DESIGN AND IMPLEMENTATION OF MOBILE APPLICATION

Design Document

Authors:
Matteo VELATI - 920929
Niccolò ZANGRANDO - 919939

Professor:
Luciano BARESI

Delivery date:
15/01/2021

Table of Contents

1 Introduction

1.1 Purpose of this document

In this document we are going to describe software design and architecture of the *Triplan* system.
In particular how the database, the application server and the mobile application have been developed and the interactions among them.

1.2 Scope

The main scope of *Triplan* application is to provide an all-in-one solution to organize and manage a trip.

In order to achieve the scope, *Triplan* offers different functionalities that can be accessed through the mobile application.

- ***Registration*** A user, in order to access all the functionalities, has to register a new account. To simplify this procedure we allow a user to sign in directly with his *Google* or *Facebook* account.

- ***Login*** A user can access the application with his credentials or through *Google* or *Facebook* account.

- ***Trip creation*** A user can create a new trip specifying the trip period and the destination.

- ***Trip editing*** A user can edit a trip already created.

- ***Trip deletion*** A user can delete a trip already created.

- ***Trip join*** A user, through an auto-generated link, can join a trip already created by another user.

- ***Weather*** 7-days forecasts with minimum and maximum temperature for each day.

- ***Trip list*** A user can create and customize the list with all the things needed during his trip. In particular a user can add or remove an element from the list.

- ***Note list*** A user can create and customize the notes of the trip.

- ***Map visualization*** A user can visualize the trip map and can customize it by adding, editing and removing different points of interest (POI).

- ***POI creation*** A user can create a new POI on the map through a search bar and saving it specifying a title and a symbol.

- ***POI editing*** A user can edit an already created POI by modifying its title or symbol.

- ***POI deletion*** A user can delete an already created POI.

- ***Expense management*** A user can manage all the trip expenses by adding, editing and removing different expenses of the trip and can check the total amount of his bill.

- ***Expense creation*** A user can create a new expense of the trip by specifying the date, import, if it is a refund or not, and for which among the participants to the trip he has paid for.

- ***Expense editing*** A user can edit an expense already created.

- ***Expense deletion*** A user can delete an expense already created.

- ***Bill visualization*** A user can visualize the bill of all the participants in order to understand immediately all the credits or debits accumulated during the trip among all the participants.

In order to be able to access all the functionalities an internet connection is required. Otherwise a user will be able only to visualize the last data fetched from the server.

2 Architectural Design

2.1 Overview

The need to design a system which allows communications with agents such as users and external systems led us to decide to use a client-server architectural approach.
We decide to implement a three-tier architecture which includes:

- *Presentation Tier*
- *Logic Tier*
- *Data Tier*

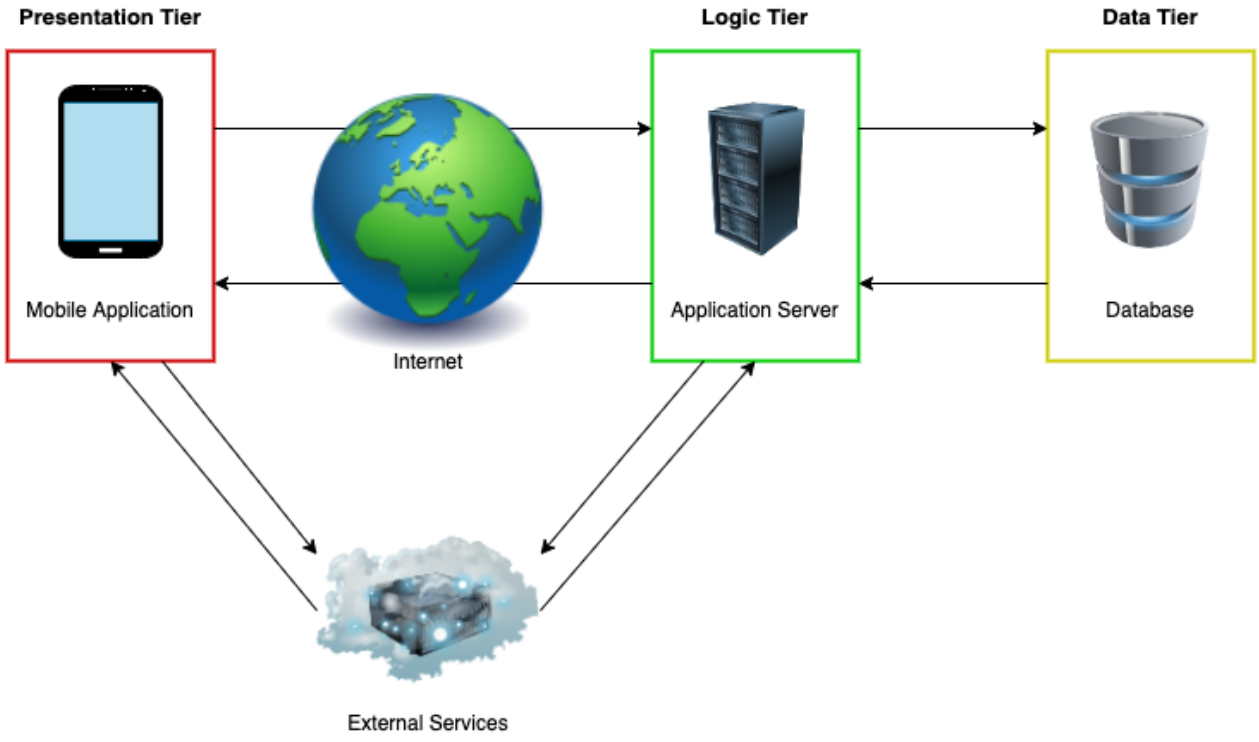


Figure 1: 3-tier architecture

To clarify at a finer level how the components are mapped in the three-tier architecture is now explained the division introduced in the diagram above:

- ***Mobile Application*** component used by the user in order to access the functionalities offered by *Triplan* system.

- ***Application Server*** component which realizes the functionalities offered by *Triplan* system.

- ***Database*** component which stores and manages the access to the data produced and needed by the *Application Server*.

2.2 Database

2.2.1 Database Design

The following diagram shows the database design of *Triplan*, in particular the entities and the relations among them.

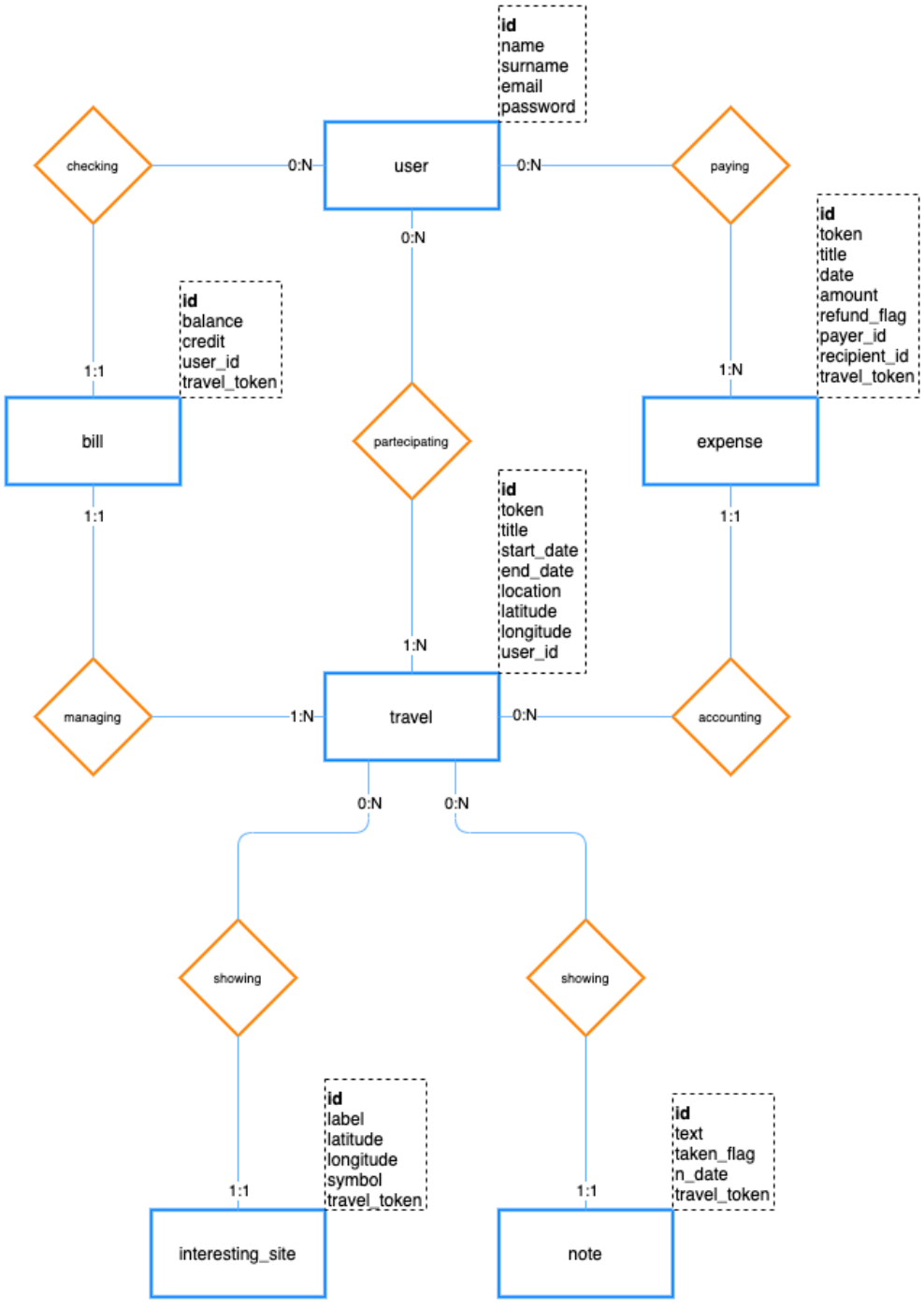


Figure 2: database design

The meaning of each entity can be defined as follow:

- ***User*** represents a registered user with a name, surname, email and password. The email must be unique.

- ***Travel*** represents a trip created by a user with a token (unique for each travel), title, start date, end date, location (intended as geographical location), latitude, longitude and the user id (indicates the id of the participant). There is an entry for each participant to the same travel.

- ***Note*** represent a note created by a user for a specific travel. It is represented by a text (note text), a taken_flag, a date and the token of the travel it belongs to.

- ***Interesting_site*** represents a point of interest (POI) created by a user for a specific travel. It is represented by a label (title), latitude, longitude, a symbol and the token of the travel it belongs to.

- ***Bill*** represent the bill of a user in a specific travel with a balance, credit, user id and the travel token.

- ***Expense*** represent a single expense of a user in a specific travel with a unique token, a title, date, amount, a refund flag (if the expense is a refund to someone), the payer id, the recipient id and the travel token which it refers.

2.2.2 Database Schema

For the implementation we decide to use the *PostgreSQL* DBMS and in the following pages is provided the database schema for each table.

USER

```
CREATE TABLE public."user"
(
    id integer NOT NULL DEFAULT nextval('user_id_seq'::regclass),
    name character varying(45) COLLATE pg_catalog."default" NOT NULL,
    surname character varying(45) COLLATE pg_catalog."default" NOT NULL,
    email character varying(45) COLLATE pg_catalog."default" NOT NULL,
    password character varying(45) COLLATE pg_catalog."default",
    CONSTRAINT user_pkey PRIMARY KEY (id),
    CONSTRAINT email_unique UNIQUE (email)
)
```

TRAVEL

```
CREATE TABLE public.travel
(
    id integer NOT NULL DEFAULT nextval('travel_id_seq'::regclass),
    token character varying(128) COLLATE pg_catalog."default" NOT NULL,
    title character varying(45) COLLATE pg_catalog."default" NOT NULL,
    start_date date NOT NULL,
    end_date date NOT NULL,
    location character varying(45) COLLATE pg_catalog."default" NOT NULL,
    latitude double precision NOT NULL,
    longitude double precision NOT NULL,
    user_id bigint NOT NULL,
    CONSTRAINT travel_pkey PRIMARY KEY (id),
    CONSTRAINT user_travel_token UNIQUE (token, user_id),
    CONSTRAINT travel_user_id FOREIGN KEY (user_id)
        REFERENCES public."user" (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
```

EXPENSE

```
CREATE TABLE public.expense
(
    id integer NOT NULL DEFAULT nextval('expense_id_seq'::regclass),
    token character varying(128) COLLATE pg_catalog."default" NOT NULL,
    title character varying(45) COLLATE pg_catalog."default" NOT NULL,
    date date NOT NULL,
    amount double precision NOT NULL,
    refund_flag boolean NOT NULL,
    payer_id bigint NOT NULL,
    recipient_id bigint NOT NULL,
    travel_token character varying(128) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT expense_pkey PRIMARY KEY (id),
    CONSTRAINT expense_recipient_id FOREIGN KEY (recipient_id)
        REFERENCES public."user" (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT expense_user_id FOREIGN KEY (payer_id)
        REFERENCES public."user" (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
```

INTERESTING SITE

```
CREATE TABLE public.interesting_site
(
    id integer NOT NULL DEFAULT nextval('interesting_site_id_seq'::regclass),
    label character varying(45) COLLATE pg_catalog."default" NOT NULL,
    latitude double precision NOT NULL,
    longitude double precision NOT NULL,
    symbol character varying(45) COLLATE pg_catalog."default" NOT NULL,
    travel_token character varying(128) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT interesting_site_pkey PRIMARY KEY (id),
    CONSTRAINT location_travel_unique UNIQUE (latitude, longitude, travel_token)
)
```

BILL

```
CREATE TABLE public.bill
(
    id integer NOT NULL DEFAULT nextval('bill_id_seq'::regclass),
    balance double precision NOT NULL DEFAULT 0,
    credit double precision NOT NULL DEFAULT 0,
    user_id bigint NOT NULL,
    travel_token character varying(128) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT bill_pkey PRIMARY KEY (id),
    CONSTRAINT bill_user_id FOREIGN KEY (user_id)
        REFERENCES public."user" (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
```

NOTE

```
CREATE TABLE public.note
(
    id integer NOT NULL DEFAULT nextval('note_id_seq'::regclass),
    text character varying(40) COLLATE pg_catalog."default" NOT NULL DEFAULT false,
    taken_flag boolean NOT NULL DEFAULT false,
    n_date date NOT NULL,
    travel_token character varying(128) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT note_pkey PRIMARY KEY (id),
    CONSTRAINT text_date_travel_token UNIQUE (text, travel_token, n_date)
)
```

2.3 Application Server

2.3.1 Triplan Model API

To simplify the communication among the *Triplan* application server and the *Triplan* mobile application we have implemented a simple API for the model of our system.
The following image shows the UML of the *Triplan Model API*.

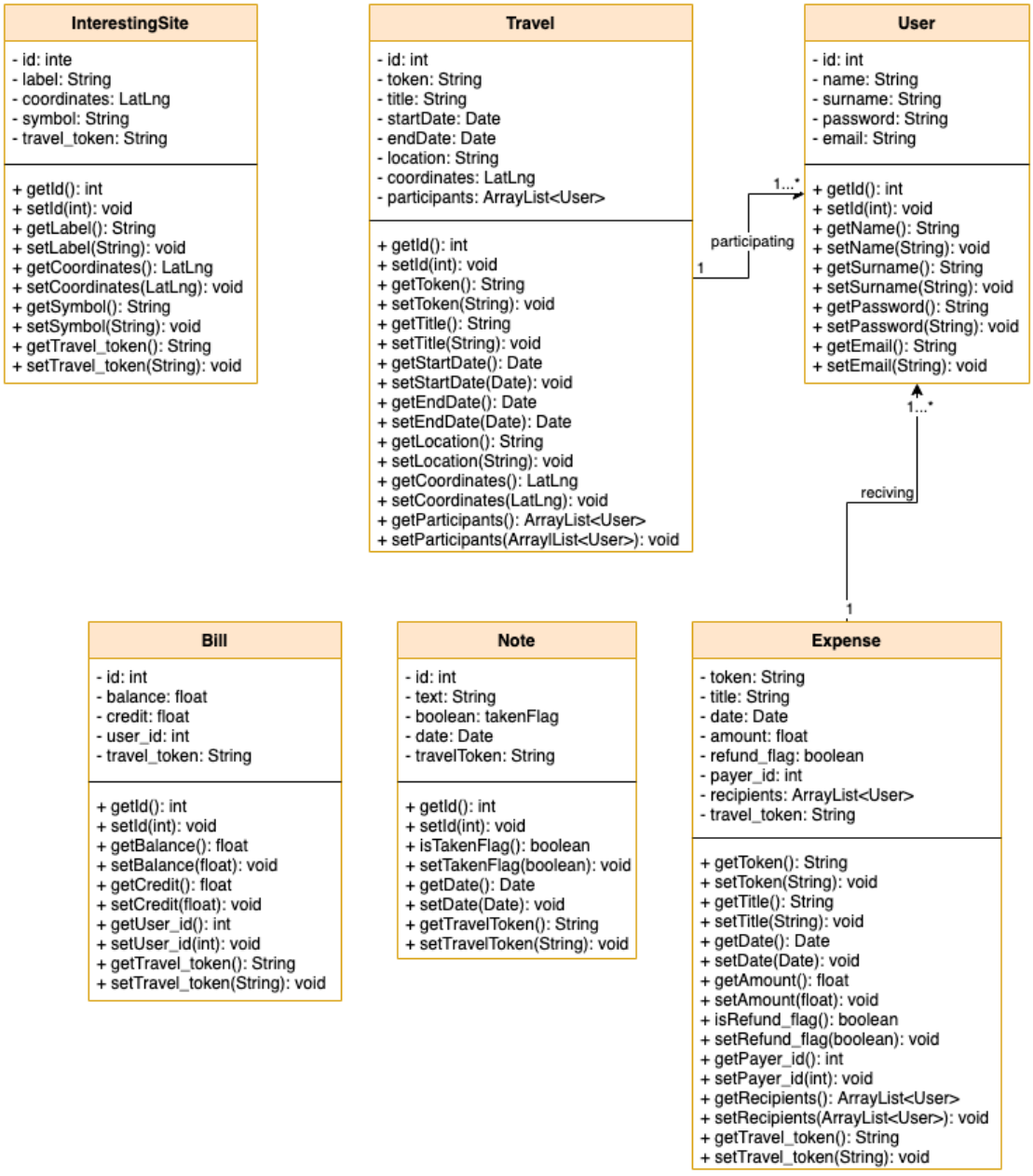


Figure 3: UML Triplan Model API

2.3.2 Servlets

The *Triplan* Application Server is based on Java Servlets. Now we are going to list all the servlets implemented and to describe their behaviors.

- ***/AddExpense*** add a new *Expense* for the requested Travel. Before adding the new value, it checks if the user is authorized to make the request.
- ***/AddNote*** add a new Note for the requested Travel. Before adding the new value, it checks if the user is authorized to make the request
- ***/AddPOI*** add a new InterestingSite for the requested Travel. Before adding the new value, it checks if the user is authorized to make the request.
- ***/AddTravel*** create a new Travel and a new Bill related to the Travel just created for the user who has made the request.
- ***/AddUserExternalService*** register a new user from an external service (*Google* or *Facebook*).
- ***/CheckLogin*** verify the credentials of the user.
- ***/CheckRegistration*** verify the validity of all the parameters and register the new user.
- ***/EditExpense*** edit an existing Expense. Before editing it, check if the user is authorized to make the request.
- ***/EditNote*** edit an existing Note. Before editing it, check if the user is authorized to make the request.
- ***/EditPOI*** edit an existing InterestingSite. Before editing it, check if the user is authorized to make the request.
- ***/EditTravel*** edit an existing Travel. Before editing it, check if the user is authorized to make the request.
- ***/GetAllExpenses*** retrieve all the Expenses of a specific Travel. Before completing the request, check if the user is authorized to make the request.

- ***/GetAllInterestingSites*** retrieve all the InterestingSites of a specific Travel. Before completing the request, check if the user is authorized to make the request.

- ***/GetAllTravelInfo*** retrieve all the informations (Travel, InterestingSites, Bills, Expenses) of a specific Travel. Before completing the request, check if the user is authorized to make the request.

- ***/GetTravelBillsList*** retrieve all the Bills of a specific Travel. Before completing the request, check if the user is authorized to make the request.

- ***/GetTravelsList*** retrieve all the Travels of a specific User. Before completing the request, check if the user is authorized to make the request.

- ***/GetUser*** retrieve a registered User after the login with an external service.

- ***/JoinTravel*** add a new participant to an existing Travel. Before completing the request, check if the user is authorized to make the request.

- ***/RemoveExpense*** remove an existing Expense from a specific Travel. Before completing the request, check if the user is authorized to make the request.

- ***/RemoveNote*** remove an existing Note from a specific Travel. Before completing the request, check if the user is authorized to make the request.

- ***/RemovePOI*** remove an existing InterestingSite from a specific Travel. Before completing the request, check if the user is authorized to make the request.

- ***/RemoveTravel*** remove a participant from an existing Travel. If no more Users participate to the Travel, remove all the information related to that Travel (InterestingSites, Bills, Expenses). Before completing the request, check if the user is authorized to make the request.

2.4 Mobile Application

We have implemented *Triplan* mobile application in **Java** for Android smartphones and tablet, targeting all the devices running an Android version greater or equal than **Android 7.0 - Nougat** (API version: 24).

2.4.1 Model-View-ViewModel (MVVM)

The architectural design pattern used to implement the *Triplan* mobile application is the **MVVM** pattern.

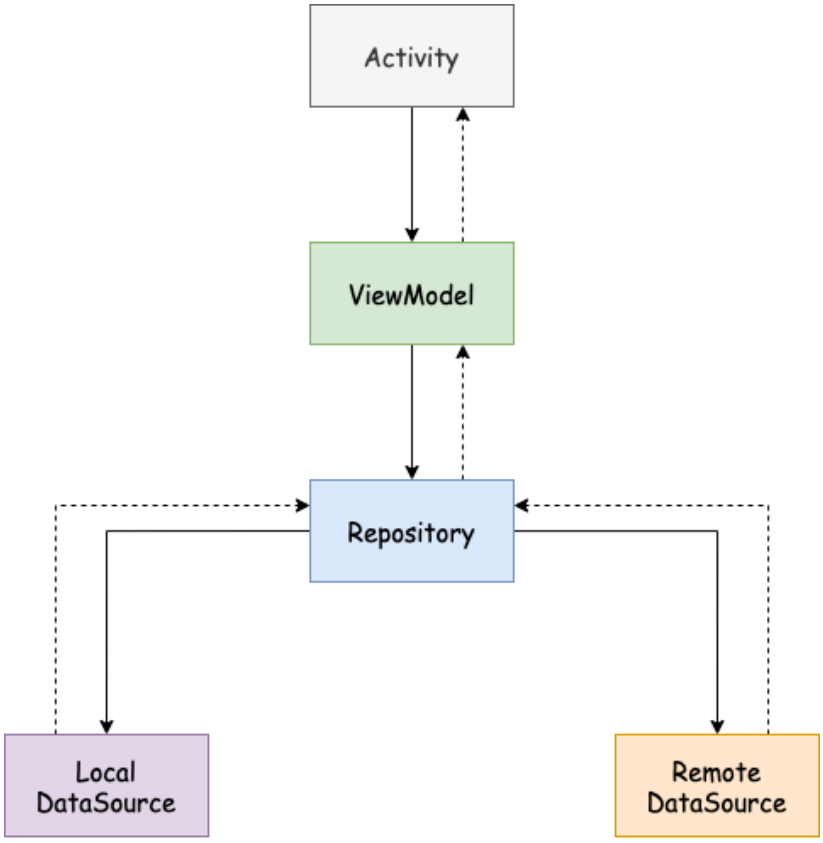


Figure 4: MVVM

The components can be defined as follow:

- ***Activity*** is the Android activity class.
- ***ViewModel*** implements the data and commands connected to the Activity to notify it of state changes via change notification events.

- ***Repository*** in charge to retrieve/modify data from the local database or from a remote server and notify the ViewModel of state changes via change notification events.

The notification events are triggered thanks to the **LiveData** object. The Repository by posting a new value in the LiveData notifies all the components that are observing the LiveData. In this way the UI can be immediately refreshed as soon as a data has changed.

2.4.2 Database Design

Triplan mobile application implements also a local database, implemented with *SQLite* DBMS.

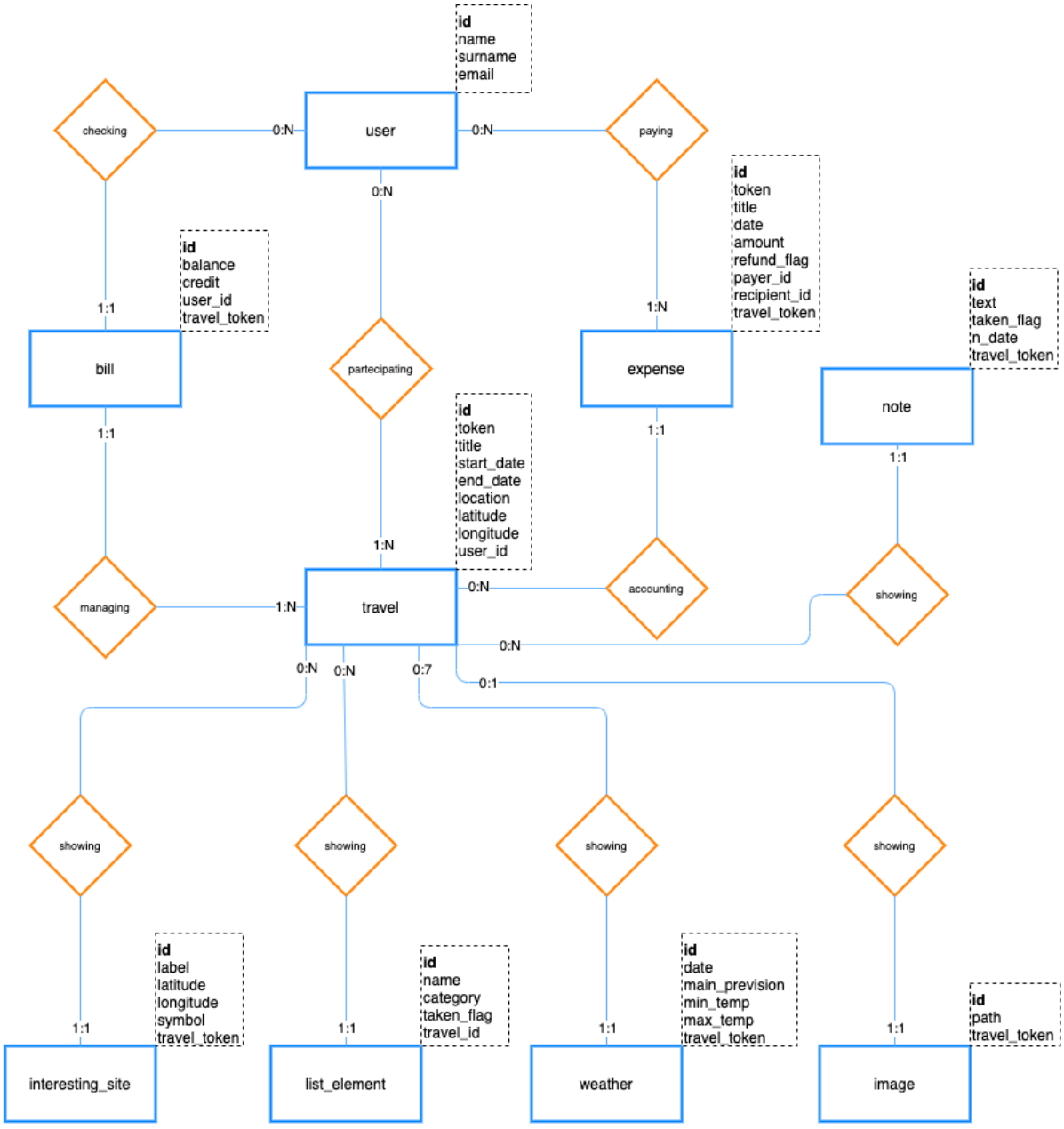


Figure 5: Mobile Application Database Design

The local database replies the online database with all the informations regarding the user with some other additional data:

- ***List_Element*** represents a single element of the 'Trip List' referring to a specific Travel.
- ***Weather*** represents a forecast of a specific day.
- ***Image*** used to store the path of the image chosen by the user of a specific travel.

The utility of the local database is to provide all the informations in case the device cannot access to the internet.

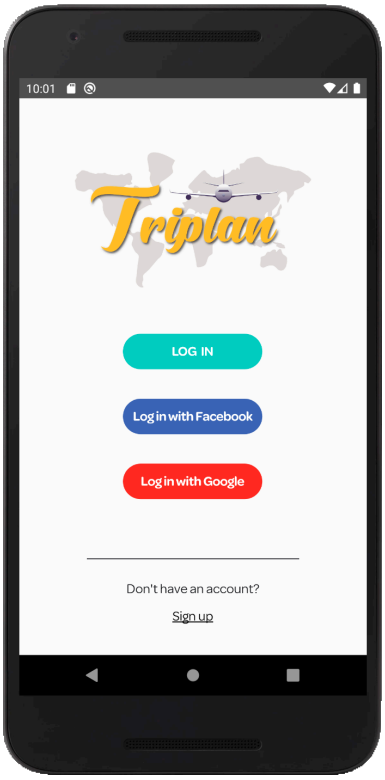
2.5 External API

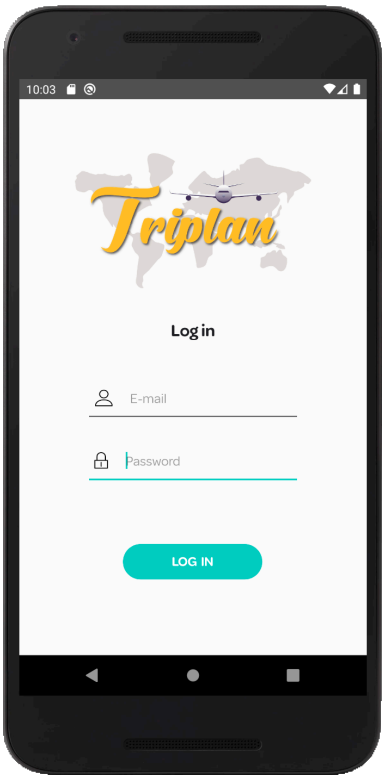
In order to be able to provide all the functionalities, *Triplan* uses different external API:

- ***Facebook Login API*** used for the registration and login with the Facebook credentials.
- ***Google Sign-In API*** used for the registration and login with the Google credentials.
- ***Google Maps API*** used to display and managing the map visualization.
- ***OpenWeather API*** used to retrieve weather informations.

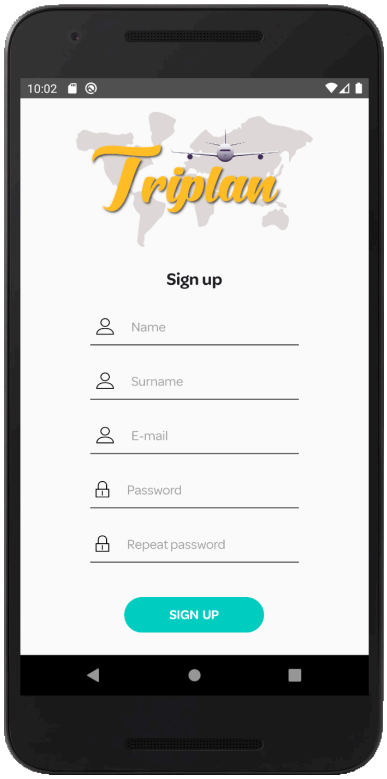
3 User Interface Design

3.1 Login

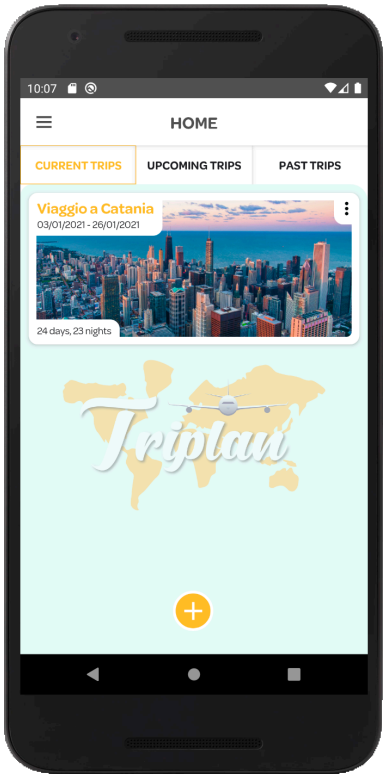


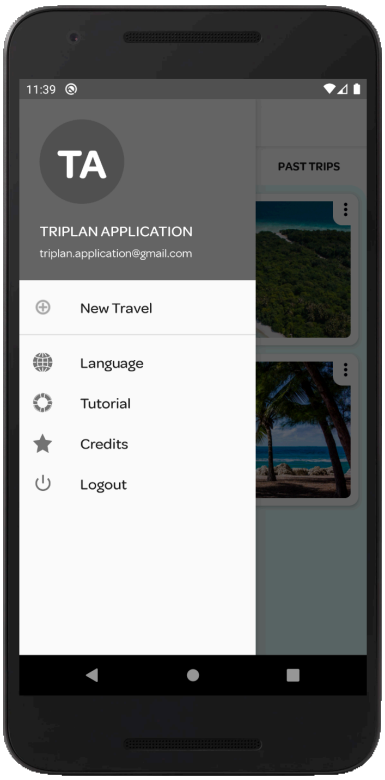


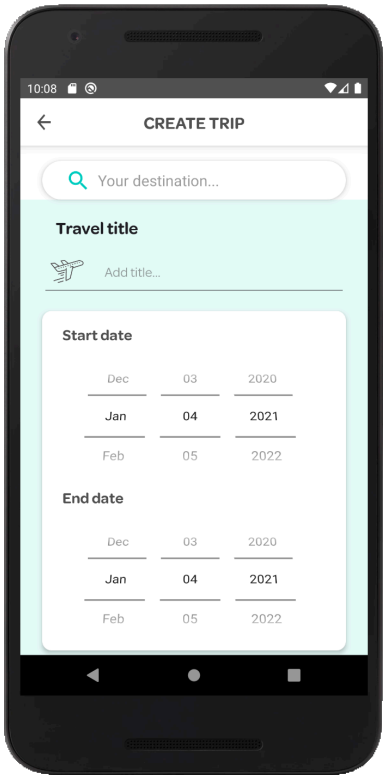
3.2 Sign up

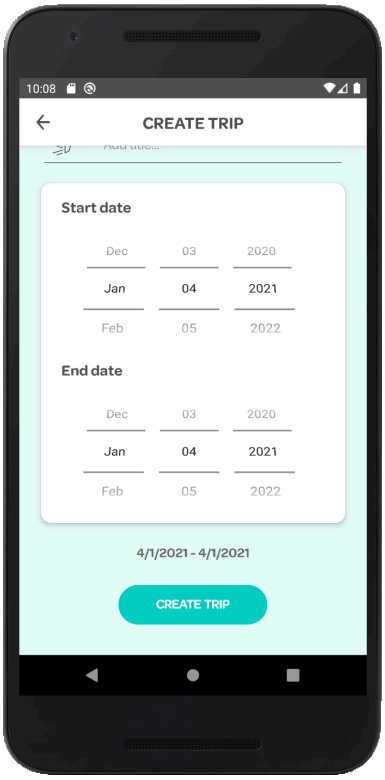


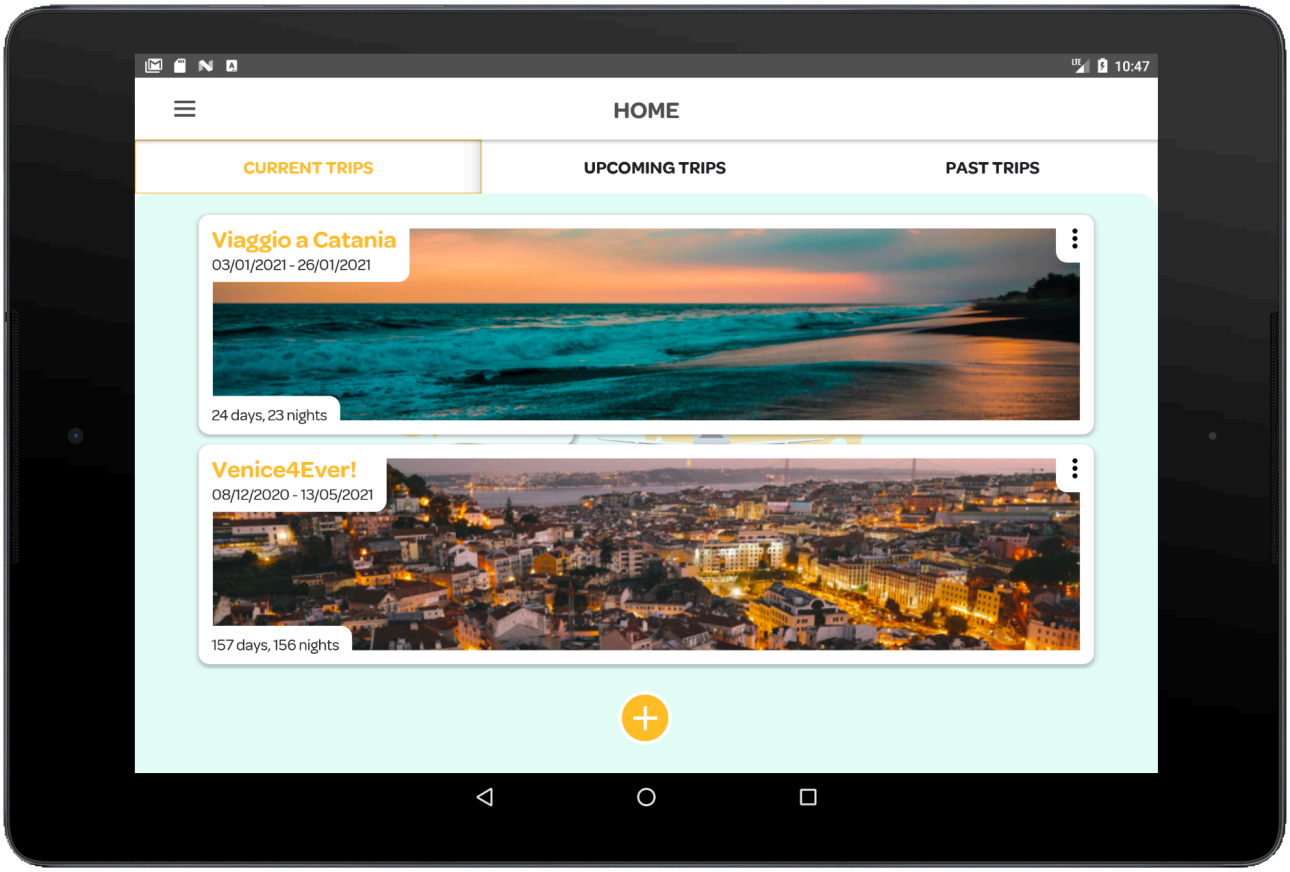
3.3 Home

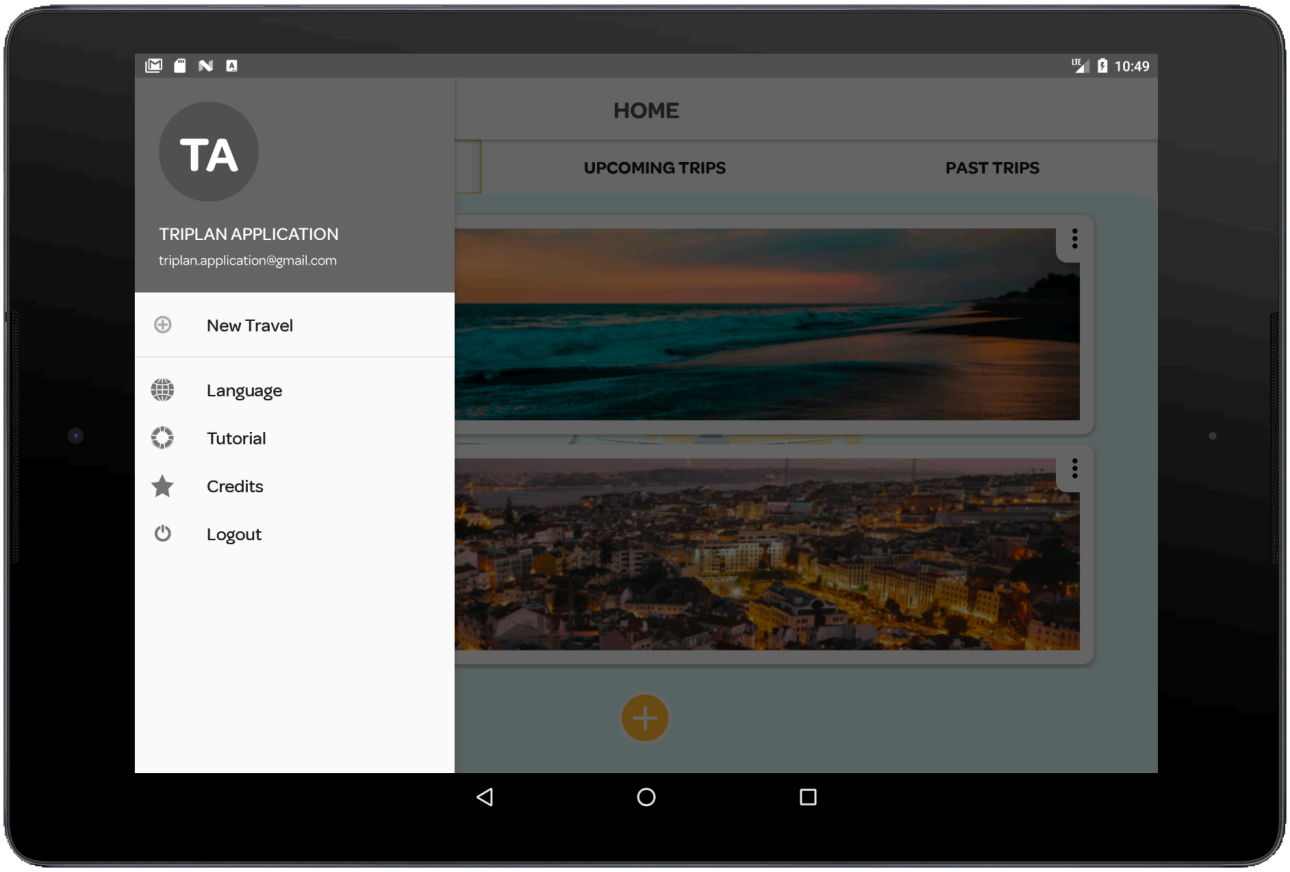

10:07
HOME
CURRENT TRIPS
UPCOMING TRIPS
PAST TRIPS
Viaggio a Catania
03/01/2021 - 26/01/2021
24 days, 23 nights
Triplan


11:39
TA
TRIPLAN APPLICATION
triplan.application@gmail.com
New Travel
Language
Tutorial
Credits
Logout
PAST TRIPS

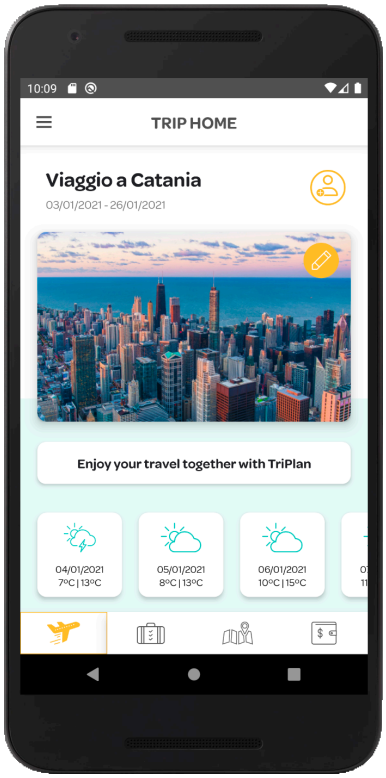

10:08
CREATE TRIP
Your destination...
Travel title
Add title...
Start date
Dec 03 2020
Jan 04 2021
Feb 05 2022
End date
Dec 03 2020
Jan 04 2021
Feb 05 2022

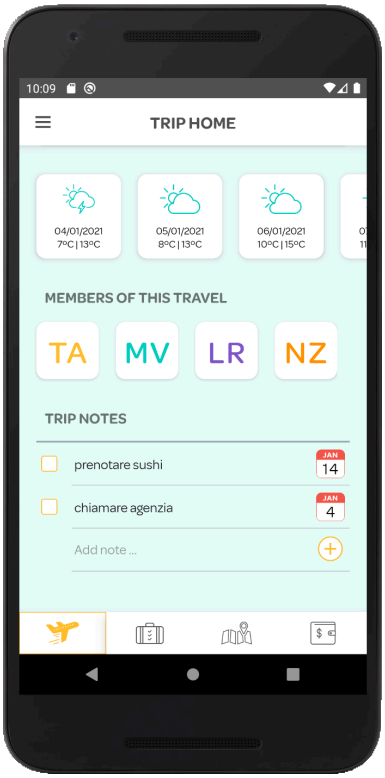

10:08
CREATE TRIP
Start date
Dec 03 2020
Jan 04 2021
Feb 05 2022
End date
Dec 03 2020
Jan 04 2021
Feb 05 2022
4/1/2021 - 4/1/2021
CREATE TRIP

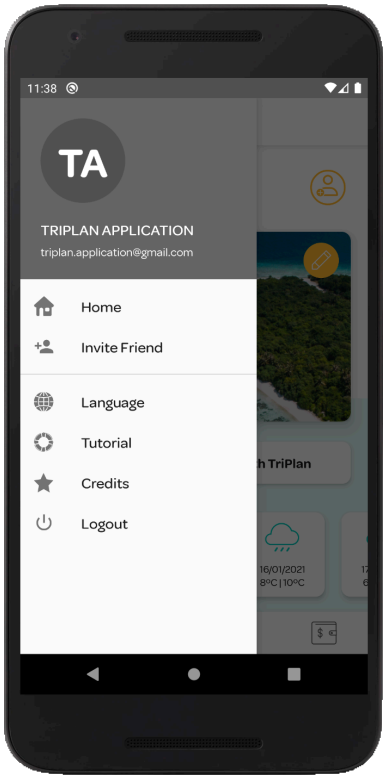

10:47
HOME
CURRENT TRIPS
UPCOMING TRIPS
PAST TRIPS
Viaggio a Catania
03/01/2021 - 26/01/2021
24 days, 23 nights
Venice4Ever!
08/12/2020 - 13/05/2021
157 days, 156 nights

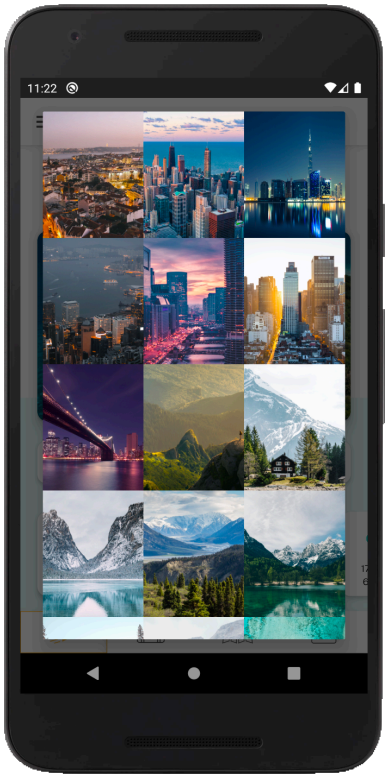

10:49
HOME
UPCOMING TRIPS
PAST TRIPS
TA
TRIPLAN APPLICATION
triplan.application@gmail.com
New Travel
Language
Tutorial
Credits
Logout

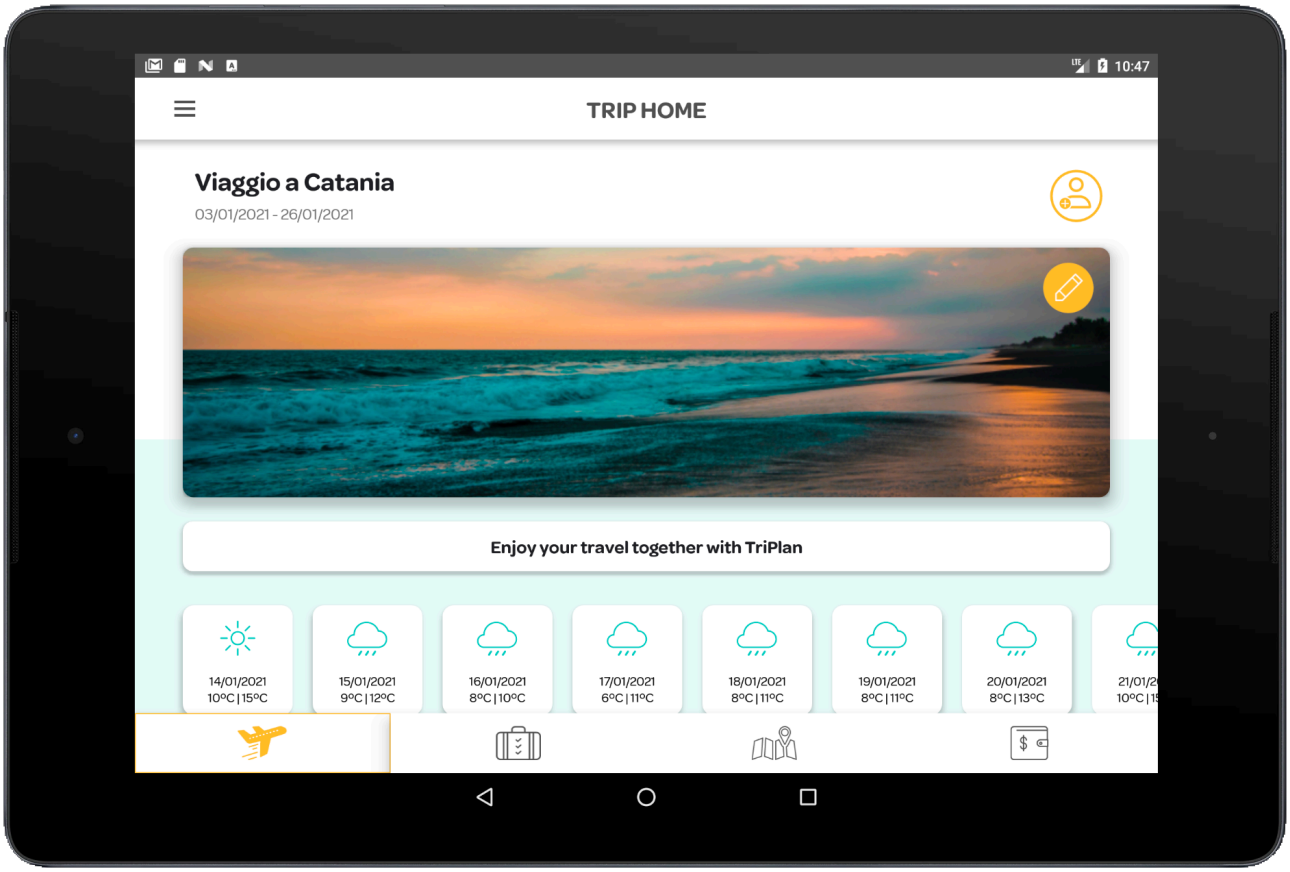
3.4 Trip Home

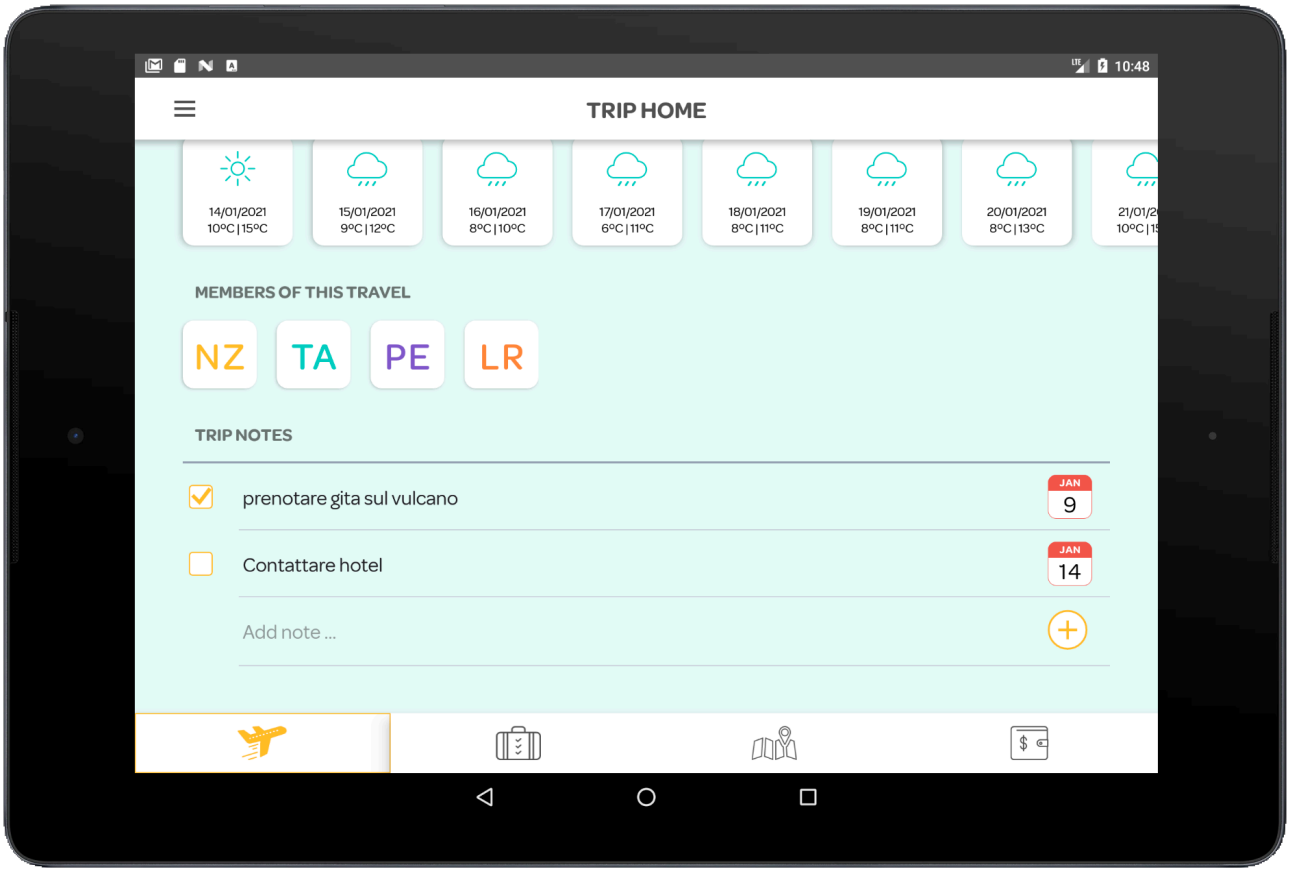

10:09
TRIP HOME
Viaggio a Catania
03/01/2021 - 26/01/2021
Enjoy your travel together with TriPlan
04/01/2021
7°C | 13°C
05/01/2021
8°C | 13°C
06/01/2021
10°C | 15°C


10:09
TRIP HOME
04/01/2021
7°C | 13°C
05/01/2021
8°C | 13°C
06/01/2021
10°C | 15°C
MEMBERS OF THIS TRAVEL
TA
MV
LR
NZ
TRIP NOTES
prenotare sushi
JAN
14
chiamare agenzia
JAN
4
Add note ...

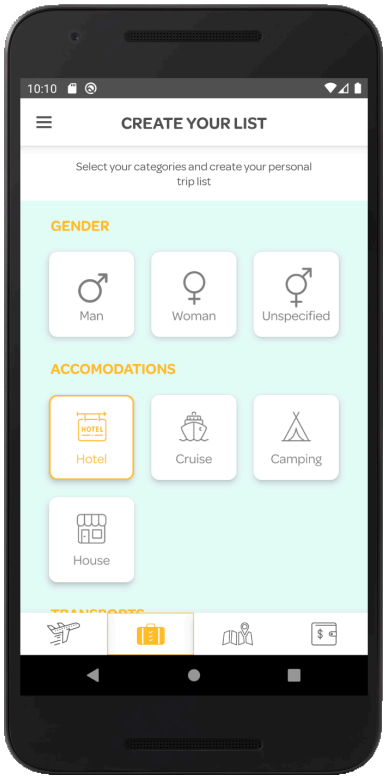

11:38
TA
TRIPLAN APPLICATION
triplan.application@gmail.com
Home
Invite Friend
Language
Tutorial
Credits
Logout

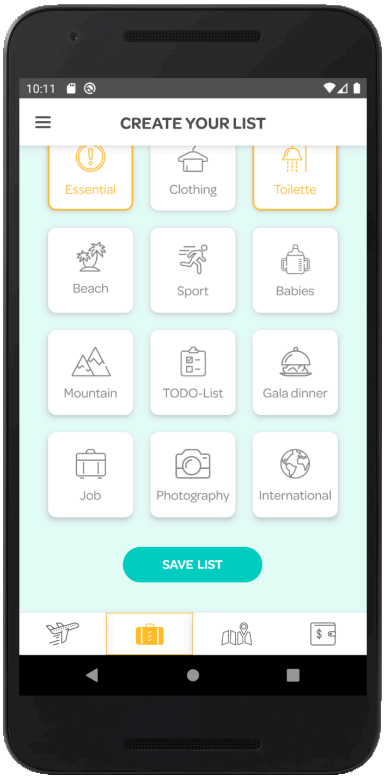

11:22

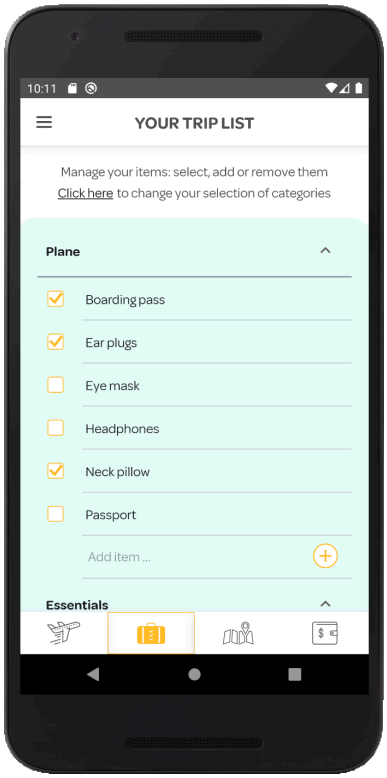

10:47
TRIP HOME
Viaggio a Catania
03/01/2021 - 26/01/2021
Enjoy your travel together with TriPlan
14/01/2021
10°C | 15°C
15/01/2021
9°C | 12°C
16/01/2021
8°C | 10°C
17/01/2021
6°C | 11°C
18/01/2021
8°C | 11°C
19/01/2021
8°C | 11°C
20/01/2021
8°C | 13°C

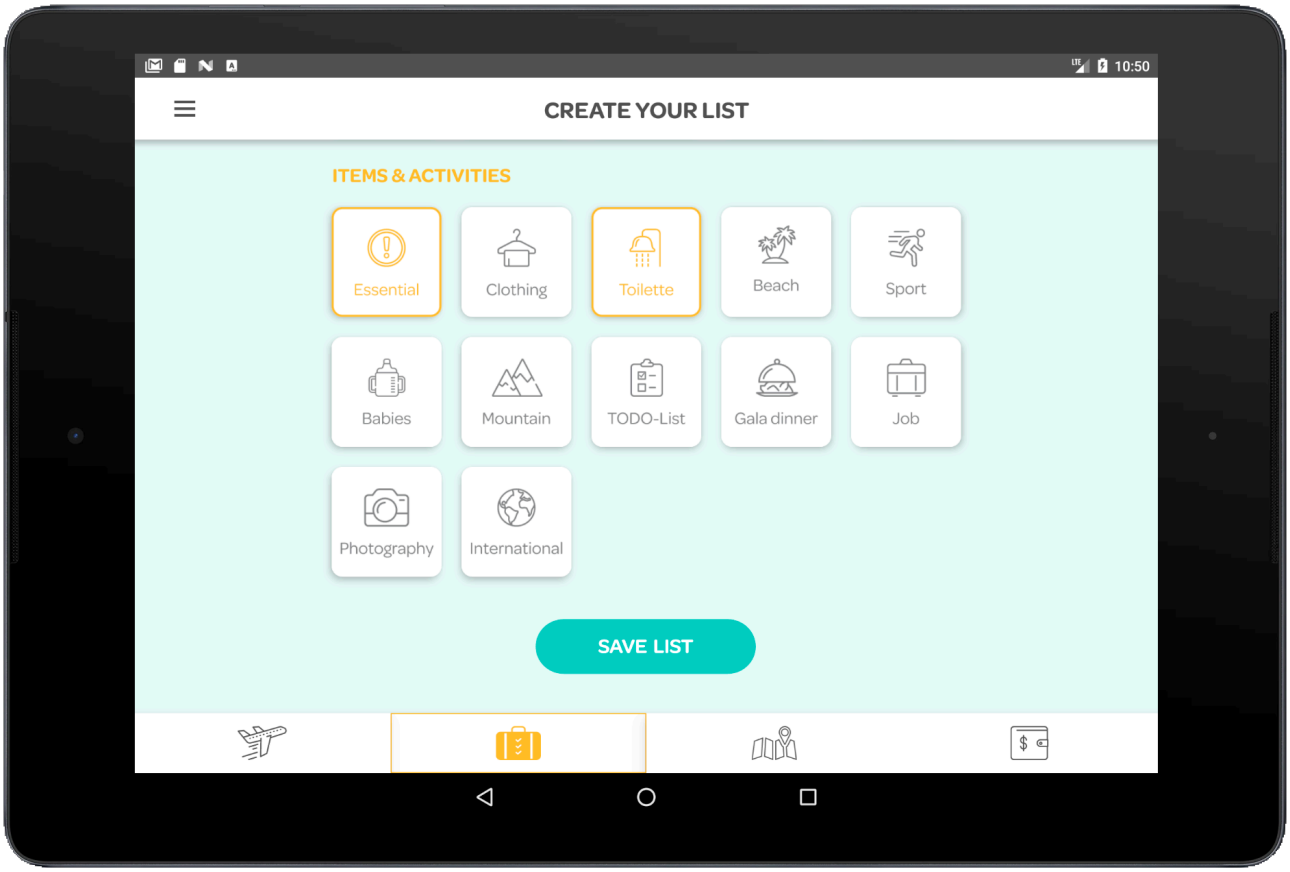

10:48
TRIP HOME
14/01/2021
10°C | 15°C
15/01/2021
9°C | 12°C
16/01/2021
8°C | 10°C
17/01/2021
6°C | 11°C
18/01/2021
8°C | 11°C
19/01/2021
8°C | 11°C
20/01/2021
8°C | 13°C
MEMBERS OF THIS TRAVEL
NZ
TA
PE
LR
TRIP NOTES
prenotare gita sul vulcano
JAN
9
Contattare hotel
JAN
14
Add note ...

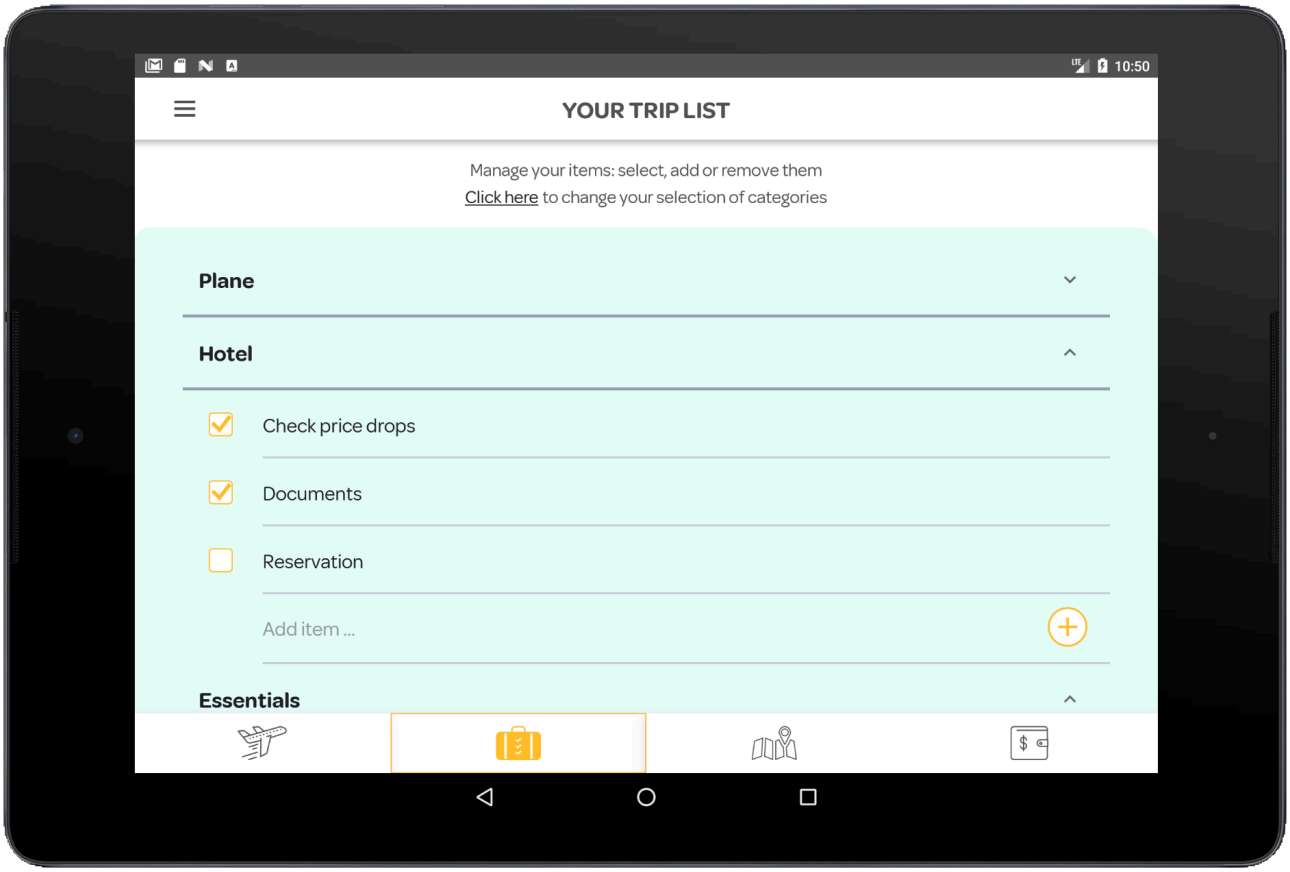
3.5 Personal list



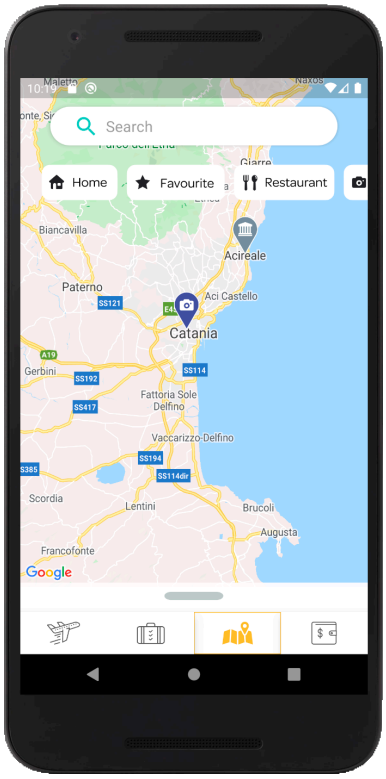


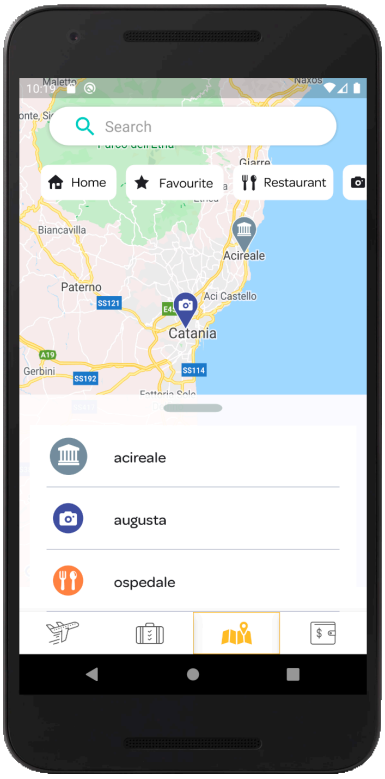


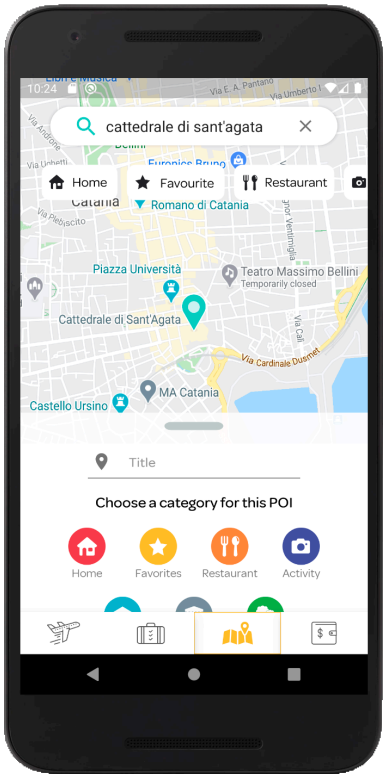

10:50
CREATE YOUR LIST
ITEMS & ACTIVITIES
Essential
Clothing
Toilette
Beach
Sport
Babies
Mountain
TODO-List
Gala dinner
Job
Photography
International
SAVE LIST

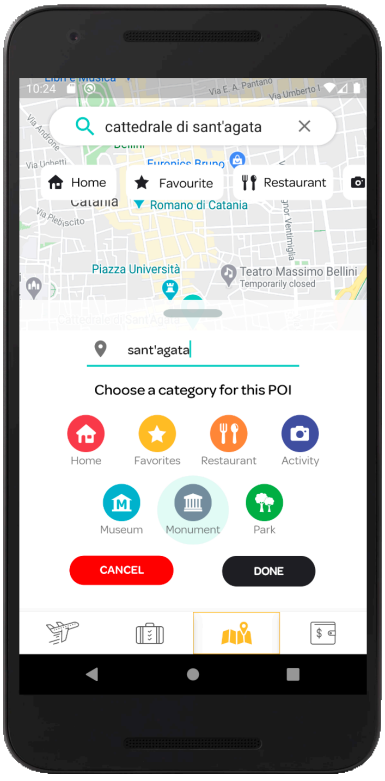

10:50
YOUR TRIP LIST
Manage your items: select, add or remove them
Click here to change your selection of categories
Plane
Hotel
Check price drops
Documents
Reservation
Add item ...
Essentials

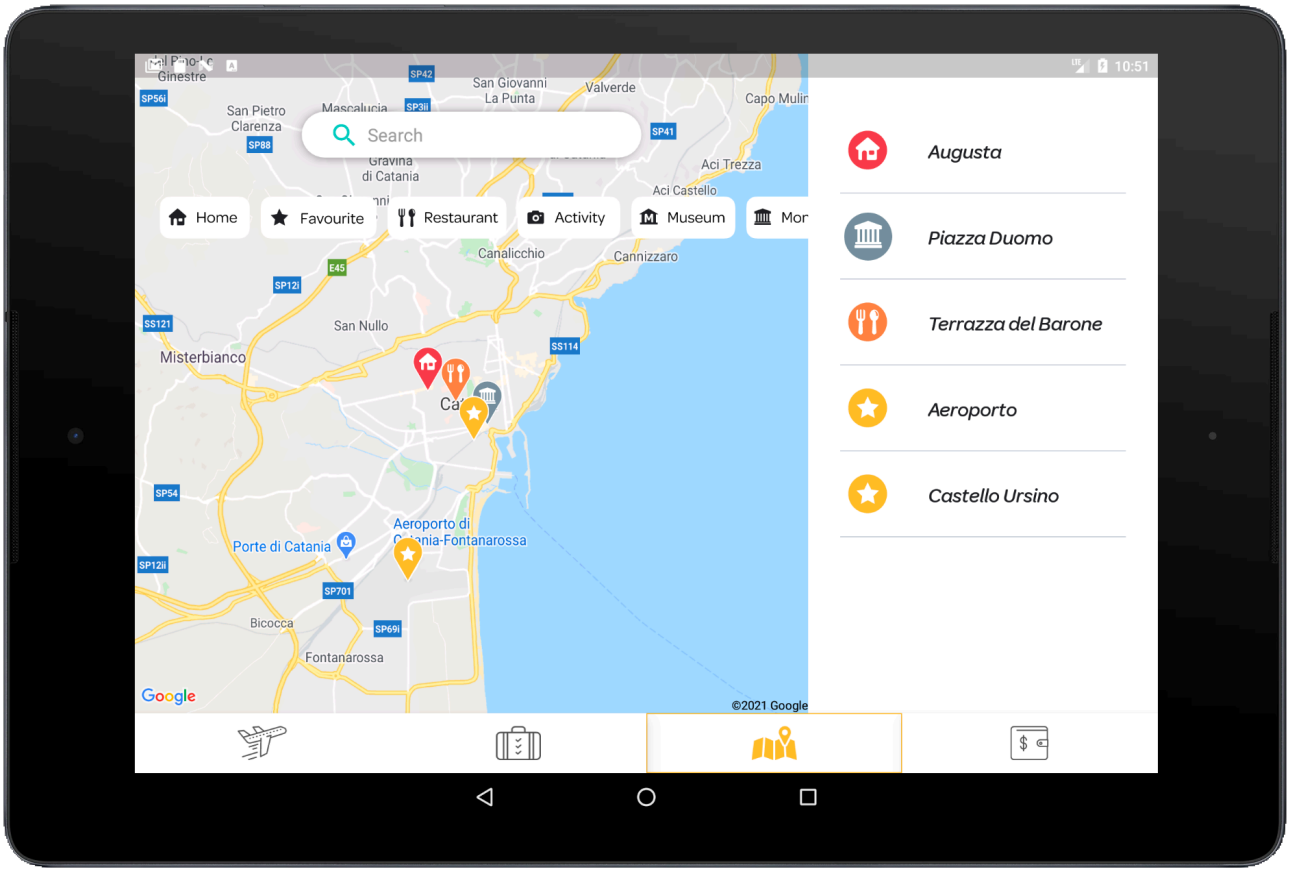
3.6 Map

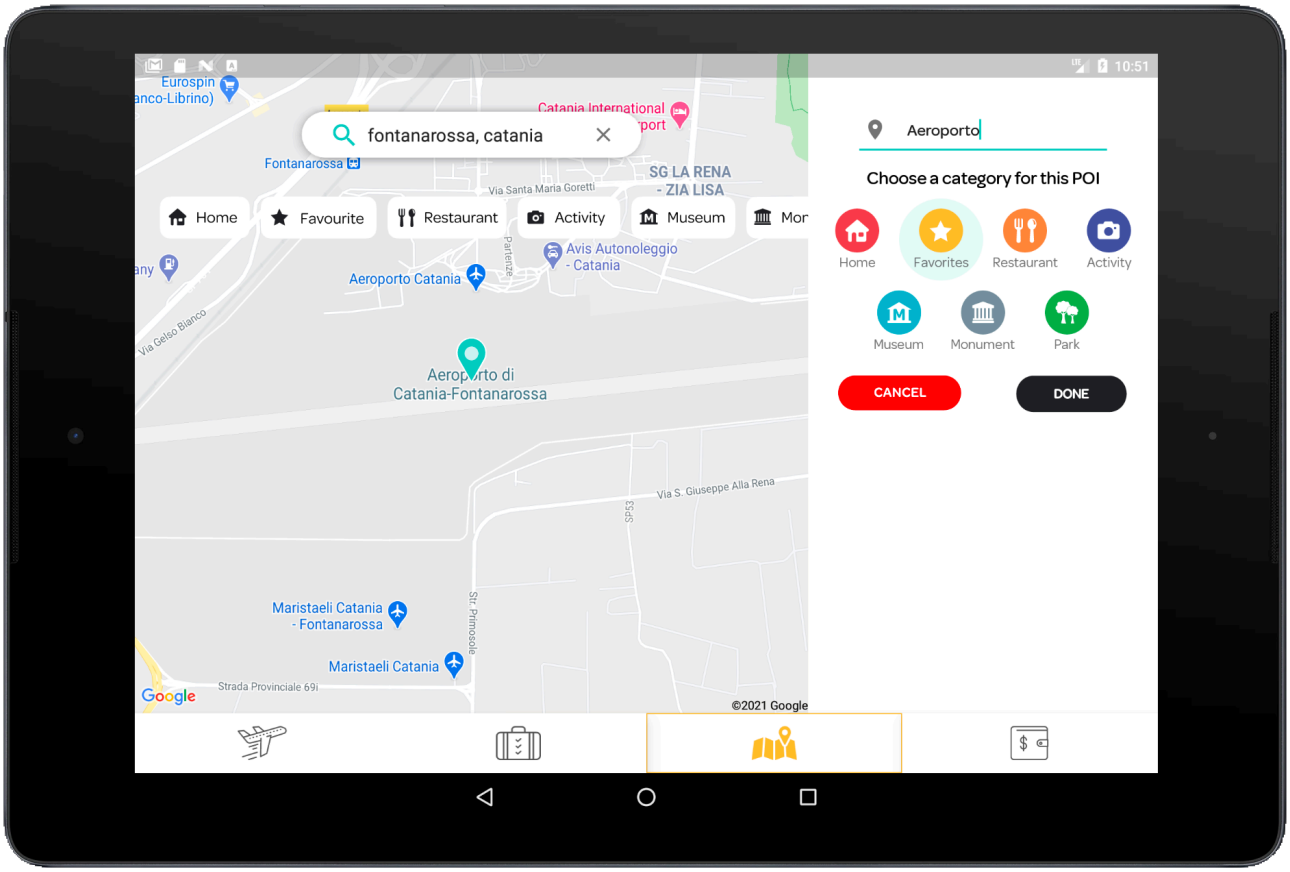

Search
Home
Favourite
Restaurant
Biancavilla
Acireale
Paterno
Aci Castello
Catania
Gerbini
Fattoria Sole
Delfino
Vaccarizzo Delfino
Scordia
Lentini
Brucoli
Augusta
Francofonte
Google


Search
Home
Favourite
Restaurant
Biancavilla
Acireale
Paterno
Aci Castello
Catania
Gerbini
acireale
augusta
ospedale

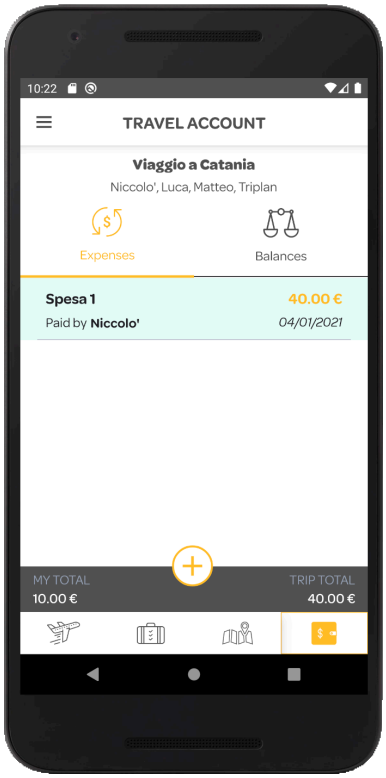

cattedrale di sant'agata
Home
Favourite
Restaurant
Romano di Catania
Piazza Università
Teatro Massimo Bellini
Temporarily closed
Cattedrale di Sant'Agata
Via Cardinale Dusmet
MA Catania
Castello Ursino
Title
Choose a category for this POI
Home
Favorites
Restaurant
Activity

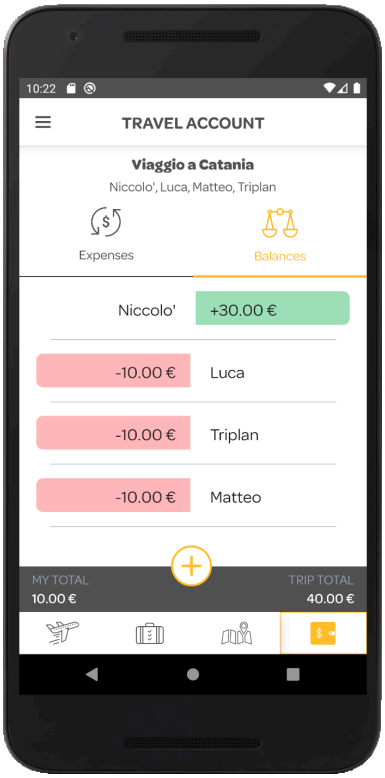

cattedrale di sant'agata
Home
Favourite
Restaurant
Romano di Catania
Piazza Università
Teatro Massimo Bellini
Temporarily closed
sant'agata
Choose a category for this POI
Home
Favorites
Restaurant
Activity
Museum
Monument
Park
CANCEL
DONE

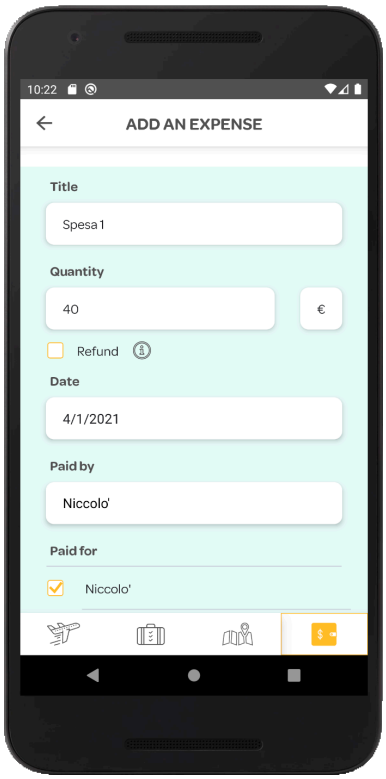

10:51
Search
Home
Favourite
Restaurant
Activity
Museum
Augusta
Piazza Duomo
Terrazza del Barone
Aeroporto
Castello Ursino
Misterbianco
San Nullo
Canalicchio
Cannizzaro
Aci Trezza
Aci Castello
Valverde
San Giovanni La Punta
San Pietro Clarenza
Gravina di Catania
Porte di Catania
Aeroporto di Catania-Fontanarossa
Bicocca
Fontanarossa
Google
©2021 Google

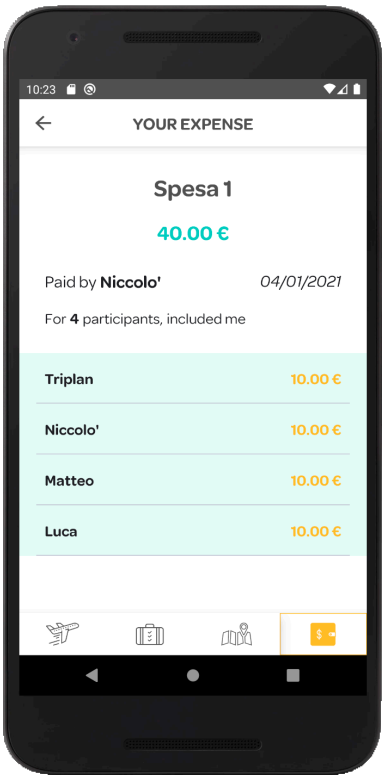

10:51
fontanarossa, catania
Aeroporto
Choose a category for this POI
Home
Favorites
Restaurant
Activity
Museum
Monument
Park
CANCEL
DONE
Home
Favourite
Restaurant
Activity
Museum
Fontanarossa
Aeroporto Catania
Avis Autonoleggio - Catania
Aeroporto di Catania-Fontanarossa
SG LA RENA - ZIA LISA
Via Santa Maria Goretti
Via Geleo Bianco
Via S. Giuseppe Alla Rena
Maristaeli Catania - Fontanarossa
Maristaeli Catania
Strada Provinciale 69i
Google
©2021 Google

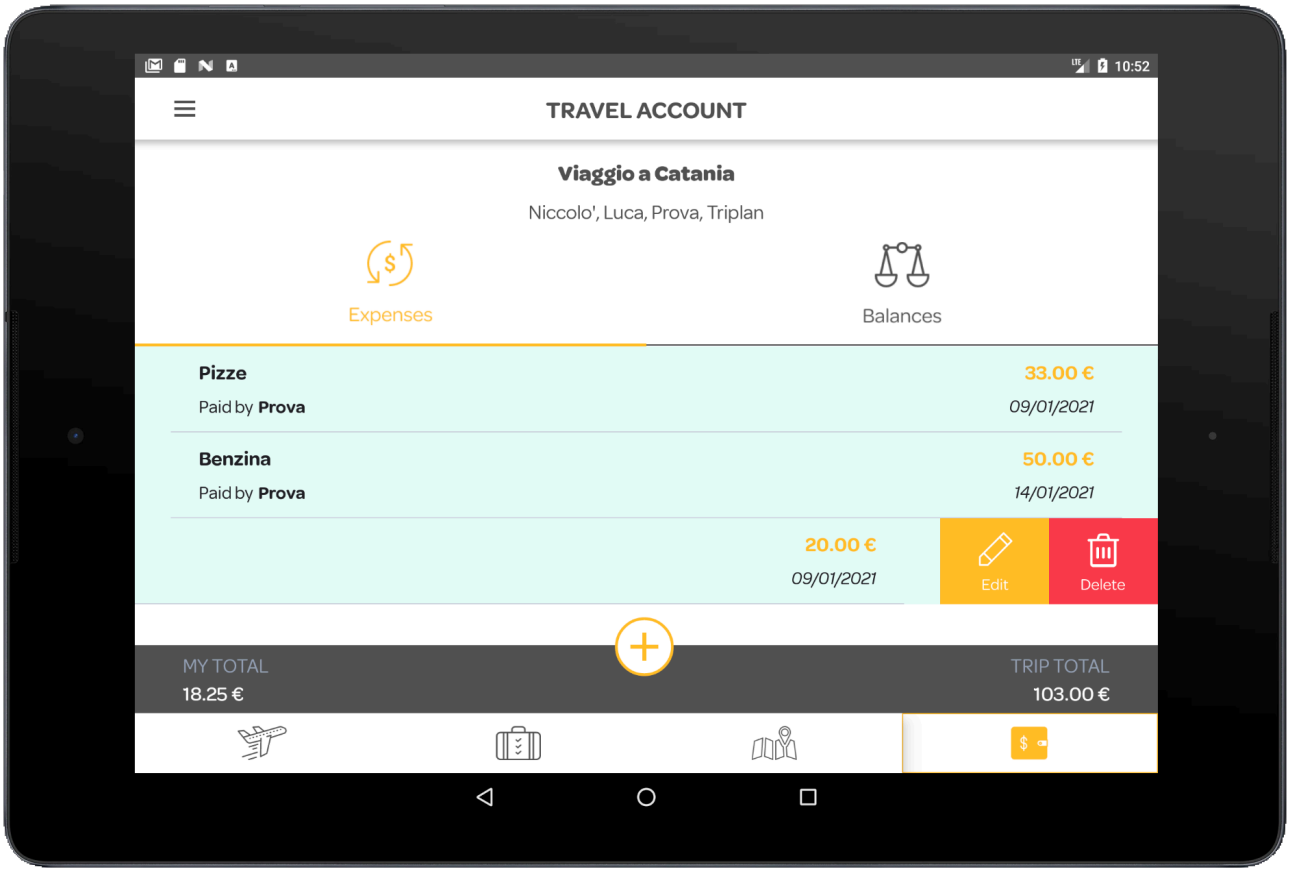
3.7 Account

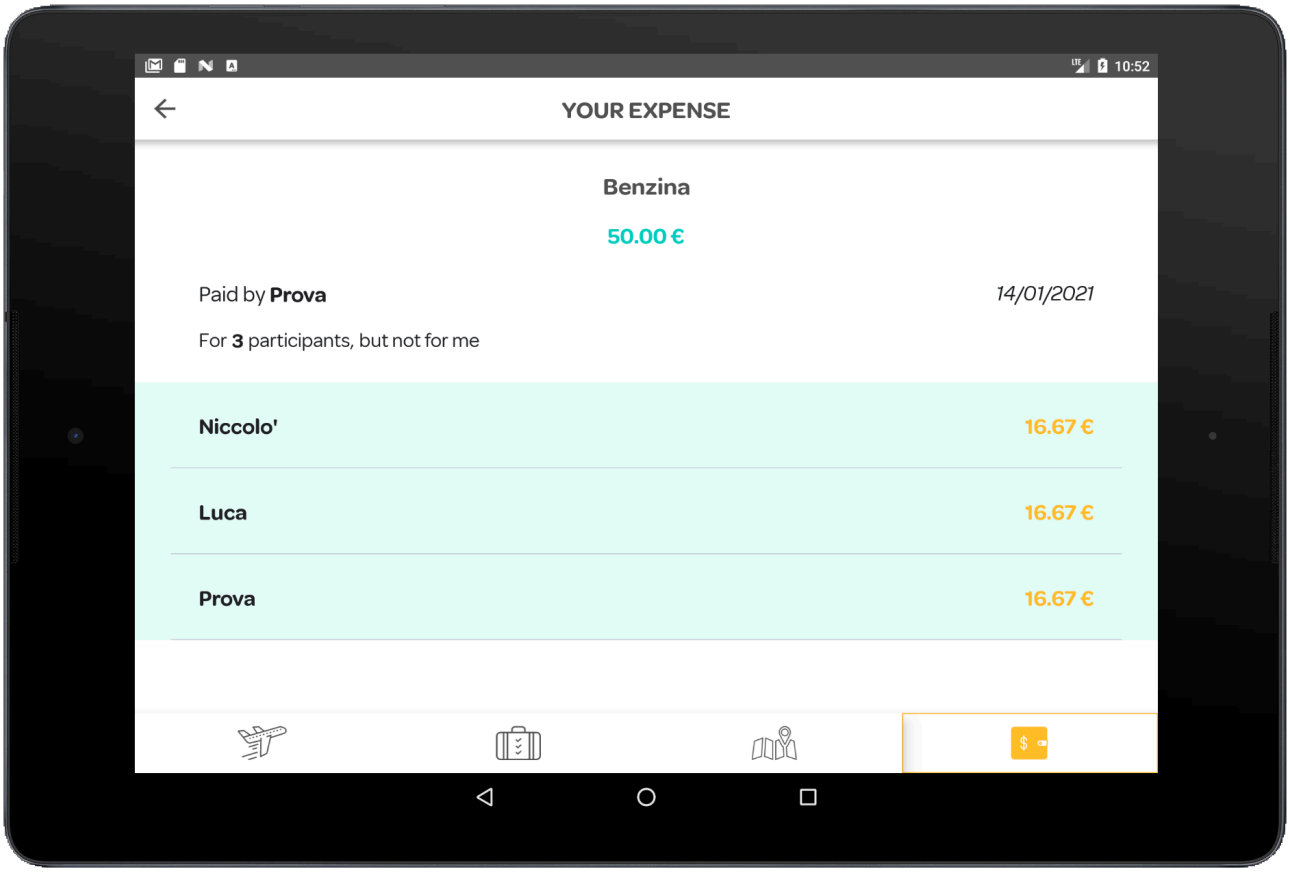

10:22
TRAVEL ACCOUNT
Viaggio a Catania
Niccolo', Luca, Matteo, Triplan
Expenses
Balances
Spesa 1
40.00 €
Paid by Niccolo'
04/01/2021
MY TOTAL
10.00 €
TRIP TOTAL
40.00 €


10:22
TRAVEL ACCOUNT
Viaggio a Catania
Niccolo', Luca, Matteo, Triplan
Expenses
Balances
Niccolo'
+30.00 €
-10.00 €
Luca
-10.00 €
Triplan
-10.00 €
Matteo
MY TOTAL
10.00 €
TRIP TOTAL
40.00 €


10:22
ADD AN EXPENSE
Title
Spesa 1
Quantity
40
€
Refund
Date
4/1/2021
Paid by
Niccolo'
Paid for
Niccolo'


10:23
YOUR EXPENSE
Spesa 1
40.00 €
Paid by Niccolo'
04/01/2021
For 4 participants, included me
Triplan
10.00 €
Niccolo'
10.00 €
Matteo
10.00 €
Luca
10.00 €


10:52
TRAVEL ACCOUNT
Viaggio a Catania
Niccolo', Luca, Prova, Triplan
Expenses
Balances
Pizze
Paid by Prova
33.00 €
09/01/2021
Benzina
Paid by Prova
50.00 €
14/01/2021
20.00 €
09/01/2021
Edit
Delete
MY TOTAL
18.25 €
TRIP TOTAL
103.00 €


10:52
YOUR EXPENSE
Benzina
50.00 €
Paid by Prova
14/01/2021
For 3 participants, but not for me
Niccolo'
16.67 €
Luca
16.67 €
Prova
16.67 €