37
   38     <TextView
   39         android:id="@+id/tv_goto_login"
   40         android:layout_width="wrap_content"
   41         android:layout_height="wrap_content"
   42         android:gravity="center_horizontal"
   43         android:paddingHorizontal="4dp"
   44         android:paddingVertical="4dp"
   45         android:text="@string/login"
   46         android:textColor="@color/colorPrimaryDark"
   47         android:textStyle="bold"
   48         app:layout_constraintBottom_toBottomOf="parent"
   49         app:layout_constraintEnd_toEndOf="parent"
   50         app:layout_constraintStart_toStartOf="parent"
   51         app:layout_constraintTop_toBottomOf="@+id/crd_form" />
   52
   53     <androidx.cardview.widget.CardView
   54         android:id="@+id/crd_form"
   55         android:layout_width="match_parent"
   56         android:layout_height="wrap_content"
   57         android:layout_marginHorizontal="24dp"
   58         app:cardCornerRadius="24dp"
   59         app:layout_constraintBottom_toTopOf="@+id/tv_goto_login"
   60         app:layout_constraintEnd_toEndOf="parent"
   61         app:layout_constraintStart_toStartOf="parent"
   62         app:layout_constraintTop_toBottomOf="@+id/app_logo">
   63
   64     <ScrollView
   65
```

```
67 android:layout_width="match_parent"
68 android:layout_height="wrap_content">
69
70 <LinearLayout
71     android:layout_width="match_parent"
72     android:layout_height="wrap_content"
73     android:background="@color/white"
74     android:orientation="vertical"
75     android:paddingHorizontal="18dp"
76     android:paddingVertical="24dp">
77
78     <TextView
79         android:layout_width="wrap_content"
80         android:layout_height="wrap_content"
81         android:layout_gravity="center_horizontal"
82         android:text="@string/register"
83         android:textColor="@color/colorPrimaryDark"
84         android:textSize="24sp"
85         android:textStyle="bold" />
86
87     <com.google.android.material.textfield.TextInputLayout
88         android:layout_width="match_parent"
89         android:layout_height="wrap_content"
90         android:layout_marginTop="32dp"
91         android:background="@color/white"
92         android:hint="@string/username">
93
94         <EditText
95             android:id="@+id/ed_username"
96             android:layout_width="match_parent"
97             android:layout_height="wrap_content"
98             android:background="@color/white"
99             android:inputType="text"
100            android:singleLine="true"
101            tools:ignore="Autofill,LabelFor" />
102     </com.google.android.material.textfield.TextInputLayout>
103
104
105     <com.google.android.material.textfield.TextInputLayout
106         android:layout_width="match_parent"
107         android:layout_height="wrap_content"
108         android:background="@color/white"
109         android:hint="@string/contact_number">
110
111         <EditText
112             android:id="@+id/ed_contact"
113             android:layout_width="match_parent"
114             android:layout_height="wrap_content"
115             android:background="@color/white"
116             android:inputType="number"
117             android:maxLength="10"
118             android:singleLine="true" />
119     </com.google.android.material.textfield.TextInputLayout>
120
121
122     <com.google.android.material.textfield.TextInputLayout
123         android:layout_width="match_parent"
124         android:layout_height="wrap_content"
125         android:background="@color/white"
126         android:hint="@string/date_of_birth">
127
128         <EditText
129             android:id="@+id/ed_dob"
130             android:layout_width="match_parent"
131             android:layout_height="wrap_content"
132             android:background="@color/white"
133             android:focusable="false"
```

```
135     android:inputType="text"
136     android:singleLine="true" />
137 </com.google.android.material.textfield.TextInputLayout>
138
139 <com.google.android.material.textfield.TextInputLayout
140     android:layout_width="match_parent"
141     android:layout_height="wrap_content"
142     android:background="@color/white"
143     android:hint="@string/e_mail">
144
145
146 <EditText
147     android:id="@+id/ed_email"
148     android:layout_width="match_parent"
149     android:layout_height="wrap_content"
150     android:background="@color/white"
151     android:inputType="textEmailAddress"
152     android:singleLine="true" />
153 </com.google.android.material.textfield.TextInputLayout>
154
155 <com.google.android.material.textfield.TextInputLayout
156     android:layout_width="match_parent"
157     android:layout_height="wrap_content"
158     android:background="@color/white"
159     android:hint="@string/password">
160
161 <EditText
162     android:id="@+id/ed_password"
163     android:layout_width="match_parent"
164     android:layout_height="wrap_content"
165     android:background="@color/white"
166     android:inputType="textPassword"
167     android:singleLine="true" />
168 </com.google.android.material.textfield.TextInputLayout>
169
170 <com.google.android.material.textfield.TextInputLayout
171     android:layout_width="match_parent"
172     android:layout_height="wrap_content"
173     android:background="@color/white"
174     android:hint="@string/confirm_password">
175
176 <EditText
177     android:id="@+id/ed_confirm_password"
178     android:layout_width="match_parent"
179     android:layout_height="wrap_content"
180     android:background="@color/white"
181     android:inputType="textPassword"
182     android:singleLine="true" />
183 </com.google.android.material.textfield.TextInputLayout>
184
185
186 <TextView
187     android:id="@+id/tv_btn_login"
188     android:layout_width="match_parent"
189     android:layout_height="48dp"
190     android:layout_gravity="center_horizontal"
191     android:layout_marginTop="24dp"
192     android:background="@drawable/bg_btn"
193     android:gravity="center"
194     android:text="@string/register"
195     android:textColor="@color/white"
196     android:textSize="18sp" />
197
198 </LinearLayout>
199
200 <ScrollView>
201
202
```

203

</androidx.cardview.widget.CardView>

</androidx.constraintlayout.widget.ConstraintLayout>

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    1   xmlns:app="http://schemas.android.com/apk/res-auto"
    2   xmlns:tools="http://schemas.android.com/tools"
    3   android:layout_width="match_parent"
    4   android:layout_height="match_parent"
    5   tools:context=".activity.SearchActivity"
    6   android:background="@android:color/transparent"
    7   android:orientation="vertical"
    8   tools:ignore="ContentDescription,Autofill,LabelFor">
    9

10  <androidx.cardview.widget.CardView
11      android:id="@+id/destCard"
12      android:layout_width="match_parent"
13      android:layout_height="180dp"
14      android:paddingStart="15dp"
15      android:paddingEnd="15dp"
16      android:transitionName="dest"
17      app:cardBackgroundColor="@android:color/white"
18      app:cardElevation="2dp"
19      app:cardMaxElevation="2dp"
20      app:cardPreventCornerOverlap="true">
21

22  <LinearLayout
23      android:layout_width="match_parent"
24      android:layout_height="wrap_content"
25      android:layout_marginTop="28dp"
26      android:orientation="vertical">
27

28  <ImageButton
29      android:id="@+id/close"
30      android:layout_width="wrap_content"
31      android:layout_height="wrap_content"
32      android:layout_marginStart="15dp"
33      android:layout_marginEnd="15dp"
34      android:background="@android:color/transparent"
35      android:src="@drawable/ic_back" />
36

37  <LinearLayout
38      android:layout_width="match_parent"
39      android:layout_height="match_parent"
40      android:layout_marginTop="15dp"
41      android:weightSum="3">
42

43  <LinearLayout
44      android:layout_width="0dp"
45      android:layout_height="match_parent"
46      android:layout_weight="0.5"
47      android:elevation="10dp"
48      android:gravity="center"
49      android:orientation="vertical">
50

51  <ImageView
52      android:layout_width="10dp"
53      android:layout_height="10dp"
54      android:layout_gravity="center"
55      android:scaleType="fitCenter"
56      android:src="@drawable/greenball" />
57

58  <ImageView
59      android:layout_width="5dp"
60      android:layout_height="5dp"
61      android:layout_gravity="center"
62      android:layout_marginTop="4dp"
63      android:scaleType="fitCenter"
64      android:src="@drawable/assball" />
65
```

```
66
67 <ImageView
68     android:layout_width="5dp"
69     android:layout_height="5dp"
70     android:layout_gravity="center"
71     android:layout_marginTop="4dp"
72     android:scaleType="fitCenter"
73     android:src="@drawable/assball" />
74
75 <ImageView
76     android:layout_width="5dp"
77     android:layout_height="5dp"
78     android:layout_gravity="center"
79     android:layout_marginTop="4dp"
80     android:scaleType="fitCenter"
81     android:src="@drawable/assball" />
82
83 <ImageView
84     android:layout_width="5dp"
85     android:layout_height="5dp"
86     android:layout_gravity="center"
87     android:layout_marginTop="4dp"
88     android:scaleType="fitCenter"
89     android:src="@drawable/assball" />
90
91 <ImageView
92     android:layout_width="10dp"
93     android:layout_height="10dp"
94     android:layout_gravity="center"
95     android:layout_marginTop="4dp"
96     android:scaleType="fitCenter"
97     android:src="@drawable/destball" />
98
99 </LinearLayout>
100
101 <LinearLayout
102     android:layout_width="0dp"
103     android:layout_height="match_parent"
104     android:layout_weight="2"
105     android:orientation="vertical">
106
107 <Button
108     android:id="@+id/cuurent"
109     android:layout_width="match_parent"
110     android:layout_height="40dp"
111     android:background="#e6e4e4"
112     android:ellipsize="end"
113     android:gravity="start|center"
114     android:paddingStart="5dp"
115     android:paddingEnd="5dp"
116     android:singleLine="true"
117     android:text="@string/current_location"
118     android:textAllCaps="false"
119     android:textColor="@color/white"
120     android:textSize="14sp" />
121
122 <EditText
123     android:id="@+id/mSearchText"
124     android:layout_width="match_parent"
125     android:layout_height="40dp"
126     android:layout_marginTop="15dp"
127     android:background="@color/colorPrimary"
128     android:ellipsize="end"
129     android:gravity="start|center"
130     android:hint="@string/where_to"
131     android:imeOptions="actionDone"
132     android:paddingStart="5dp"
133
```

```
134     android:paddingEnd="5dp"
135     android:singleLine="true"
136     android:textAllCaps="false"
137     android:textColor="@color/white"
138     android:textColorHint="@color/white"
139     android:textSize="14sp" />
140 </LinearLayout>
141
142     </LinearLayout>
143 </LinearLayout>
144 </androidx.cardview.widget.CardView>
145
146 <FrameLayout
147     android:layout_width="match_parent"
148     android:layout_height="wrap_content">
149
150     <TextView
151         android:layout_width="match_parent"
152         android:layout_height="8dp"
153         android:background="#dcdbdb" />
154
155     <ImageView
156         android:id="@+id/iv_line"
157         android:layout_width="match_parent"
158         android:layout_height="wrap_content"
159         android:layout_marginLeft="0dp"
160         android:layout_marginRight="0dp"
161         android:background="@drawable/avd_line" />
162
163 </FrameLayout>
164
165 <include layout="@layout/setloc" />
166
167
168 <androidx.recyclerview.widget.RecyclerView
169     android:id="@+id/list_search"
170     android:layout_width="match_parent"
171     android:layout_height="match_parent"
172     android:background="@android:color/white"
173     android:paddingStart="12dp"
174     android:paddingEnd="12dp"
175     android:scrollbars="none"
176     android:visibility="gone" />
177
178 <FrameLayout
179     android:id="@+id/frame"
180     android:layout_width="match_parent"
181     android:layout_height="match_parent"
182     android:visibility="gone"
183     tools:visibility="visible">
184
185     <fragment
186         android:id="@+id/map"
187         android:name="com.google.android.gms.maps.SupportMapFragment"
188         android:layout_width="match_parent"
189         android:layout_height="match_parent"
190         tools:context="logicbeanzs.com.CheapRidedDrivernz.SearchActivity" />
191
192     <ImageView
193         android:id="@+id/pin"
194         android:layout_width="25dp"
195         android:layout_height="28dp"
196         android:layout_gravity="center"
197         android:layout_marginBottom="30dp"
198         android:elevation="15dp"
199         android:scaleType="fitCenter"
200         android:src="@drawable/drag_mark"
```

```
202     android:visibility="gone" />
203
204 <Button
205     android:id="@+id/done"
206     android:layout_width="wrap_content"
207     android:layout_height="wrap_content"
208     android:layout_gravity="bottom|center"
209     android:layout_marginBottom="25dp"
210     android:paddingStart="80dp"
211     android:paddingEnd="80dp"
212     android:text="@string/done"
213     android:textAllCaps="false"
214     android:textColor="@android:color/white"
215     android:textSize="15sp"
216     android:visibility="gone" />
217
218 <View
219     android:layout_width="match_parent"
220     android:layout_height="8dp"
221     android:background="#dcdbdb" />
222 </FrameLayout>
</LinearLayout>
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".activity.UpdateProfileActivity">
8
8 <com.google.android.material.appbar.AppBarLayout
9     android:id="@+id/appbar"
10    android:layout_width="match_parent"
11    android:layout_height="wrap_content"
12    android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"
13    app:elevation="0dp"
14    app:layout_constraintEnd_toEndOf="parent"
15    app:layout_constraintStart_toStartOf="parent"
16    app:layout_constraintTop_toTopOf="parent">
17
18 <androidx.appcompat.widget.Toolbar
19     android:id="@+id/toolbar"
20     android:layout_width="match_parent"
21     android:layout_height="?attr/actionBarSize"
22     android:background="@color/colorPrimary"
23     app:layout_scrollFlags="scroll|enterAlways"
24     app:popupTheme="@style/ThemeOverlay.AppCompat.Light">
25
26 <LinearLayout
27     android:layout_width="match_parent"
28     android:layout_height="wrap_content"
29     android:orientation="horizontal">
30
31 <ImageView
32     android:id="@+id/iv_back"
33     android:layout_width="wrap_content"
34     android:layout_height="wrap_content"
35     app:srcCompat="@drawable/ic_back_white" />
36
37 <TextView
38     android:layout_width="0dp"
39     android:layout_height="wrap_content"
40     android:layout_weight="1"
41     android:gravity="center"
42     android:text="@string/statistic"
43     android:textColor="@color/white"
44     android:textSize="24sp" />
45
46 <ImageView
47     android:id="@+id/iv_filter"
48     android:layout_width="wrap_content"
49     android:layout_height="wrap_content"
50     android:layout_marginEnd="@dimen/dimen_12dp"
51     app:srcCompat="@drawable/ic_filter" />
52
53 </LinearLayout>
54
55
56 </androidx.appcompat.widget.Toolbar>
57 </com.google.android.material.appbar.AppBarLayout>
58
59
60 <androidx.constraintlayout.widget.ConstraintLayout
61     android:id="@+id/crd_count"
62     android:layout_width="match_parent"
63     android:layout_height="wrap_content"
64     android:background="@color/colorPrimaryDark"
65     android:paddingHorizontal="@dimen/margin_8dp"
66     android:paddingVertical="@dimen/dimen_4dp"
```

```
67 app:layout_constraintEnd_toEndOf="parent"
68 app:layout_constraintStart_toStartOf="parent"
69 app:layout_constraintTop_toBottomOf="@+id/appbar">
70
71 <TextView
72     android:id="@+id/txtCO2Title"
73     android:layout_width="wrap_content"
74     android:layout_height="wrap_content"
75     android:text="@string/total_co2"
76     android:textColor="@color/white"
77     android:textSize="@dimen/margin_14sp"
78     android:textStyle="bold"
79     app:layout_constraintStart_toStartOf="parent"
80     app:layout_constraintTop_toTopOf="parent" />
81
82 <TextView
83     android:id="@+id/txtCO2"
84     android:layout_width="wrap_content"
85     android:layout_height="wrap_content"
86     android:text="@string/_0"
87     android:textColor="@color/white"
88     android:textSize="@dimen/margin_14sp"
89     android:textStyle="bold"
90     app:layout_constraintStart_toStartOf="parent"
91     app:layout_constraintTop_toBottomOf="@+id/txtCO2Title" />
92
93 <TextView
94     android:id="@+id/txtCaloriesTitle"
95     android:layout_width="wrap_content"
96     android:layout_height="wrap_content"
97     android:text="@string/total_calories"
98     android:textColor="@color/white"
99     android:textSize="@dimen/margin_14sp"
100    android:textStyle="bold"
101    app:layout_constraintEnd_toEndOf="parent"
102    app:layout_constraintTop_toTopOf="parent" />
103
104 <TextView
105     android:id="@+id/txtCalories"
106     android:layout_width="wrap_content"
107     android:layout_height="wrap_content"
108     android:text="@string/_0"
109     android:textColor="@color/white"
110     android:textSize="@dimen/margin_14sp"
111     android:textStyle="bold"
112     app:layout_constraintEnd_toEndOf="parent"
113     app:layout_constraintTop_toBottomOf="@+id/txtCaloriesTitle" />
114
115
116 </androidx.constraintlayout.widget.ConstraintLayout>
117
118 <androidx.recyclerview.widget.RecyclerView
119     android:id="@+id/recStatistic"
120     android:layout_width="match_parent"
121     android:layout_height="0dp"
122     app:layout_constraintBottom_toBottomOf="parent"
123     app:layout_constraintEnd_toEndOf="parent"
124     app:layout_constraintStart_toStartOf="parent"
125     app:layout_constraintTop_toBottomOf="@+id/crd_count" />
126
127 <TextView
128     android:id="@+id/txtNoData"
129     android:layout_width="wrap_content"
130     android:layout_height="wrap_content"
131     android:text="@string/no_data_available"
132     app:layout_constraintBottom_toBottomOf="parent"
133     app:layout_constraintEnd_toEndOf="parent"
```

135
136 app:layout_constraintStart_toStartOf="parent"
137 app:layout_constraintTop_toBottomOf="@+id/appbar" />

</androidx.constraintlayout.widget.ConstraintLayout>

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context=".activity.StatisticsActivity"
8     tools:ignore="ContentDescription,Autofill,LabelFor">
9
10 <ImageView
11     android:layout_width="match_parent"
12     android:layout_height="500dp"
13     android:background="@drawable/bg_bottom_circle"
14     app:layout_constraintEnd_toEndOf="parent"
15     app:layout_constraintHorizontal_bias="0.0"
16     app:layout_constraintStart_toStartOf="parent"
17     app:layout_constraintTop_toTopOf="parent" />
18
19 <ImageView
20     android:id="@+id/app_logo"
21     android:layout_width="128dp"
22     android:layout_height="128dp"
23     android:adjustViewBounds="true"
24     android:src="@drawable/app_icon"
25     app:layout_constraintBottom_toTopOf="@+id/crd_form"
26     app:layout_constraintEnd_toEndOf="parent"
27     app:layout_constraintStart_toStartOf="parent"
28     app:layout_constraintTop_toTopOf="parent" />
29
30 <androidx.cardview.widget.CardView
31     android:id="@+id/crd_form"
32     android:layout_width="match_parent"
33     android:layout_height="wrap_content"
34     android:layout_marginHorizontal="24dp"
35     app:cardCornerRadius="24dp"
36     app:layout_constraintBottom_toBottomOf="parent"
37     app:layout_constraintEnd_toEndOf="parent"
38     app:layout_constraintStart_toStartOf="parent"
39     app:layout_constraintTop_toBottomOf="@+id/app_logo">
40
41 <ScrollView
42     android:layout_width="match_parent"
43     android:layout_height="match_parent">
44
45     <LinearLayout
46         android:layout_width="match_parent"
47         android:layout_height="wrap_content"
48         android:background="@color/white"
49         android:orientation="vertical"
50         android:paddingHorizontal="18dp"
51         android:paddingVertical="24dp">
52
53     <TextView
54         android:layout_width="wrap_content"
55         android:layout_height="wrap_content"
56         android:layout_gravity="center_horizontal"
57         android:text="@string/update_profile"
58         android:textColor="@color/colorPrimary"
59         android:textSize="24sp"
60         android:textStyle="bold" />
61
62     <com.google.android.material.textfield.TextInputLayout
63         android:layout_width="match_parent"
64         android:layout_height="wrap_content"
65         android:layout_marginTop="32dp"
66         android:background="@color/white"
```

```
67    android:hint="@string/username">
68
69 <EditText
70     android:id="@+id/ed_username"
71     android:layout_width="match_parent"
72     android:layout_height="wrap_content"
73     android:background="@color/white"
74     android:inputType="text"
75     android:singleLine="true" />
76 </com.google.android.material.textfield.TextInputLayout>
77
78
79 <com.google.android.material.textfield.TextInputLayout
80     android:layout_width="match_parent"
81     android:layout_height="wrap_content"
82     android:background="@color/white"
83     android:hint="@string/contact_number">
84
85 <EditText
86     android:id="@+id/ed_contact"
87     android:layout_width="match_parent"
88     android:layout_height="wrap_content"
89     android:background="@color/white"
90     android:inputType="number"
91     android:maxLength="10"
92     android:singleLine="true" />
93 </com.google.android.material.textfield.TextInputLayout>
94
95
96 <com.google.android.material.textfield.TextInputLayout
97     android:layout_width="match_parent"
98     android:layout_height="wrap_content"
99     android:background="@color/white"
100    android:hint="Date of birth">
101
102 <EditText
103     android:id="@+id/ed_dob"
104     android:layout_width="match_parent"
105     android:layout_height="wrap_content"
106     android:background="@color/white"
107     android:focusable="false"
108     android:inputType="text"
109     android:singleLine="true" />
110 </com.google.android.material.textfield.TextInputLayout>
111
112 <com.google.android.material.textfield.TextInputLayout
113     android:layout_width="match_parent"
114     android:layout_height="wrap_content"
115     android:background="@color/white"
116     android:hint="@string/e_mail">
117
118
119 <EditText
120     android:id="@+id/ed_email"
121     android:layout_width="match_parent"
122     android:layout_height="wrap_content"
123     android:background="@color/white"
124     android:enabled="false"
125     android:inputType="textEmailAddress"
126     android:singleLine="true" />
127 </com.google.android.material.textfield.TextInputLayout>
128
129
130 <TextView
131     android:id="@+id/tv_btn_login"
132     android:layout_width="match_parent"
133     android:layout_height="48dp"
```

```
135 android:layout_gravity="center_horizontal"
136 android:layout_marginTop="24dp"
137 android:background="@drawable/bg_btn"
138 android:gravity="center"
139 android:text="@string/submit"
140 android:textColor="@color/white"
141 android:textSize="18sp" />
142
143 </LinearLayout>
144
145 </ScrollView>
146
147
148 </androidx.cardview.widget.CardView>
149
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:ignore="ContentDescription,Autofill,LabelFor">

    <androidx.cardview.widget.CardView
        android:id="@+id/destCard"
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:layout_marginStart="15dp"
        android:layout_marginTop="124dp"
        android:layout_marginEnd="15dp"
        android:transitionName="dest"
        app:cardBackgroundColor="@android:color/white"
        app:cardCornerRadius="5dp"
        app:cardElevation="10dp"
        app:cardMaxElevation="10dp"
        app:cardPreventCornerOverlap="true">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="horizontal">

            <LinearLayout
                android:layout_width="50dp"
                android:layout_height="match_parent"
                android:gravity="center">

                <ImageView
                    android:layout_width="10dp"
                    android:layout_height="10dp"
                    android:layout_gravity="center"
                    android:scaleType="fitCenter"
                    android:src="@drawable/destball" />
            </LinearLayout>

            <Button
                android:id="@+id/destBtn"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:background="@android:color/transparent"
                android:gravity="start|center"
                android:text="@string/where_to"
                android:textAllCaps="false"
                android:textColor="#898888"
                android:textSize="16sp" />
        </LinearLayout>
    </androidx.cardview.widget.CardView>

    <com.sothree.slidinguppanel.SlidingUpPanelLayout
        android:id="@+id/sliding_layout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="bottom">

        <!-- MAIN CONTENT -->
        <androidx.constraintlayout.widget.ConstraintLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            tools:visibility="gone">
```

```
65
66 <fragment
67     android:id="@+id/map"
68     android:name="com.google.android.gms.maps.SupportMapFragment"
69     android:layout_width="match_parent"
70     android:layout_height="match_parent"
71     android:layout_marginTop="?attr/actionBarSize"
72     tools:context=".activityMapsActivity" />
73
74 <Button
75     android:id="@+id/btn_proceed"
76     android:layout_width="wrap_content"
77     android:layout_height="45dp"
78     android:layout_gravity="center_horizontal"
79     android:layout_marginTop="@dimen/margin_16dp"
80     android:layout_marginBottom="@dimen/margin_24dp"
81     android:background="@drawable/bg_btn"
82     android:paddingLeft="@dimen/margin_60dp"
83     android:paddingRight="@dimen/margin_60dp"
84     android:text="@string/start"
85     android:textColor="@color/white"
86     android:textSize="@dimen/margin_14sp"
87     android:visibility="gone"
88     app:layout_constraintBottom_toBottomOf="parent"
89     app:layout_constraintEnd_toEndOf="parent"
90     app:layout_constraintStart_toStartOf="parent" />
91
92 </androidx.constraintlayout.widget.ConstraintLayout>
93 <!-- SLIDING LAYOUT -->
94 <FrameLayout
95     android:id="@+id/sliderLin"
96     android:layout_width="match_parent"
97     android:layout_height="@dimen/margin_50dp">
98
99 <LinearLayout
100    android:layout_width="match_parent"
101    android:layout_height="wrap_content"
102    android:orientation="vertical">
103
104 <com tmall ultraviewpager UltraViewPager
105    android:id="@+id/recycle"
106    android:layout_width="match_parent"
107    android:layout_height="72dp"
108    android:layout_marginBottom="@dimen/margin_8dp"
109    android:background="@android:color/white"
110    android:visibility="gone" />
111
112
113 <Button
114     android:id="@+id/fetch"
115     android:layout_width="wrap_content"
116     android:layout_height="wrap_content"
117     android:layout_gravity="center"
118     android:layout_marginTop="10dp"
119     android:layout_marginBottom="20dp"
120     android:background="@drawable/bg_btn"
121     android:gravity="center"
122     android:paddingStart="60dp"
123     android:paddingEnd="60dp"
124     android:text="@string/request_ride"
125     android:textAllCaps="false"
126     android:textColor="@android:color/white"
127     android:textSize="15sp"
128     android:visibility="gone" />
129
130 </LinearLayout>
131
132
```

```
133 <LinearLayout  
134     android:layout_width="match_parent"  
135     android:layout_height="wrap_content"  
136     android:layout_margin="12dp"  
137     android:orientation="vertical"  
138     >  
139  
140     <ImageView  
141         android:id="@+id/img"  
142         android:layout_width="80dp"  
143         android:layout_height="80dp"  
144         android:layout_gravity="center"  
145         android:scaleType="fitXY"  
146         android:transitionName="img" />  
147  
148     <TextView  
149         android:id="@+id/txt"  
150         android:layout_width="match_parent"  
151         android:layout_height="wrap_content"  
152         android:layout_gravity="center"  
153         android:layout_marginTop="10dp"  
154         android:gravity="center"  
155         android:textColor="@android:color/background_dark"  
156         android:textSize="15sp" />  
157     </LinearLayout>  
158 </FrameLayout>  
159  
160 </com.sothree.slidinguppanel.SlidingUpPanelLayout>  
161  
162  
163 <LinearLayout  
164     android:id="@+id/lin_mode"  
165     android:layout_width="match_parent"  
166     android:layout_height="56dp"  
167     android:layout_marginHorizontal="@dimen/margin_8dp"  
168     android:layout_marginTop="64dp"  
169     android:visibility="gone"  
170     tools:visibility="visible">  
171  
172     <ImageView  
173         android:id="@+id	btn_car"  
174         android:layout_width="0dp"  
175         android:layout_height="wrap_content"  
176         android:layout_marginHorizontal="2dp"  
177         android:layout_weight="1"  
178         android:adjustViewBounds="true"  
179         android:background="@drawable/btn_bg_selection"  
180         android:gravity="center"  
181         android:padding="@dimen/margin_8dp"  
182         android:textColor="@drawable/btn_text_selection"  
183         android:textSize="@dimen/margin_18sp"  
184         android:textStyle="bold"  
185         app:srcCompat="@drawable/ic_car" />  
186  
187     <ImageView  
188         android:id="@+id	btn_train"  
189         android:layout_width="0dp"  
190         android:layout_height="wrap_content"  
191         android:layout_marginHorizontal="2dp"  
192         android:layout_weight="1"  
193         android:adjustViewBounds="true"  
194         android:background="@drawable/btn_bg_selection_train"  
195         android:gravity="center"  
196         android:padding="@dimen/margin_8dp"  
197         android:textColor="@drawable/btn_text_selection"  
198         android:textSize="@dimen/margin_18sp"  
199         android:textStyle="bold"  
200     </LinearLayout>
```

```
201     app:srcCompat="@drawable/ic_train" />
202
203 <ImageView
204     android:id="@+id	btn_bus"
205     android:layout_width="0dp"
206     android:layout_height="wrap_content"
207     android:layout_weight="1"
208     android:adjustViewBounds="true"
209     android:background="@drawable/btn_bg_selection_bus"
210     android:gravity="center"
211     android:layout_marginHorizontal="2dp"
212     android:padding="@dimen/margin_8dp"
213     android:textColor="@drawable/btn_text_selection"
214     android:textSize="@dimen/margin_18sp"
215     android:textStyle="bold"
216     app:srcCompat="@drawable/ic_bus" />
217
218 <ImageView
219     android:id="@+id	btn_flight"
220     android:layout_width="0dp"
221     android:layout_height="wrap_content"
222     android:layout_marginHorizontal="2dp"
223     android:layout_weight="1"
224     android:adjustViewBounds="true"
225     android:background="@drawable/btn_bg_selection_flight"
226     android:gravity="center"
227     android:padding="@dimen/margin_8dp"
228     android:textColor="@drawable/btn_text_selection"
229     android:textSize="@dimen/margin_18sp"
230     android:textStyle="bold"
231     android:visibility="gone"
232     app:srcCompat="@drawable/ic_flight" />
233
234
235 <ImageView
236     android:id="@+id	btn_cycle"
237     android:layout_width="0dp"
238     android:layout_height="wrap_content"
239     android:layout_marginHorizontal="2dp"
240     android:layout_weight="1"
241     android:adjustViewBounds="true"
242     android:background="@drawable/btn_green_bg_selection"
243     android:gravity="center"
244     android:padding="@dimen/margin_8dp"
245     android:textColor="@drawable/btn_text_selection"
246     android:textSize="@dimen/margin_18sp"
247     android:textStyle="bold"
248     android:visibility="gone"
249     app:srcCompat="@drawable/ic_bicycle" />
250
251 <ImageView
252     android:id="@+id	btn_Walking"
253     android:layout_width="0dp"
254     android:layout_height="wrap_content"
255     android:layout_marginHorizontal="2dp"
256     android:layout_weight="1"
257     android:adjustViewBounds="true"
258     android:background="@drawable/btn_green_bg_selection"
259     android:gravity="center"
260     android:padding="@dimen/margin_8dp"
261     android:textColor="@drawable/btn_text_selection"
262     android:textSize="@dimen/margin_18sp"
263     android:textStyle="bold"
264     android:visibility="gone"
265     app:srcCompat="@drawable/ic_walk" />
266
267
268
```

```
269 </LinearLayout>
270
271 <com.google.android.material.appbar.AppBarLayout
272     android:layout_width="match_parent"
273     android:layout_height="wrap_content"
274     android:theme="@style/Theme.CustomCo2.AppBarOverlay">
275
276     <androidx.appcompat.widget.Toolbar
277         android:id="@+id/toolbar"
278         android:layout_width="match_parent"
279         android:layout_height="?attr/actionBarSize"
280         android:background="?attr/colorPrimary"
281         app:popupTheme="@style/Theme.CustomCo2.PopupOverlay">
282
283
284     <RelativeLayout
285
286         android:layout_width="match_parent"
287         android:layout_height="wrap_content">
288
289         <TextView
290             android:layout_width="wrap_content"
291             android:layout_height="wrap_content"
292             android:layout_centerInParent="true"
293             android:gravity="center"
294             android:paddingRight="74dp"
295             android:text="@string/proenv_y"
296             android:textColor="@color/white"
297             android:textSize="25sp"
298             android:textStyle="bold" />
299
300         <ImageView
301             android:layout_width="wrap_content"
302             android:layout_height="wrap_content"
303             android:layout_alignParentRight="true"
304             android:layout_centerVertical="true"
305             android:layout_marginRight="@dimen/margin_16dp" />
306
307     </RelativeLayout>
308
309
310     </androidx.appcompat.widget.Toolbar>
311
312 </com.google.android.material.appbar.AppBarLayout>
313
314 </FrameLayout>
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     android:id="@+id/l_l_marker_right"
5     android:layout_width="350dp"
6     android:layout_height="40dp"
7     android:layout_margin="5dp"
8     android:orientation="horizontal">
9
10 <androidx.cardview.widget.CardView
11     android:layout_width="match_parent"
12     android:layout_height="match_parent"
13     android:background="@drawable/shadow"
14     android:padding="0dp"
15     app:cardBackgroundColor="@android:color/transparent"
16     app:cardCornerRadius="6dp"
17     app:cardElevation="10dp"
18     app:cardMaxElevation="10dp"
19     app:cardPreventCornerOverlap="true"
20     app:cardUseCompatPadding="false">
21
22     <LinearLayout
23         android:layout_width="match_parent"
24         android:layout_height="match_parent"
25         android:layout_gravity="center"
26         android:background="@drawable/shadow"
27         android:gravity="center"
28         android:orientation="horizontal"
29         android:weightSum="1">
30
31         <androidx.cardview.widget.CardView
32             android:layout_width="0dp"
33             android:layout_height="match_parent"
34             android:layout_weight="0.4"
35             android:background="#262626"
36             android:visibility="gone"
37             app:cardBackgroundColor="#262626"
38             app:cardElevation="10dp"
39             app:cardMaxElevation="10dp">
40
41             <TextView
42                 android:id="@+id/tv_timing_right"
43                 android:layout_width="match_parent"
44                 android:layout_height="match_parent"
45                 android:gravity="center_vertical|center"
46                 android:maxLines="4"
47                 android:padding="4dp"
48                 android:textColor="@android:color/white"
49                 android:textSize="12sp" />
50         </androidx.cardview.widget.CardView>
51
52         <androidx.cardview.widget.CardView
53             android:layout_width="0dp"
54             android:layout_height="match_parent"
55             android:layout_weight="0.6"
56             app:cardBackgroundColor="@android:color/white"
57             app:cardCornerRadius="6dp"
58             app:cardElevation="10dp"
59             app:cardMaxElevation="10dp">
60
61             <TextView
62                 android:id="@+id/tv_address_right"
63                 android:layout_width="match_parent"
64                 android:layout_height="match_parent"
65                 android:ellipsize="end"
66                 android:gravity="center_vertical"
```

```
68  
69     android:maxLines="4"  
70     android:padding="5dp"  
71     android:textColor="@android:color/background_dark"  
72     android:textSize="12sp" />  
73   </androidx.cardview.widget.CardView>  
74 </LinearLayout>  
75 </androidx.cardview.widget.CardView>  
  
</LinearLayout>
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:app="http://schemas.android.com/apk/res-auto"
4     android:layout_width="match_parent"
5     android:layout_height="wrap_content"
6     android:padding="16dp">
7
8
9   <TextView
10    android:id="@+id/tv_forgot_title"
11    android:layout_width="match_parent"
12    android:layout_height="wrap_content"
13    android:layout_gravity="center_horizontal"
14    android:layout_marginTop="24dp"
15    android:gravity="center"
16    android:text="@string/forgot_password"
17    android:textColor="@color/colorPrimaryDark"
18    android:textSize="22sp"
19    app:layout_constraintEnd_toEndOf="parent"
20    app:layout_constraintStart_toStartOf="parent"
21    app:layout_constraintTop_toTopOf="parent" />
22
23   <TextView
24     android:id="@+id	btn_forgot_password"
25     android:layout_width="208dp"
26     android:layout_height="48dp"
27     android:layout_marginBottom="24dp"
28     android:layout_gravity="center_horizontal"
29     android:layout_marginTop="48dp"
30     android:background="@drawable/bg_btn"
31     android:gravity="center"
32     android:text="@string/submit"
33     android:textColor="@color/white"
34     android:textSize="18sp"
35     app:layout_constraintEnd_toEndOf="parent"
36     app:layout_constraintStart_toStartOf="parent"
37     app:layout_constraintTop_toBottomOf="@+id/til_1" />
38
39
40   <com.google.android.material.textfield.TextInputLayout
41     android:id="@+id/til_1"
42     android:layout_width="match_parent"
43     android:layout_height="wrap_content"
44     android:layout_marginTop="24dp"
45     android:background="@color/white"
46     android:hint="@string/e_email"
47     app:layout_constraintEnd_toEndOf="parent"
48     app:layout_constraintStart_toStartOf="parent"
49     app:layout_constraintTop_toBottomOf="@+id/tv_forgot_title">
50
51
52   <EditText
53     android:id="@+id/ed_email"
54     android:layout_width="match_parent"
55     android:layout_height="wrap_content"
56     android:background="@color/white"
57     android:inputType="textEmailAddress"
58     android:singleLine="true" />
59   </com.google.android.material.textfield.TextInputLayout>
60
61 </androidx.constraintlayout.widget.ConstraintLayout>
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="wrap_content"
5     android:background="@color/white"
6     android:gravity="center"
7     android:orientation="vertical"
8     android:padding="@dimen/margin_16dp">
9
10    <TextView
11        android:layout_width="wrap_content"
12        android:layout_height="wrap_content"
13        android:layout_marginTop="@dimen/margin_8dp"
14        android:text="@string/error"
15        android:textColor="@color/colorPrimaryDark"
16        android:textSize="20sp"
17        android:textStyle="bold" />
18
19    <TextView
20        android:layout_width="wrap_content"
21        android:layout_height="wrap_content"
22        android:layout_marginTop="@dimen/margin_24dp"
23        android:gravity="center"
24        android:text="@string/trouble_finding_a_route_please_try_another_location"
25        android:textSize="18sp" />
26
27
28    <TextView
29        android:id="@+id	btn_submit"
30        android:layout_width="match_parent"
31        android:layout_height="48dp"
32        android:layout_gravity="center_horizontal"
33        android:layout_margin="@dimen/margin_24dp"
34        android:background="@drawable/bg_btn"
35        android:gravity="center"
36        android:text="@string/okay"
37        android:textColor="@color/white"
38        android:textSize="18sp" />
39
40    </LinearLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_gravity="center_horizontal"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingVertical="24dp">

    <ProgressBar
        android:id="@+id/progress"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="24dp"
        android:text="@string/please_wait"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="22sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/progress"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/white"
    android:padding="@dimen/margin_16dp">

    <TextView
        android:id="@+id/txtTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="@dimen/margin_8dp"
        android:text="@string/select_date"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/txtDate"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="@dimen/margin_24dp"
        android:background="@drawable/edt_bacground"
        android:paddingHorizontal="@dimen/dimen_12dp"
        android:paddingVertical="@dimen/dimen_12dp"
        android:layout_marginHorizontal="@dimen/margin_18dp"
        android:hint="@string/select_date"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="@dimen/margin_16sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/txtTitle" />

    <TextView
        android:id="@+id	btn_submit"
        android:layout_width="0dp"
        android:layout_height="48dp"
        android:layout_gravity="center_horizontal"
        android:layout_margin="@dimen/margin_24dp"
        android:background="@drawable/bg_btn"
        android:gravity="center"
        android:text="@string/submit"
        android:textColor="@color/white"
        android:textSize="18sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/txtDate" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/white"
    android:padding="@dimen/margin_16dp">

    <TextView
        android:id="@+id/txtTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="@dimen/margin_8dp"
        android:text="@string/trip_completed"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/txtsourceTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="@dimen/margin_24dp"
        android:text="@string/source_address"
        android:textSize="18sp"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/txtTitle" />

    <TextView
        android:id="@+id/txtsourcAddress"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="@dimen/margin_4dp"
        android:text="Source"
        android:textColor="@color/black"
        android:textSize="12sp"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/txtsourceTitle" />

    <TextView
        android:id="@+id/txtDestinationTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="@dimen/margin_8dp"
        android:text="@string/destination_address"
        android:textSize="18sp"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/txtsourcAddress" />

    <TextView
        android:id="@+id/txtDestinationAddress"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="@dimen/margin_4dp"
        android:text="@string/destination"
        android:textColor="@color/black"
        android:textSize="12sp"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/txtDestinationTitle" />

    <TextView
        android:id="@+id/txtDestenceTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="@dimen/margin_8dp"
        android:text="@string/distance"
        android:textSize="18sp"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/txtDestinationAddress" />

    <TextView
        android:id="@+id/txtDestence"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="@dimen/margin_4dp"
        android:text="@string/_5_km"
        android:textColor="@color/black"
        android:textSize="12sp"
        android:textStyle="bold"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/txtDestenceTitle" />

    <Spinner
        android:id="@+id/spVehicleType"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

        android:layout_marginTop="@dimen/dimen_4dp"
        android:background="@drawable/spinner_background"
        android:entries="@array/vehicleType"
        android:paddingVertical="@dimen/margin_8dp" />
```

```
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/txtDestence" />
```

<Spinner

```
    android:id="@+id/spFuelType"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="@dimen/margin_8dp"
    android:background="@drawable/spinner_background"
    android:entries="@array/FuleType"
    android:paddingVertical="@dimen/margin_8dp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/spVehicleType" />
```

<TextView

```
    android:id="@+id/txtCO2Title"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="@dimen/margin_8dp"

    android:text="@string/total_co2_kg"
    android:textSize="18sp"
    android:textStyle="bold"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/spFuelType" />
```

<TextView

```
    android:id="@+id/txtCO2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="@dimen/margin_4dp"
    android:text="@string/_5_km"
    android:textColor="@color/black"
    android:textSize="12sp"
    android:textStyle="bold"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/txtCO2Title" />
```

<TextView

```
    android:id="@+id/txtdec1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="@dimen/margin_4dp"
    android:text="@string/disclaimer_for_electric_phev_cars_the_following_results_are_based_on_the_assumption_that_the_energy_mix_to_charge_the_vehicle_is_certified_green_electricity"
    android:textColor="@color/colorRed"
    android:textSize="12sp"
    android:visibility="gone"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/txtCO2" />
```

<TextView

```
    android:id="@+id/txtdec2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="@dimen/margin_8dp"

    android:text="@string/disclaimer_the_following_results_are_an_approximation_and_may_notRepresent_the_actual_co2_emissions_as_it_may_vary_from_model_to_model_and_manufacture_year"
```

```
    android:textColor="@color/colorRed"
    android:textSize="12sp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/txtdec1" />
```

<TextView

```
    android:id="@+id/btn_submit"
    android:layout_width="0dp"
    android:layout_height="48dp"
    android:layout_gravity="center_horizontal"
    android:layout_margin="@dimen/margin_24dp"
    android:background="@drawable/bg_btn"
    android:gravity="center"
    android:text="@string/submit"
    android:textColor="@color/white"
    android:textSize="18sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/txtdec2" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:app="http://schemas.android.com/apk/res-auto"
4   xmlns:tools="http://schemas.android.com/tools"
5   android:layout_width="match_parent"
6   android:layout_height="wrap_content"
7
8   app:cardElevation="2dp"
9   app:cardUseCompatPadding="true">
10
11 <androidx.constraintlayout.widget.ConstraintLayout
12   android:layout_width="match_parent"
13   android:background="#08000000"
14   android:padding="@dimen/margin_8dp"
15   android:layout_height="wrap_content"
16   android:orientation="vertical">
17
18   <TextView
19     android:id="@+id/text1"
20     android:layout_width="match_parent"
21     android:layout_height="wrap_content"
22     android:singleLine="true"
23     android:textAlignment="viewStart"
24     android:textAppearance="?android:attr/textAppearanceListItem"
25     android:visibility="gone"
26     app:layout_constraintEnd_toEndOf="parent"
27     app:layout_constraintStart_toStartOf="parent"
28     app:layout_constraintTop_toTopOf="parent"
29     tools:visibility="visible" />
30
31   <TextView
32     android:id="@+id/text2"
33     android:layout_width="match_parent"
34     android:layout_height="wrap_content"
35     android:layout_marginTop="2dp"
36     android:textAlignment="viewStart"
37     android:textAppearance="?android:attr/textAppearanceSmall"
38     app:layout_constraintEnd_toEndOf="parent"
39     app:layout_constraintStart_toStartOf="parent"
40     app:layout_constraintTop_toBottomOf="@+id/text1" />
41   </androidx.constraintlayout.widget.ConstraintLayout>
42
43 </androidx.cardview.widget.CardView>
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="wrap_content"
7     app:cardCornerRadius="@dimen/dimen_4dp"
8     app:cardElevation="4dp"
9     app:cardUseCompatPadding="true">
10
11
12 <androidx.constraintlayout.widget.ConstraintLayout
13     android:layout_width="match_parent"
14     android:layout_height="wrap_content"
15     android:padding="@dimen/margin_12dp">
16
17     <TextView
18         android:id="@+id/txtsourceTitle"
19         android:layout_width="wrap_content"
20         android:layout_height="wrap_content"
21         android:text="@string/source"
22         android:textColor="@color/black"
23         android:textSize="@dimen/margin_16sp"
24         android:textStyle="bold"
25         app:layout_constraintStart_toStartOf="parent"
26         app:layout_constraintTop_toTopOf="parent" />
27
28     <TextView
29         android:id="@+id/txtsource"
30         android:layout_width="wrap_content"
31         android:layout_height="wrap_content"
32         android:textSize="@dimen/margin_16sp"
33         app:layout_constraintStart_toStartOf="parent"
34         app:layout_constraintTop_toBottomOf="@+id/txtsourceTitle"
35         tools:text="Source" />
36
37     <TextView
38         android:id="@+id/txtDestinationTitle"
39         android:layout_width="wrap_content"
40         android:layout_height="wrap_content"
41         android:layout_marginTop="@dimen/margin_8dp"
42         android:text="@string/destination"
43         android:textColor="@color/black"
44         android:textSize="@dimen/margin_16sp"
45         android:textStyle="bold"
46         app:layout_constraintStart_toStartOf="parent"
47         app:layout_constraintTop_toBottomOf="@+id/txtsource" />
48
49     <TextView
50         android:id="@+id/txtDestination"
51         android:layout_width="wrap_content"
52         android:layout_height="wrap_content"
53         android:textSize="@dimen/margin_16sp"
54         app:layout_constraintStart_toStartOf="parent"
55         app:layout_constraintTop_toBottomOf="@+id/txtDestinationTitle"
56         tools:text="Destination" />
57
58
59
60     <TextView
61         android:id="@+id/txtVehicleTitle"
62         android:layout_width="wrap_content"
63         android:layout_height="wrap_content"
64         android:layout_marginTop="@dimen/margin_8dp"
65         android:text="@string/ride_type"
```

```
67 android:textColor="@color/black"
68 android:textSize="@dimen/margin_16sp"
69 android:textStyle="bold"
70 app:layout_constraintStart_toStartOf="parent"
71 app:layout_constraintTop_toBottomOf="@+id/txtDestination" />
72
73 <TextView
74     android:id="@+id/txtVehicle"
75     android:layout_width="wrap_content"
76     android:layout_height="wrap_content"
77     android:textSize="@dimen/margin_16sp"
78     app:layout_constraintStart_toStartOf="parent"
79     app:layout_constraintTop_toBottomOf="@+id/txtVehicleTitle"
80     tools:text="Vehicle" />
81
82 <View
83     android:id="@+id/view"
84     android:layout_width="match_parent"
85     android:layout_height="1dp"
86     android:layout_marginTop="@dimen/margin_8dp"
87     android:background="@color/black"
88     app:layout_constraintEnd_toEndOf="parent"
89     app:layout_constraintStart_toStartOf="parent"
90     app:layout_constraintTop_toBottomOf="@+id/txtVehicle" />
91
92 <TextView
93     android:id="@+id/txtDistanceTitle"
94     android:layout_width="wrap_content"
95     android:layout_height="wrap_content"
96     android:layout_marginTop="@dimen/margin_8dp"
97     android:text="@string/distance"
98     android:textColor="@color/black"
99     android:textSize="@dimen/margin_16sp"
100    android:textStyle="bold"
101    app:layout_constraintStart_toStartOf="parent"
102    app:layout_constraintTop_toBottomOf="@+id/view" />
103
104 <TextView
105     android:id="@+id/txtDistance"
106     android:layout_width="wrap_content"
107     android:layout_height="wrap_content"
108     android:textSize="@dimen/margin_16sp"
109     app:layout_constraintStart_toStartOf="parent"
110     app:layout_constraintTop_toBottomOf="@+id/txtDistanceTitle"
111     tools:text="Distance" />
112
113 <TextView
114     android:id="@+id/txtCo2Title"
115     android:layout_width="wrap_content"
116     android:layout_height="wrap_content"
117     android:layout_marginTop="@dimen/margin_8dp"
118     android:text="@string/co2_kg"
119     android:textColor="@color/black"
120     android:textSize="@dimen/margin_16sp"
121     android:textStyle="bold"
122     app:layout_constraintEnd_toEndOf="parent"
123     app:layout_constraintTop_toBottomOf="@+id/view" />
124
125 <TextView
126     android:id="@+id/txtCO2"
127     android:layout_width="wrap_content"
128     android:layout_height="wrap_content"
129     android:textSize="@dimen/margin_16sp"
130     app:layout_constraintEnd_toEndOf="parent"
131     app:layout_constraintTop_toBottomOf="@+id/txtDistanceTitle"
132     tools:text="Source" />
133
134
```

```
135 <TextView  
136     android:id="@+id/txtDate"  
137     android:layout_width="wrap_content"  
138     android:layout_height="wrap_content"  
139     android:textSize="@dimen/margin_16sp"  
140     app:layout_constraintEnd_toEndOf="parent"  
141     app:layout_constraintTop_toTopOf="parent"  
142     app:layout_constraintTop_toBottomOf="@+id/txtDistanceTitle"  
143     tools:text="Source" />  
144  
145 </androidx.constraintlayout.widget.ConstraintLayout>  
</androidx.cardview.widget.CardView>
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     android:layout_width="match_parent"
5     android:layout_height="wrap_content"
6     android:background="@color/colorPrimaryDark"
7     android:gravity="center"
8     android:orientation="horizontal"
9     android:paddingLeft="@dimen/margin_16dp"
10    android:paddingTop="@dimen/margin_32dp"
11    android:paddingRight="@dimen/margin_16dp"
12    android:paddingBottom="@dimen/margin_16dp">
13
14 <de.hdodenhof.circleimageview.CircleImageView
15     android:id="@+id/img_logo"
16     android:layout_width="84dp"
17     android:layout_height="84dp"
18     android:layout_marginTop="24dp"
19     android:contentDescription="@string/nav_header_desc"
20     android:src="@drawable/app_icon"
21     app:civ_border_color="@color/white"
22     app:civ_border_width="2dp"
23     app:layout_constraintEnd_toEndOf="parent"
24     app:layout_constraintStart_toStartOf="parent"
25     app:layout_constraintTop_toTopOf="parent" />
26
27
28 <TextView
29     android:id="@+id/tvUsername"
30     android:layout_width="wrap_content"
31     android:layout_height="wrap_content"
32     android:layout_marginTop="4dp"
33     android:paddingTop="@dimen/nav_header_vertical_spacing"
34     android:text=""
35     android:textColor="@color/white"
36     android:textSize="@dimen/margin_16sp"
37     android:textStyle="bold"
38     app:layout_constraintEnd_toEndOf="parent"
39     app:layout_constraintStart_toStartOf="parent"
40     app:layout_constraintTop_toBottomOf="@+id/img_logo" />
41
42 <TextView
43     android:id="@+id/tvEmail"
44     android:layout_width="wrap_content"
45     android:layout_height="wrap_content"
46     android:layout_marginTop="4dp"
47     android:layout_marginBottom="@dimen/margin_8dp"
48     android:paddingTop="@dimen/nav_header_vertical_spacing"
49     android:text=""
50     android:textAppearance="@style/TextAppearance.AppCompat.Body1"
51     android:textColor="@color/white"
52     android:textSize="@dimen/margin_14sp"
53     app:layout_constraintBottom_toBottomOf="parent"
54     app:layout_constraintEnd_toEndOf="parent"
55     app:layout_constraintStart_toStartOf="parent"
56     app:layout_constraintTop_toBottomOf="@+id/tvUsername" />
57
58
59 </androidx.constraintlayout.widget.ConstraintLayout>
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:background="@color/colorWhite"
5     android:id="@+id/lin_set_location"
6     android:layout_height="45dp"
7     android:orientation="vertical">
8
9     <LinearLayout
10         android:layout_width="match_parent"
11         android:layout_height="44.2dp"
12         android:orientation="horizontal">
13
14
15         <TextView
16             android:layout_width="match_parent"
17             android:layout_height="wrap_content"
18             android:layout_gravity="center"
19             android:layout_marginStart="15dp"
20             android:text="@string/set_location_on_map"
21             android:textColor="@android:color/background_dark"
22             android:textSize="15sp" />
23     </LinearLayout>
24
25     <TextView
26         android:layout_width="match_parent"
27         android:layout_height="2dp"
28         android:background="@android:color/background_dark" />
29
30 </LinearLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<!--Side drawer menu item-->
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    tools:showIn="navigation_view">

    <group android:checkableBehavior="single">
        <item
            android:id="@+id/nav_home"
            android:icon="@drawable/ic_home"
            android:title="@string/home" />
        <item
            android:id="@+id/nav_Score_Board"
            android:icon="@drawable/ic_trophy"
            android:title="@string/score_board" />
        <item
            android:id="@+id/nav_Profile"
            android:icon="@drawable/ic_profile"
            android:title="@string/profile" />
        <item
            android:id="@+id/nav_Change_Password"
            android:icon="@drawable/ic_profile"
            android:title="@string/change_password" />
        <item
            android:id="@+id/nav_Logout"
            android:icon="@drawable/ic_logout"
            android:title="@string/logout" />
    </group>

</menu>
```

```
<?xml version="1.0" encoding="utf-8"?>
<adaptive-icon xmlns:android="http://schemas.android.com/apk/res/android">
    <background android:drawable="@drawable/ic_launcher_background" />
    <foreground android:drawable="@mipmap/ic_launcher_foreground" />
</adaptive-icon>
```

```
<?xml version="1.0" encoding="utf-8"?>
<adaptive-icon xmlns:android="http://schemas.android.com/apk/res/android">
    <background android:drawable="@drawable/ic_launcher_background" />
    <foreground android:drawable="@mipmap/ic_launcher_foreground" />
</adaptive-icon>
```

```
<?xml version="1.0" encoding="utf-8"?>
<transitionSet >
  <changeTransform />
  <changeBounds />
</transitionSet>
```

```
<?xml version="1.0" encoding="utf-8"?>
<!--Resource color file to app global colors-->
<resources>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFF</color>

    <color name="colorPrimary">#69BE94</color>
    <color name="colorPrimaryDark">#0D98BA</color>
    <color name="colorWhite">#FFFFFF</color>
    <color name="colorButton">#4695B8</color>
    <color name="colorBlueLight">#04417E</color>

    <color name="colorRed">#FF0000</color>
</resources>
```

```
<resources>
```

```
    <!-- Default screen margins, per the Android Design guidelines. -->
```

```
    <dimen name="nav_header_vertical_spacing">8dp</dimen>
```

```
    <dimen name="margin_16dp">16dp</dimen>
```

```
    <dimen name="margin_32dp">32dp</dimen>
```

```
    <dimen name="margin_50dp">50dp</dimen>
```

```
    <dimen name="margin_60dp">60dp</dimen>
```

```
    <dimen name="margin_8dp">8dp</dimen>
```

```
    <dimen name="margin_14sp">14sp</dimen>
```

```
    <dimen name="margin_16sp">16sp</dimen>
```

```
    <dimen name="margin_18sp">18sp</dimen>
```

```
    <dimen name="margin_18dp">18dp</dimen>
```

```
    <dimen name="margin_24dp">24dp</dimen>
```

```
    <dimen name="margin_4dp">4dp</dimen>
```

```
    <dimen name="margin_12dp">12dp</dimen>
```

```
    <dimen name="dimen_12dp">12dp</dimen>
```

```
    <dimen name="dimen_4dp">4dp</dimen>
```

```
</resources>
```

```

<!--Common string listed-->
<resources>
    <string name="app_name">PROENV- Y</string>

    <string name="Map_Api_Key">AlzaSyCIEy7iOrMOyD7_hLoALTz6o_1vur7mYc</string>
    <string name="navigation_drawer_open">Open navigation drawer</string>
    <string name="navigation_drawer_close">Close navigation drawer</string>
    <string name="nav_header_desc">Navigation header</string>
    <string name="title_activity_main">HomeActivity</string>

    <string name="home">Home</string>
    <string name="score_board">Statistic</string>
    <string name="profile">Profile</string>
    <string name="logout">Logout</string>
    <string name="change_password">Change Password</string>
    <string name="current_password">Current Password</string>
    <string name="new_password">New Password</string>
    <string name="confirm_password">Confirm Password</string>
    <string name="submit">Submit</string>
    <string name="don_t_have_an_account">Don't have an Account ?</string>
    <string name="register">Register</string>
    <string name="login">Login</string>
    <string name="e_mail">E-mail</string>
    <string name="password">Password</string>
    <string name="forgot_password">Forgot Password ?</string>
    <string name="set_pick_up">Set Pick Up Location</string>
    <string name="done">Done</string>
    <string name="already_have_an_account">Already have an account ?</string>
    <string name="username">Username</string>
    <string name="contact_number">Contact number</string>
    <string name="date_of_birth">Date of birth</string>
    <string name="current_location">Current Location</string>
    <string name="where_to">Where to?</string>
    <string name="statistic">Statistics</string>
    <string name="total_co2">Total CO2(Kg) </string>
    <string name="_0">0</string>
    <string name="total_calories">Total Calories</string>
    <string name="no_data_available">No trip history available </string>
    <string name="update_profile">Update Profile</string>
    <string name="start">Start</string>
    <string name="request_ride">Request Ride</string>
    <string name="proenv_y">PROENV-Y</string>
    <string name="trouble_finding_a_route_please_try_another_location">Trouble finding a route, please try another location.</string>
    <string name="error">Error</string>
    <string name="okay">Okay</string>
    <string name="please_wait">Please Wait</string>
    <string name="select_date">Select Date</string>
    <string name="trip_completed">Trip Completed</string>
    <string name="source_address">Source Address</string>
    <string name="destination_address">Destination Address</string>
    <string name="destination">Destination</string>
    <string name="distance">Distance</string>
    <string name=".5_km">.5 KM</string>
    <string name="total_co2_kg">Total Co2(Kg)</string>
    <string name="disclaimer_for_electric_phew_cars_the_following_results_are_based_on_the_assumption_that_the_energy_mix_to_charge_the_vehicle_is_certified_green_electricity">Disclaimer - The following results are based on the assumption that the energy mix to charge the vehicle is CERTIFIED GREEN ELECTRICITY.</string>
    <string name="disclaimer_the_following_results_are_an_approximation_and_may_notRepresent_the_actual_co2_emissions_as_it_may_vary_from_model_to_model_and_manufacture_year">Disclaimer - The following results are an approximation and may not represent the actual co2 emissions as it may vary from model to model and manufacture year.</string>
    <string name="source">Source</string>
    <string name="ride_type">Ride Type</string>
    <string name="co2_kg">CO2(kg)</string>
    <string name="set_location_on_map">Set Location on Map</string>

    <array name="vehicleType">
        <item>Compact car</item>
        <item>Mid range</item>
        <item>Luxury/SUV/Van</item>
    </array>

    <array name="FuleType">
        <item>Petrol</item>
        <item>Diesel</item>
        <item>Hybrid</item>
        <item>PHEV</item>
        <item>Electric</item>
    </array>

```

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Theme.CustomCo2" parent="Theme.MaterialComponents.Light.DarkActionBar">
        <!-- Primary brand color. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryVariant">@color/colorPrimaryDark</item>
        <item name="colorOnPrimary">@color/white</item>
        <item name="colorAccent">@color/colorPrimary</item>
        <!-- Secondary brand color. -->
        <item name="colorSecondary">@color/colorWhite</item>
        <item name="colorSecondaryVariant">@color/teal_700</item>
        <item name="colorOnSecondary">@color/black</item>
        <!-- Status bar color. -->
        <item name="android:statusBarColor" tools:targetApi="I">?attr/colorPrimaryVariant</item>

    </style>

    <!--No action bar theme-->
    <style name="Theme.CustomCo2.NoActionBar">
        <item name="windowActionBar">false</item>
        <item name="windowNoTitle">true</item>
    </style>

    <!--dark action bar theme-->
    <style name="Theme.CustomCo2.AppBarOverlay" parent="ThemeOverlay.AppCompat.Dark.ActionBar" />

    <!--light action bar theme-->
    <style name="Theme.CustomCo2.PopupOverlay" parent="ThemeOverlay.AppCompat.Light" />

</resources>
```

```
plugins {
    id 'com.android.application'
}

android {
    compileSdkVersion 29
    buildToolsVersion "30.0.1"

    defaultConfig {
        applicationId "com.custom.co2"
        minSdkVersion 21
        targetSdkVersion 29
        versionCode 1
        versionName "1.0"
        multiDexEnabled true
        vectorDrawables.useSupportLibrary = true
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'),
            'proguard-rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
}

dependencies {

    implementation 'androidx.appcompat:appcompat:1.2.0'
    implementation 'com.google.android.material:material:1.2.1'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
    implementation 'com.google.android.material:material:1.3.0-rc01'
    implementation 'androidx.navigation:navigation-fragment:2.2.2'
    implementation 'androidx.navigation:navigation-ui:2.2.2'
    implementation 'androidx.lifecycle:lifecycle-livedata-ktx:2.2.0'
    implementation 'androidx.lifecycle:lifecycle-viewmodel-ktx:2.2.0'
    testImplementation 'junit:junit:4.13.1'
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'
    implementation 'com.google.firebaseio:firebase-analytics'
    implementation platform('com.google.firebase:firebase-bom:26.2.0')
    implementation 'com.google.firebaseio:firebase-firebase'
    implementation 'com.google.android.gms:play-services-maps:17.0.0'
    implementation 'com.google.android.gms:play-services-location:17.1.0'
    implementation 'com.google.android.gms:play-services-places:17.0.0'

    implementation('com.alibaba.android:ultraviewpager:1.0.7.7@aar') {
        transitive = true
    }
    implementation 'com.google.maps.android:android-maps-utils:0.5'

    //circle image
}
```

```
implementation 'de.hdodenhof:circleimageview:3.0.0'
implementation 'com.sothree.slidinguppanel:library:3.4.0'
implementation 'com.github.tintinscorpion:Dual-color-Polyline-Animation:1.0'

//Place API
implementation 'com.google.android.libraries.places:places:2.0.0'
}
apply plugin: 'com.android.application'
apply plugin: 'com.google.gms.google-services'
```

```
// Top-level build file where you can add configuration options common to all sub-
projects/modules.
buildscript {
    repositories {
        google()
        jcenter()
    }
    dependencies {
        classpath "com.android.tools.build:gradle:3.3.2"
        classpath 'com.google.gms:google-services:4.3.4'

        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
    }
}

allprojects {
    repositories {
        google()
        jcenter()
        maven { url "https://jitpack.io" }
    }
}

task clean(type: Delete) {
    delete rootProject.buildDir
}
```

#Fri Jan 22 10:34:58 PST 2021

distributionBase=GRADLE_USER_HOME

distributionPath=wrapper/dists

zipStoreBase=GRADLE_USER_HOME

zipStorePath=wrapper/dists

distributionUrl=https://services.gradle.org/distributions/gradle-5.4.1-all.zip

```
# Add project specific ProGuard rules here.  
# You can control the set of applied configuration files using the  
# proguardFiles setting in build.gradle.  
#  
# For more details, see  
# http://developer.android.com/guide/developing/tools/proguard.html
```

```
# If your project uses WebView with JS, uncomment the following  
# and specify the fully qualified class name to the JavaScript interface  
# class:  
#-keepclassmembers class fqcn.of.javascript.interface.for.webview {  
#   public *;  
#}
```

```
# Uncomment this to preserve the line number information for  
# debugging stack traces.  
#-keepattributes SourceFile,LineNumberTable
```

```
# If you keep the line number information, uncomment this to  
# hide the original source file name.  
#-renamesourcefileattribute SourceFile
```

```
{  
  "project_info": {  
    "project_number": "803694163387",  
    "project_id": "customco2",  
    "storage_bucket": "customco2.appspot.com"  
  },  
  "client": [  
    {  
      "client_info": {  
        "mobilesdk_app_id": "1:803694163387:android:e5ce1a39e755905954809e",  
        "android_client_info": {  
          "package_name": "com.custom.co2"  
        }  
      },  
      "oauth_client": [  
        {  
          "client_id": "803694163387-4dtdm7dr7q4cj3r16ipeqnpe5ouglgkt.apps.googleusercontent.com",  
          "client_type": 1,  
          "android_info": {  
            "package_name": "com.custom.co2",  
            "certificate_hash": "958b4d304cabc71db80c052914b4f1510fe1c0e8"  
          }  
        },  
        {  
          "client_id": "803694163387-v0f7ot82e65oacghrhd4bueliplqltms.apps.googleusercontent.com",  
          "client_type": 3  
        }  
      ],  
      "api_key": [  
        {  
          "current_key": "AlzaSyCIEy7iOrtMOyD7_hLoALTz6o_1vur7mYc"  
        }  
      ],  
      "services": {  
        "appinvite_service": {  
          "other_platform_oauth_client": [  
            {  
              "client_id": "803694163387-v0f7ot82e65oacghrhd4bueliplqltms.apps.googleusercontent.com",  
              "client_type": 3  
            }  
          ]  
        }  
      }  
    ],  
    "configuration_version": "1"  
  }  
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.custom.co2">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />

    <uses-feature
        android:glEsVersion="0x00020000"
        android:required="true" />

    <application
        android:name=".utils.App"
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.CustomCo2">
        <activity
            android:name=".activity.HomeActivity"
            android:label="@string/title_activity_main"
            android:theme="@style/Theme.CustomCo2.NoActionBar"/>
        <activity android:name=".activity.RegisterActivity" />
        <activity android:name=".activity.LoginActivity" />
        <activity android:name=".activity.PickLocation" />
        <activity android:name=".activity.UpdateProfileActivity" />
        <activity android:name=".activity.StatisticsActivity" />
        <activity android:name=".activity.changepasswordactivity" />
        <activity android:name=".activity.SplashActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity
            android:name=".activity.SearchActivity"

        android:configChanges="mcc|mnc|locale|keyboard|keyboardHidden|screenLayout|fontScale|uiMode|orientation|screenSize|layoutDirection"

            android:screenOrientation="unspecified"
            tools:replace="android:screenOrientation" />
        <meta-data
            android:name="com.google.android.gms.version"
            android:value="@integer/google_play_services_version" />
        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="@string/Map_Api_Key" />
    </application>

</manifest>
```