REAL ESTATE MANAGER APP

Diego Fajardo
OpenClassrooms Project
October 2018

INDEX

1. INTRODUCTION

“Real Estate Manager App” is an application that allows real estate agents to keep track of the listings they are currently in charge of. The app allows to create the listings, modify and display them in an intuitive and elegant way. It also provides a “loan simulator” and a search engine for searching for listings. according to all the specific features of each listing. The app can be used in both, handsets and tablets.

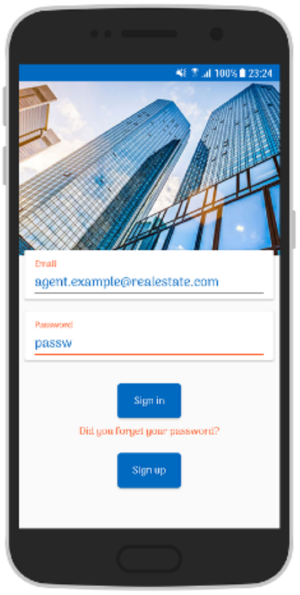
As additional features, the app manages photos efficiently and allow the agent to keep its information secure providing an email/password service.

2. USER INTERFACE

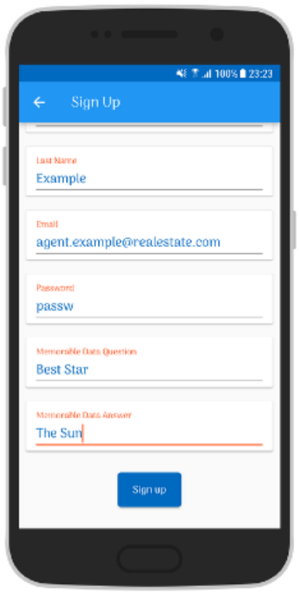
The user interface is divided into several screens:

- **Splash Screen**: the user is welcomed to use the app via a simple but elegant splash screen.

- **Authentication Screen:** in this screen, the user will be able to input its email and password. If the user is not already registered, he/she can do it using the “Sign Up” button. The user can also retrieve his/her password clicking the “Did you forget your password?” button.



- Sign Up Screen: the user can use this screen to sign up. It allows the user to input 6 fields (See "3.3.1. Local Database", Agent Table). Once the user clicks the signup button, the app will check that a user with the same email does not exist. If the email is free, the app will create the user; otherwise, it will notify the user that that email is currently being used. The accounts cannot be deleted.



Last Name
Example

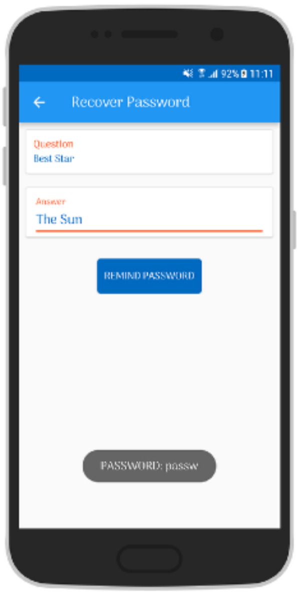
Email
agent.example@realestate.com

Password
passw

Memorable Data Question
Best Star

Memorable Data Answer
The Sun

- Forgot Password Screen: this screen will display the "memorable data question" to the user to give a hint about the "memorable data answer". The app will notify the user whether the answer is correct or not. If it's correct, it will show the password.



Question
Best Star

Answer
The Sun

PASSWORD: passw

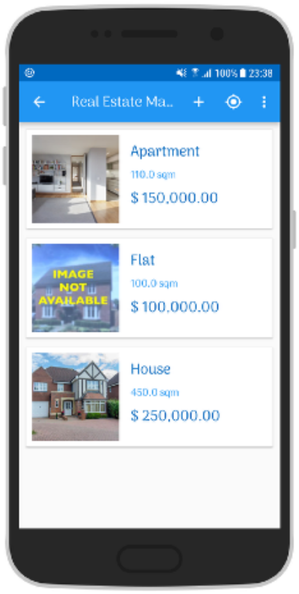
- Main Screen: the “Main Screen” shows different information depending on if the user is using a Handset or a Tablet. Nevertheless, if the database is empty (no real estates have been introduced yet), the “Main Screen” will display some text showing the user how to Create a new listing (clicking the “+” button on the Toolbar). When at least one listing is in the database, the text will disappear and some information will be displayed (depends on the type of device).

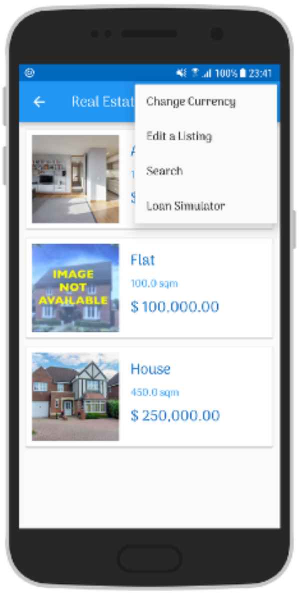
1. For handsets, the “Main Screen” will display the complete list of all the real estates of the database (additionally, information related to the listing will be displayed using a RecyclerView: the type of building, the price, the surface area and, if the real estate has been sold, a label notifying it). If the user clicks an item of the list, a “Detail Screen’ will be launched shown all the available information and a map including all nearby points of interest).
2. For tablets, the RecyclerView and the “Detail Screen” will be merged in one (the “Main Screen”). The screen will show the items and when one is clicked, information related to it.

The “Main Screen” allows entering the “Edit Mode” (see more information below). When the “Edit Mode” is active, the toolbar will notify it and if an item of the RecyclerView is clicked, the “Edit Screen” will be launched which allows the user to modify a real estate that is already in the database.

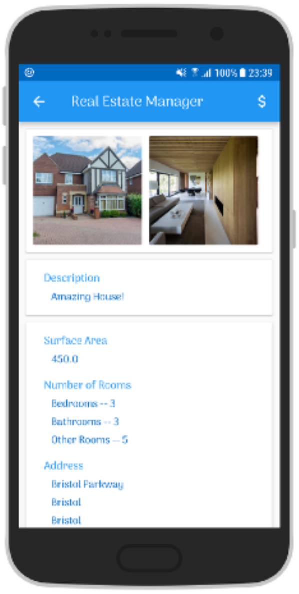
The toolbar of this screen will display several buttons:

1. “Create New Listing”: brings the user to Create New Listing Screen (see below) which allows the user to fill the database with a new real estate.
2. “Position’: brings the user to Position Screen which displays the current location of the user and all the listings in a map.
3. “Change Currency”: allows the user to change between dollars and euros. This button can be found also in all the screens that display prices.
4. “Edit Listing’: allows switching between “Edit Mode” and “Normal Mode”. If “Edit Mode” is active, when an item of the RecyclerView is clicked, the “Edit Screen” will be launched.
5. ‘Search”: launches the “Search Screen “which allows searching for real estates in the database according to different criteria.
6. “Loan Simulator’: launches the “Loan Simulator Screen” which allows the user to simulate a loan using different values.





- Detail Screen: this screen displays the information about the real estate when the user clicks an item in "Main Screen". It will show the related images, the description, general information (surface area, price, number of bedrooms, address, sale estate and agent) and a map showing the position of the real estate and the nearby points of interest. If the user clicks an image, the description of the photo will be shown on a Toast.

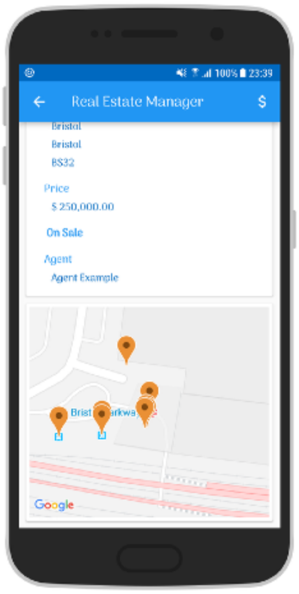


Description
Amazing House!

Surface Area
450.0

Number of Rooms
Bedrooms -- 3
Bathrooms -- 3
Other Rooms -- 5

Address
Bristol Parkway
Bristol
Bristol



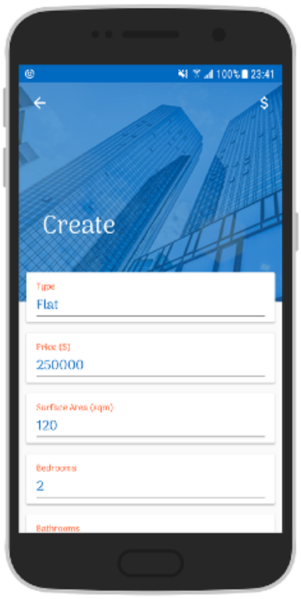
Bristol
Bristol
BS32

Price
$ 250,000.00

On Sale

Agent
Agent Example

- Create New Listing Screen: this screen allows the user to create a new real estate and introduce it in the database. The only information that is necessary to introduce is the "address" of the real estate which will be used to be displayed using a map. This process requires fetching information using several "Google Services" (see "3.2.2. Fetching Process"). This screen also displays photos of the real estate (using a RecyclerView) but they have to be introduced using the "Add Photos Screen". Once a photo has been added, it cannot be deleted. See "3.2.1. Real Estate Stored Information" to find a list of the information the user can introduce about the real estate (the "date sale" will be automatically put as today, the "date sale" will be kept empty and the "found" value will be set to false; the "date sale" field can be changed in "Edit Screen").



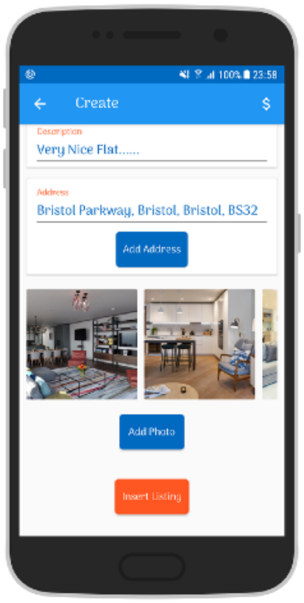
Type
Flat

Price ($)
250000

Surface Area (sqm)
120

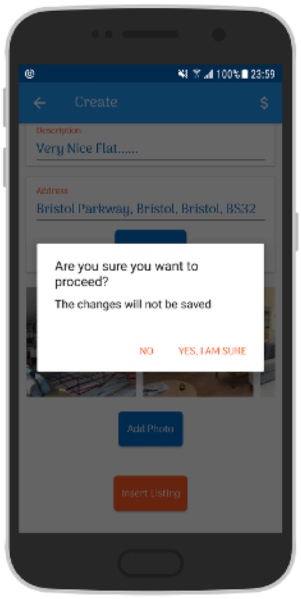
Bedrooms
2

Bathrooms



Description
Very Nice Flat......

Address
Bristol Parkway, Bristol, Bristol, BS32



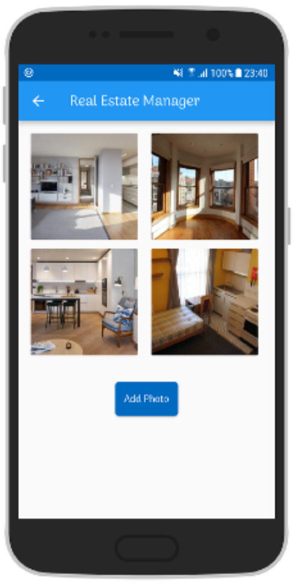
Description
Very Nice Flat......

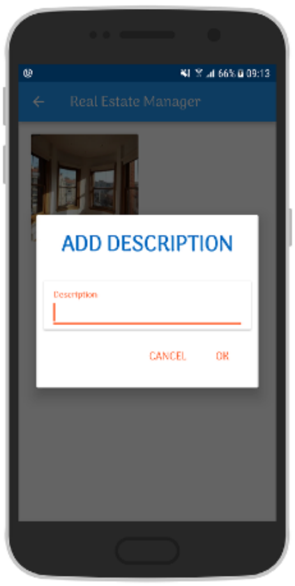
Address
Bristol Parkway, Bristol, Bristol, BS32

Are you sure you want to proceed?

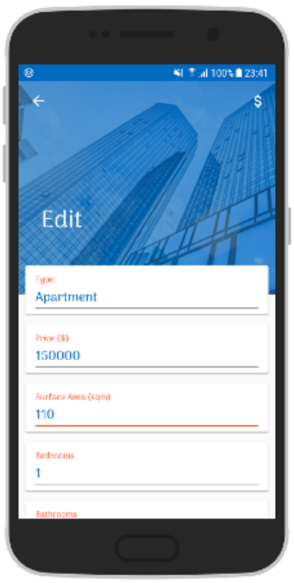
The changes will not be saved

- **Add Photos Screen:** this screen allows the user to input photos of the real estate using the "Gallery App" of the phone. The app will display the photos in a grid. If the user clicks a picture, a dialog will appear allowing the user to enter a description for the real estate.

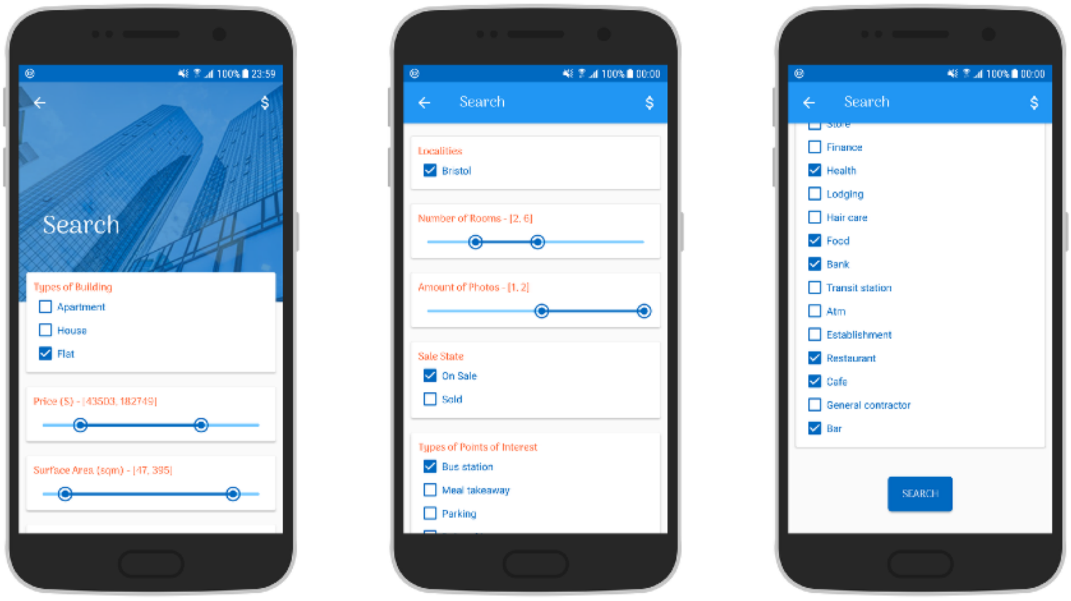




- **Edit Screen:** the "Edit Screen" works in a similar way as the "Create New Listing Screen". The only difference is that the user can modify in this screen the "Sale State" of the real estate (basically, if it has been already sold or not). For that, the user can click a checkbox and use a dialog fragment in order to introduce the date which cannot be after today's date). In order for the changes to be saved, the "Edit Button" has to be clicked (otherwise, they will not be saved). The "address" cannot be changed and no photos can be deleted.

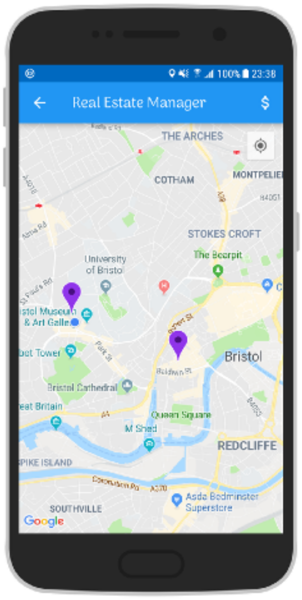


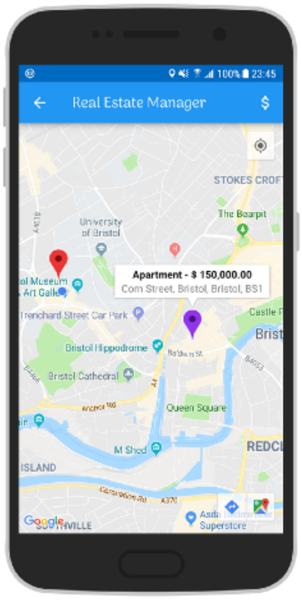
- Search Engine Screen: this screen allows the user to find real estates according to a specific criteria. Some information will be displayed in checkboxes (depending on the information in the database) and other will be displayed using "Seek Bars" so the user can introduce 2 values, maximum and minimum. See "3.4. Search Engine" to know how the layout is distributed. The user can also search for real estates according to the "Sale State" (sold or not sold yet).



- Search Results Screen: this screen is similar to the "Main Screen" but it has limited functionality (eg. the "Edit Mode" cannot be activated and several screens cannot be launched from this screen). It will display the found articles after they are found by the "Search Engine".

- Position Screen: this screen displays information about the current position of the user and all the real estates on the database. If a marker of the map is clicked, the price (the user can modify the currency using the button on the toolbar) and the address will be displayed. If the label is clicked, the "Detail Screen" will be launched. Those real estates that are currently sold will appear in red (pin colour) while those that are not will appear in purple (pin colour).





- Loan Simulator Screen: this screen allows the agent to simulate a loan according to the information inputted. The toolbar displays 3 buttons that can be used for different features:

1. Show Loan button: makes the loan table occupy all the screen or reduce its size (depends on the current state of the own table).

2. Edit Loan: launches a dialog fragment to input new information that will modify the loan. The information that can be added is the following:

a) Loan Amount in dollars or euros (the user can use the currency button to change between both currencies).

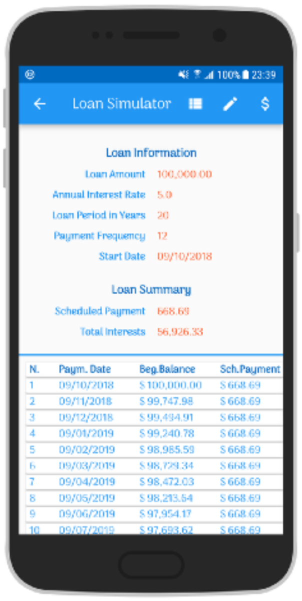
b) Annual Interest Rate.

c) Loan Period in Years.

d) Payment Frequency.

3. Currency button.

The loan will be calculated using today as Start Date (this functionality cannot be changed). When the user introduces the information using the dialog and presses ok, the layout will automatically be updated and the newly calculated loan will be displayed. The user can, at any time, change between euros and dollars.



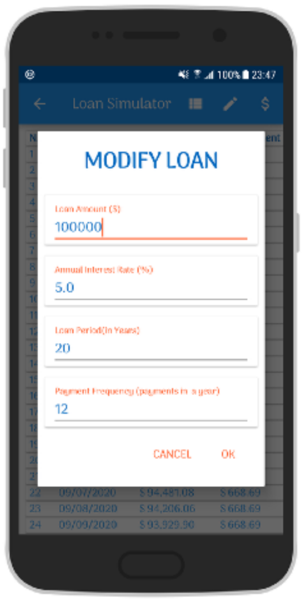
Loan Information

Loan Amount	100,000.00
Annual Interest Rate	5.0
Loan Period in Years	20
Payment Frequency	12
Start Date	09/10/2018

Loan Summary

Scheduled Payment	668.69
Total Interests	56,926.33

N.	Paym. Date	Beg.Balance	Sch.Payment
1	09/10/2018	$ 100,000.00	$ 668.69
2	09/11/2018	$ 99,747.98	$ 668.69
3	09/12/2018	$ 99,494.91	$ 668.69
4	09/01/2019	$ 99,240.78	$ 668.69
5	09/02/2019	$ 98,985.59	$ 668.69
6	09/03/2019	$ 98,729.34	$ 668.69
7	09/04/2019	$ 98,472.03	$ 668.69
8	09/05/2019	$ 98,213.54	$ 668.69
9	09/06/2019	$ 97,954.17	$ 668.69
10	09/07/2019	$ 97,693.62	$ 668.69



MODIFY LOAN

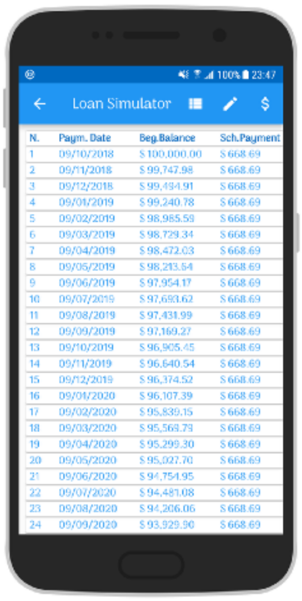
Loan Amount ($)
100000

Annual Interest Rate (%)
5.0

Loan Period(In Years)
20

Payment Frequency (payments in a year)
12

22	09/07/2020	$ 94,481.08	$ 668.69
23	09/08/2020	$ 94,206.06	$ 668.69
24	09/09/2020	$ 93,929.90	$ 668.69



N.	Paym. Date	Beg.Balance	Sch.Payment
1	09/10/2018	$ 100,000.00	$ 668.69
2	09/11/2018	$ 99,747.98	$ 668.69
3	09/12/2018	$ 99,494.91	$ 668.69
4	09/01/2019	$ 99,240.78	$ 668.69
5	09/02/2019	$ 98,985.59	$ 668.69
6	09/03/2019	$ 98,729.34	$ 668.69
7	09/04/2019	$ 98,472.03	$ 668.69
8	09/05/2019	$ 98,213.54	$ 668.69
9	09/06/2019	$ 97,954.17	$ 668.69
10	09/07/2019	$ 97,693.62	$ 668.69
11	09/08/2019	$ 97,431.99	$ 668.69
12	09/09/2019	$ 97,169.27	$ 668.69
13	09/10/2019	$ 96,905.45	$ 668.69
14	09/11/2019	$ 96,640.54	$ 668.69
15	09/12/2019	$ 96,374.52	$ 668.69
16	09/01/2020	$ 96,107.39	$ 668.69
17	09/02/2020	$ 95,839.15	$ 668.69
18	09/03/2020	$ 95,569.79	$ 668.69
19	09/04/2020	$ 95,299.30	$ 668.69
20	09/05/2020	$ 95,027.70	$ 668.69
21	09/06/2020	$ 94,754.95	$ 668.69
22	09/07/2020	$ 94,481.08	$ 668.69
23	09/08/2020	$ 94,206.06	$ 668.69
24	09/09/2020	$ 93,929.90	$ 668.69

3. TECHNICAL OVERVIEW

The app uses several APIs to create a smooth, steadily and rich user experience. We will have a look at the most important ones:

3.1. GOOGLE APIs

The app uses Google services (Maps and Places) to display all the information related to the real estates.

3.1.1. MAPS

Google Maps is used to display a map in a fragment (in the Main screen). This map will show the current location of the user and all the real estates' locations. If the real estate has already been sold, the pin will be red. If not, the pin will be purple.

3.1.2. PLACES

The app uses Google's Places API to fetch the real estate's information from Google servers. It uses 3 services:

- **Find Place Request**: this service is used to guarantee that the address of a real estate exists. If it does, the app will save the address and use the following services to get the latitude and longitude of the address and all nearby places that are points of interest.

- **Place Details Request:** this service allows to get specific information from a given place (its latitude and longitude).

- **Nearby Search Request:** this service allows to find all nearby points of interest to a given location

3.2. REAL ESTATE INFORMATION FETCHING PROCESS

3.2.1. REAL ESTATE STORED INFORMATION

In order to work properly, the app requires the following information given a real estate:

- Id: unique identifier generated by the app.
- Type: the type of the building (flat, apartment, etc.).
- Surface Area: the surface area of the real estate in Square meters.
- Price: the price of the real estate in dollars.
- Rooms: the rooms of the real estate (bedrooms, bathrooms and other rooms).
- Description: the description of the real estate.
- List of the Ids of the Images: list with all the ids related to the images also related to the real estate.
- Address: the address of the real estate (Street, Locality, City and Postcode). Once the real estate is saved in the database, the address cannot be changed.
- Latitude.
- Longitude.
- List of Ids of the Nearby Points of Interest: list with all the ids related to the points of interest also related to the real estate.
- Date Put: the date when the real estate was inserted into the database.
- Date Sale: the date when the real estate was sold (if it was).
- Found: a boolean value used to indicate if a real estate has been found by the search engine.

3.2.2. FETCHING PROCESS

The app will do the fetching process only in “Create New Listing Screen”. This screen allows inserting real estates in the database. In order to show them in a map, we need the location (latitude and longitude) of the real estate. That information can only be retrieved from Google servers if the address exists, so checking that the address is valid is the first step in the fetching process.

The fetching process follows the next pattern:

1st. The system will do a “Find Place Request”. This request will guarantee that the address is valid (if the app gets some information back). If it is, it will be stored and the next step will begin using the PlaceId retrieved bu Google Servers. If it is not, the process is interrupted and the app will notify the user that the address is not valid.

URL Request Example:

*"https://maps.googleapis.com/maps/api/place/findplacefromtext/json?
input=Corn%20Street,%20BS1%201SY
&inputtype=textquery
&key=API-KEY*

2nd. The system will do a “Place Details Request”. This request retrieves information related to the address inputted before. The app will store the latitude and longitude and will use this values for the next request.

URL Request Example:

*https://maps.googleapis.com/maps/api/place/details/json?
placeid=ChIJo8eWxtqNcUgRqzOCyl2ECWg
&key=API-KEY*

3rd. The system will do a “Nearby Search Request”. According to Google documentation, “A Nearby Search lets you search for places within a specified area”. It returns a number of points of interest lower than 30 with information related to each of them. The information stored by this service (and related to each point of interest) is the following:

- Place Id.
- Name.
- Address.
- Type.
- Latitude.
- Longitude.

URL Request Example:

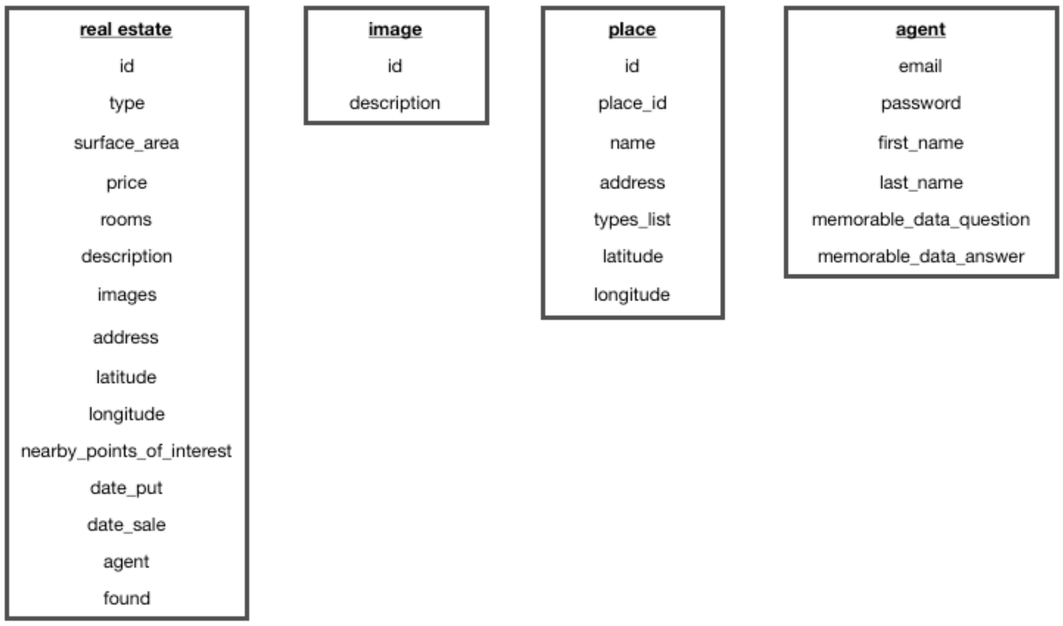
*https://maps.googleapis.com/maps/api/place/nearbysearch/json?location=0.0000,0.000
&rankby=distance&type=restaurant
&key=API-KEY*

3.3. LOCAL STORAGE

3.3.1. LOCAL DATABASE (ROOM)

The app stores the necessary information in a local database. Then uses several "Jetpack Components (Room, LiveData and ViewModel)" to display the information.

The data is stored in several tables using Room ORM library. They follow the next structure:



- Real Estate Table:
This table stores information about the real estates inputted by the user. For more information, see "3.2.1. Real Estate Stored Information".

- Image Table:
This table stores information about the images the user inputs when creating a new listing.
- Id: unique identifier of the image.
- Description: description of the image.

When inserting a new listing with an image into the database, a new file with the Bitmap will be created in "Images Directory" using the image id (See "3.3.4. Internal Storage").

- Place Table:
This table stores information about the nearby points of interests retrieved by the Google Servers related to a specific location.
- Id: unique identifier of the place.
- PlaceId: Google Servers' identifier of the place.
- Name: name of the place according to Google Servers.
- Address: address of the place according to Google Servers.
- List of Types: all the different types of point of interest the place matches according to Google Servers.
- Latitude.
- Longitude.

- Agent Table:

This table stores information related to the user/users.

- Email: email of the user.
- Password: password associated with a specific email
- First Name: first name of the user.
- Last Name: last name of the user.
- Memorable Data Question: question the will be shown if the user has forgotten his/her password and wants to get it back.
- Memorable Data Answer: answer that is necessary to get the user's password back if he/she forgot it.

3.3.2. SHARED PREFERENCES

The app also stores information using Shared Preferences Android feature which allows accessing this information in a fast a clean way. The information stored in Shared Preferences is the following:

- Currency: the app stores the current currency the app is using. For that, it uses an "Integer" value (for choosing between 2 values the app could use a "Boolean" value for efficiency purposes but an "Integer" leaves the possibility to add more currencies in the future). The currency value allows the app to display the information in dollars or euros in each screen.

- Agent Data: the app stores the agent data in order to display information in the authentication screen (email).

3.3.3. DATA REPOSITORY

The app uses a "Data Repository" to handle the communication between the Local Database (Room) and the Views (elements that allow user interaction with the application) of the application via the ViewModel.

The "Data Repository" has several functions:

- It stores a cache object (and images and nearby places objects) that will be used during the "Creation and Editing" of listings. When entering the "Create New Listing Screen" (or "Edit Screen"), a new cache object will be created in the "Data Repository". This object will be updated with the information inputted by the user in both screens, "Create New Listing" and "Add Photos Screen". When the insertion (or editing) process starts, the information of the cache object is the one that will be pushed to the local database.

- It stores a cache of Bitmaps with a maximum size. When the maximum memory of this cache is reached by the objects it contains (the maximum memory depends on the device and is calculated at runtime) the first object of the data structure is deleted. This cache is necessary due to several reasons:

1) Memory Issues: the Bitmap objects occupy much memory. Using a cache that recycles itself we can avoid memory problems.
2) When the user inputs photos but has not inserted the object in the database, the images are stored in this cache.
3) It allows loading images faster (instead of reading them from the disk). If the image is available in the cache, the system will read it from there.

- It handles the communication between the database and the user entry points (Views).

3.3.4. INTERNAL STORAGE

The app stores the Bitmaps (images) in the internal storage. Each time a new listing with any photo is created, the photo receives an id which is used to name the file that will be created in the directory. The same id is used to retrieve the Bitmap each time it is necessary.

3.4. SEARCH ENGINE

The app has a "Search Engine" that allows the agents to search for real estates. It displays information on the screen depending on the real estates already inputted to the database.

The following options will be displayed in checkboxes:
- Type of Building.
- Cities.
- Localities.
- Sale Estate: the app will display both options, "On Sale" and "Sold".
- Types of Points of Interest (if the user checks a checkbox related to a point of interest, the Search Engine will only find those real estates that have at least one point of interest of the types selected).

The following options will be displayed in seek bars:
- Price.
- Surface area.
- Number of rooms.
- Amount of photos.

The real estate must match at least one criteria of each group in order to be found by the Search Engine (if, for instance, the user chooses two "Types of Points of Interest" and the real estate only matches one, it will still be shown in the results).

3.5. LOAN SIMULATOR

The app incorporates a “Loan Simulator” that allows the user to create a loan example and display. The user can input 4 different values:

- Loan Amount.
- Annual Interest Rate.
- Loan Period in years.
- Payment frequency

The maximum values are:
- Loan Amount: 1,000,000 $.
- Interest Rate: 15%.
- Loan Period in years: 20.
- Payment Frequency (payments in a year): 12.

The minimum values are:
- Loan Amount: 50,000 $.
- Loan Period in years: 5.
- Payment Frequency (payments in a year): 1.

The user can swap between euros and dollars, but still the maximum and minimum values for the loan amount are the same.

3.6. APP PERMISSIONS

In order to work correctly, the app requires the user to give access (give permission) to two features of the mobiles device: access to the current location and access to the internal storage. The user can give both permissions to the app when the dialogs appear on the screen immediately after the app is launched.

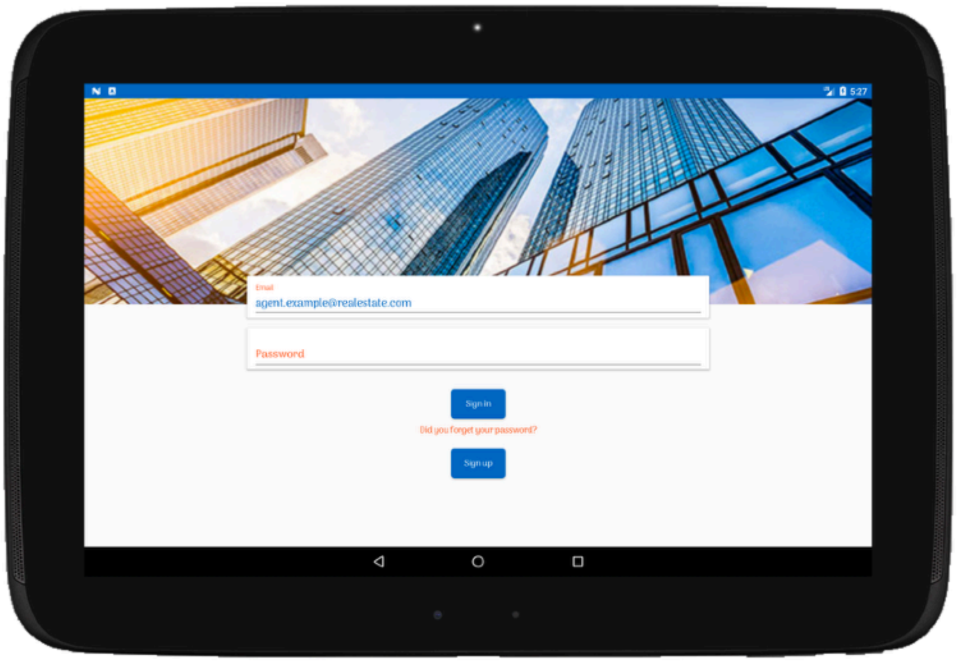
3.7. LIBRARIES

[ButterKnife](https://github.com/JakeWharton/butterknife/)
[Android Support Library](https://developer.android.com/topic/libraries/support-library/)
[Glide](https://github.com/bumptech/glide/)
[Google Maps Platform](https://cloud.google.com/maps-platform/)
[Retrofit](https://github.com/square/retrofit/)
[Retrofit Converter Gson](https://github.com/google/gson)
[RxJava 2 - Retrofit Adapter](https://github.com/square/retrofit/tree/master/retrofit-adapters/rxjava2)
[RxJava - Room](https://medium.com/google-developers/room-rxjava-acb0cd4f3757)
[RxBinding](https://github.com/JakeWharton/RxBinding)
[Gson](https://github.com/google/gson/)
[RxJava](https://github.com/ReactiveX/RxJava)
[RxAndroid](https://github.com/ReactiveX/RxAndroid)
[Room](https://developer.android.com/topic/libraries/architecture/room)
[ViewModel](https://developer.android.com/topic/libraries/architecture/viewmodel)
[LiveData](https://developer.android.com/topic/libraries/architecture/livedata)
[Android-Storage](https://github.com/sromku/android-storage)
[Android Debug Database](https://github.com/amitshekhariitbhu/Android-Debug-Database)

ANNEX

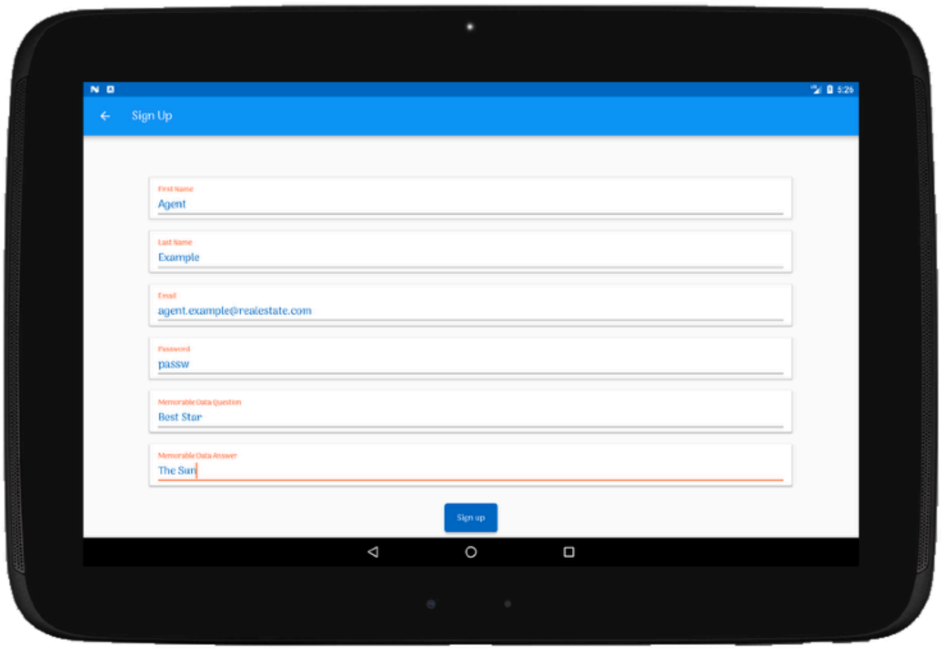
TABLET SCREENSHOTS

The app is prepared to work in both handsets and tablet devices. The user experience is similar but the “Main Screen” displays the information differently (mixing Main Screen and Detail Screen in the same layout). The next images show the tablet layout for some screens. They are available in both Vertical and Landscape modes.



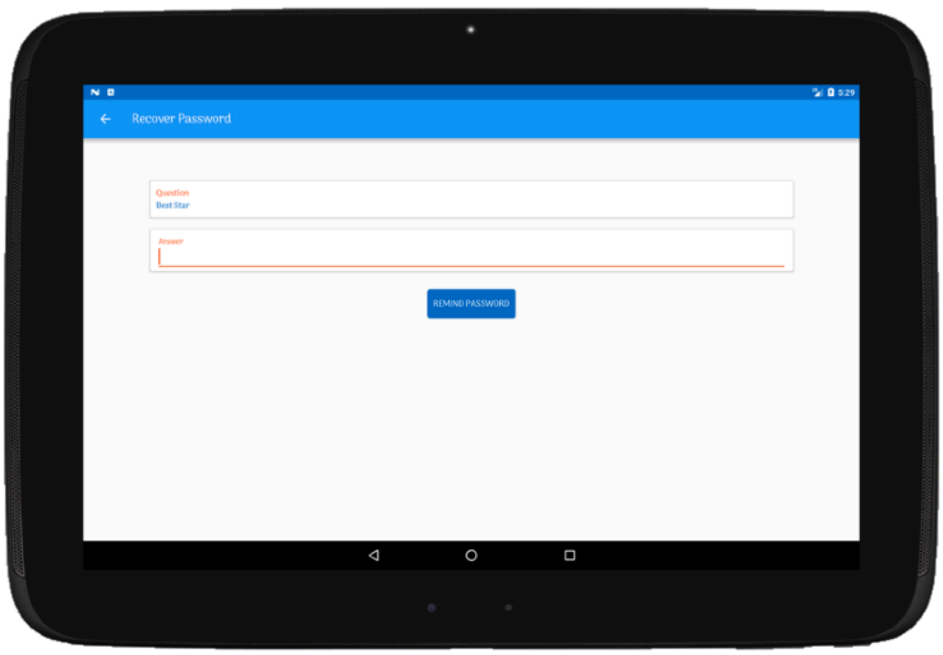
Email
agent.example@realestate.com
Password

Authentication Screen.



First Name
Agent
Last Name
Example
Email
agent.example@realestate.com
Password
passw
Memorable Data Question
Best Star
Memorable Data Answer
The Sun

Sign Up Screen

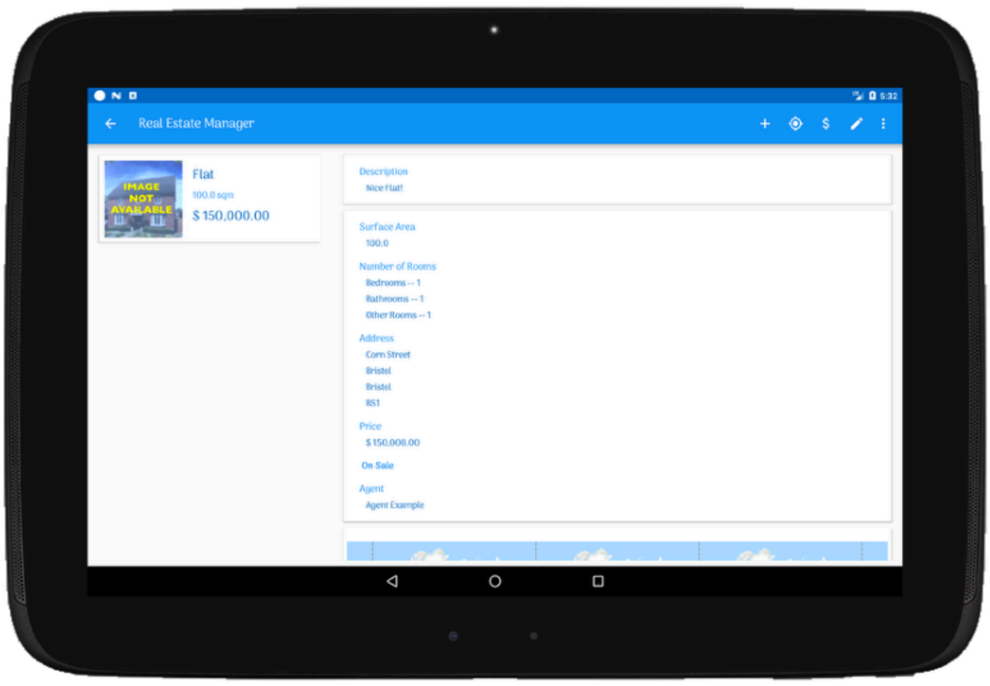


Recover Password

Question
Best Star

Answer

Forgot Password Screen



Real Estate Manager

Flat
100.0 sqm
$ 150,000.00

Description
Nice Flat!

Surface Area
100.0

Number of Rooms
Bedrooms -- 1
Bathrooms -- 1
Other Rooms -- 1

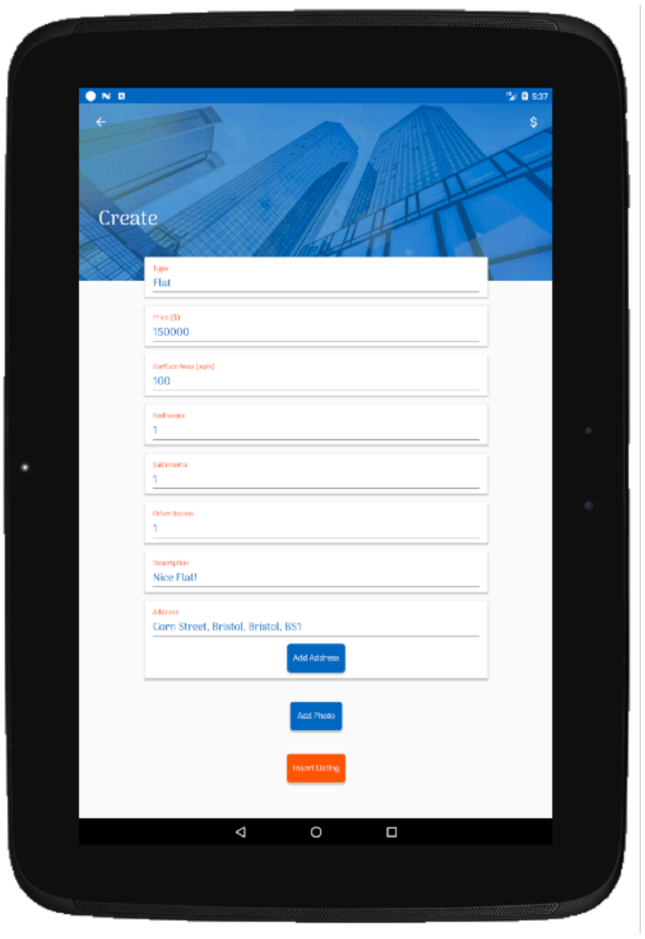
Address
Corn Street
Bristol
Bristol
BS1

Price
$ 150,000.00

On Sale

Agent
Agent Example

Main Screen (Detail Screen included in Tablets)



Create

Type
Flat

Price ($)
150000

Surface Area (sqm)
100

Bedrooms
1

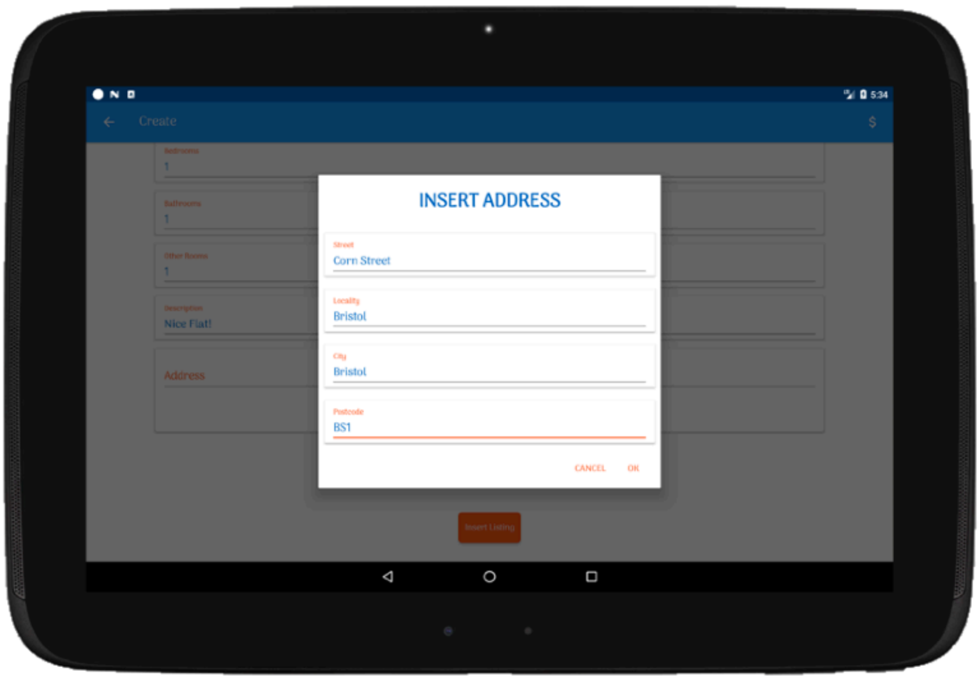
Bathrooms
1

Other Rooms
1

Description
Nice Flat!

Address
Corn Street, Bristol, Bristol, BS1

Create New Listing Screen (also available in landscape)



INSERT ADDRESS

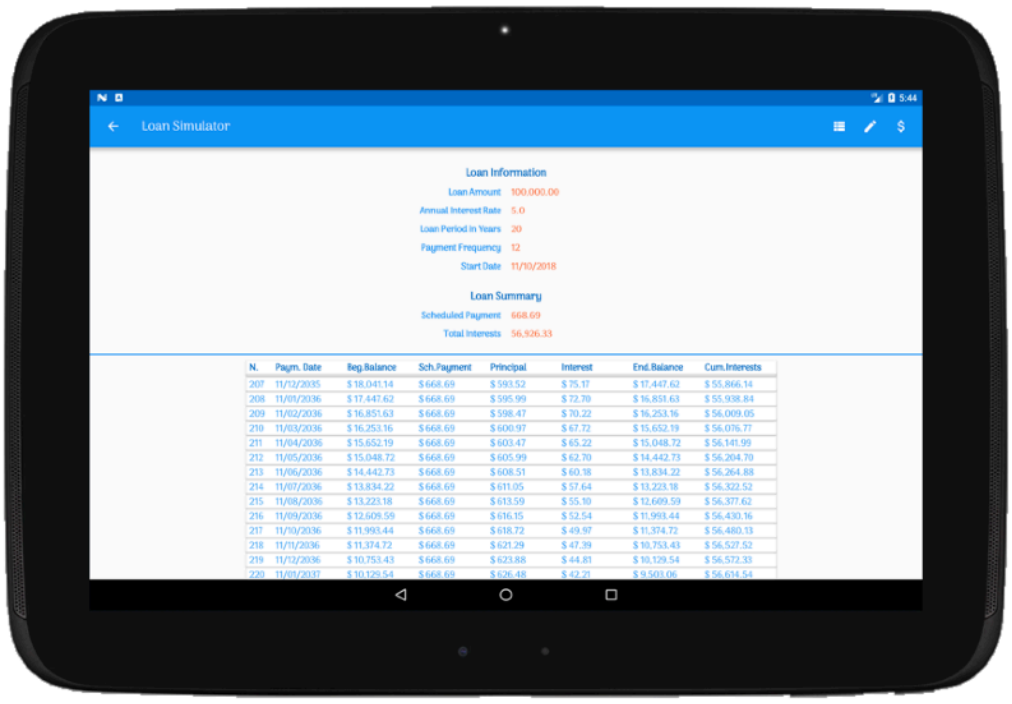
Street
Corn Street

Locality
Bristol

City
Bristol

Postcode
BS1

Insert Address Dialog



Loan Information

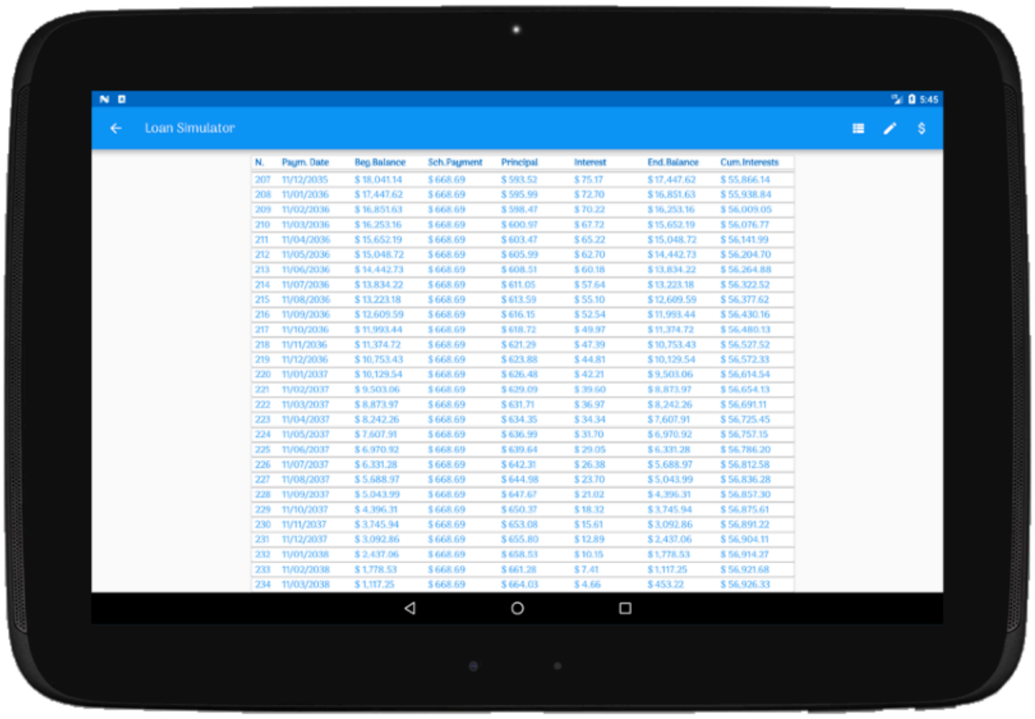
Loan Amount	100,000.00
Annual Interest Rate	5.0
Loan Period In Years	20
Payment Frequency	12
Start Date	11/10/2018

Loan Summary

Scheduled Payment	668.69
Total Interests	56,926.33

N.	Paym. Date	Beg.Balance	Sch.Payment	Principal	Interest	End.Balance	Cum.Interests
207	11/12/2035	$ 18,041.14	$ 668.69	$ 593.52	$ 75.17	$ 17,447.62	$ 55,866.14
208	11/01/2036	$ 17,447.62	$ 668.69	$ 595.99	$ 72.70	$ 16,851.63	$ 55,938.84
209	11/02/2036	$ 16,851.63	$ 668.69	$ 598.47	$ 70.22	$ 16,253.16	$ 56,009.05
210	11/03/2036	$ 16,253.16	$ 668.69	$ 600.97	$ 67.72	$ 15,652.19	$ 56,076.77
211	11/04/2036	$ 15,652.19	$ 668.69	$ 603.47	$ 65.22	$ 15,048.72	$ 56,141.99
212	11/05/2036	$ 15,048.72	$ 668.69	$ 605.99	$ 62.70	$ 14,442.73	$ 56,204.70
213	11/06/2036	$ 14,442.73	$ 668.69	$ 608.51	$ 60.18	$ 13,834.22	$ 56,264.88
214	11/07/2036	$ 13,834.22	$ 668.69	$ 611.05	$ 57.64	$ 13,223.18	$ 56,322.52
215	11/08/2036	$ 13,223.18	$ 668.69	$ 613.59	$ 55.10	$ 12,609.59	$ 56,377.62
216	11/09/2036	$ 12,609.59	$ 668.69	$ 616.15	$ 52.54	$ 11,993.44	$ 56,430.16
217	11/10/2036	$ 11,993.44	$ 668.69	$ 618.72	$ 49.97	$ 11,374.72	$ 56,480.13
218	11/11/2036	$ 11,374.72	$ 668.69	$ 621.29	$ 47.39	$ 10,753.43	$ 56,527.52
219	11/12/2036	$ 10,753.43	$ 668.69	$ 623.88	$ 44.81	$ 10,129.54	$ 56,572.33
220	11/01/2037	$ 10,129.54	$ 668.69	$ 626.48	$ 42.21	$ 9,503.06	$ 56,614.54

Loan Simulator Screen



N.	Paym. Date	Beg.Balance	Sch.Payment	Principal	Interest	End.Balance	Cum.Interests
207	11/12/2035	$ 18,041.14	$ 668.69	$ 593.52	$ 75.17	$ 17,447.62	$ 55,866.14
208	11/01/2036	$ 17,447.62	$ 668.69	$ 595.99	$ 72.70	$ 16,851.63	$ 55,938.84
209	11/02/2036	$ 16,851.63	$ 668.69	$ 598.47	$ 70.22	$ 16,253.16	$ 56,009.05
210	11/03/2036	$ 16,253.16	$ 668.69	$ 600.97	$ 67.72	$ 15,652.19	$ 56,076.77
211	11/04/2036	$ 15,652.19	$ 668.69	$ 603.47	$ 65.22	$ 15,048.72	$ 56,141.99
212	11/05/2036	$ 15,048.72	$ 668.69	$ 605.99	$ 62.70	$ 14,442.73	$ 56,204.70
213	11/06/2036	$ 14,442.73	$ 668.69	$ 608.51	$ 60.18	$ 13,834.22	$ 56,264.88
214	11/07/2036	$ 13,834.22	$ 668.69	$ 611.05	$ 57.64	$ 13,223.18	$ 56,322.52
215	11/08/2036	$ 13,223.18	$ 668.69	$ 613.59	$ 55.10	$ 12,609.59	$ 56,377.62
216	11/09/2036	$ 12,609.59	$ 668.69	$ 616.15	$ 52.54	$ 11,993.44	$ 56,430.16
217	11/10/2036	$ 11,993.44	$ 668.69	$ 618.72	$ 49.97	$ 11,374.72	$ 56,480.13
218	11/11/2036	$ 11,374.72	$ 668.69	$ 621.29	$ 47.39	$ 10,753.43	$ 56,527.52
219	11/12/2036	$ 10,753.43	$ 668.69	$ 623.88	$ 44.81	$ 10,129.54	$ 56,572.33
220	11/01/2037	$ 10,129.54	$ 668.69	$ 626.48	$ 42.21	$ 9,503.06	$ 56,614.54
221	11/02/2037	$ 9,503.06	$ 668.69	$ 629.09	$ 39.60	$ 8,873.97	$ 56,654.13
222	11/03/2037	$ 8,873.97	$ 668.69	$ 631.71	$ 36.97	$ 8,242.26	$ 56,691.11
223	11/04/2037	$ 8,242.26	$ 668.69	$ 634.35	$ 34.34	$ 7,607.91	$ 56,725.45
224	11/05/2037	$ 7,607.91	$ 668.69	$ 636.99	$ 31.70	$ 6,970.92	$ 56,757.15
225	11/06/2037	$ 6,970.92	$ 668.69	$ 639.64	$ 29.05	$ 6,331.28	$ 56,786.20
226	11/07/2037	$ 6,331.28	$ 668.69	$ 642.31	$ 26.38	$ 5,688.97	$ 56,812.58
227	11/08/2037	$ 5,688.97	$ 668.69	$ 644.98	$ 23.70	$ 5,043.99	$ 56,836.28
228	11/09/2037	$ 5,043.99	$ 668.69	$ 647.67	$ 21.02	$ 4,396.31	$ 56,857.30
229	11/10/2037	$ 4,396.31	$ 668.69	$ 650.37	$ 18.32	$ 3,745.94	$ 56,875.61
230	11/11/2037	$ 3,745.94	$ 668.69	$ 653.08	$ 15.61	$ 3,092.86	$ 56,891.22
231	11/12/2037	$ 3,092.86	$ 668.69	$ 655.80	$ 12.89	$ 2,437.06	$ 56,904.11
232	11/01/2038	$ 2,437.06	$ 668.69	$ 658.53	$ 10.15	$ 1,778.53	$ 56,914.27
233	11/02/2038	$ 1,778.53	$ 668.69	$ 661.28	$ 7.41	$ 1,117.25	$ 56,921.68
234	11/03/2038	$ 1,117.25	$ 668.69	$ 664.03	$ 4.66	$ 453.22	$ 56,926.33

Loan Simulator Screen (loan table extended)