Polytechnic University of Timișoara

Faculty of Automation and Computers

**Department of Computer and Information Technology**

**Split the Bill**

Bachelor Thesis

*Candidate:*

**Emese Máthé**

*Supervisor:*

Lect. dr. ing. **Oana Iulia Casandra HOLOTESCU**

Timișoara

2019

**`**

**Contents**

1. Introduction
   1. Problem Statement
   2. Scope of the project
2. Use cases
   1. Application Description
   2. Use cases
   3. Application Features
3. Application design
   1. Architecture
   2. Strucutre of the data model
4. Technologies
   1. Flutter SDK
      1. Dart
      2. Scope Models
   2. Firebase
      1. Authentication
      2. Real Time Databse
      3. RESTful API
5. Application Implementation
   1. Account
      1. Registration
      2. Authetication
      3. Editing user infromation
      4. Signout
   2. Billboard page
   3. Supply page
   4. Chores page
   5. More page
      1. Apartment infromation
      2. Change theme
      3. Report issue
6. Examples & Screen shots
7. Conclusion & Perspective
   1. Conclutions
   2. Contribution
   3. Future work
8. References

1 **Introduction**

Problem Statement

Nowadays a lot of people move into the city, at first people think of getting their own place but it is rather expensive to do so, and for the sake of saving up money people search for rooms in shared flats. Living with more than three people in an apartment can become difficult, especially when it comes to sharing the expensies and splitting the money for the cleaning supplies that everybody uses. It is also difficult to keep track of who contributed to the cleaning of the common areas in the flat.

Conflict can arrise from tha lack of communication between the roommates, because of their busy schedule, and they have no time to check in with each other, thus making it hard to live with one another. Such as having some friends over without asking, or having forgotten that the landlord was coming and people did not know about it. These can lead to conflicts and missunderstandings, thus forcing some people to find a new place to live in.

In order to prevent such things the people who live in shared flats must well organized, and have to have a good money management skills, but a lot of people who live in shared flats are young adult who do not yet posses there skill sets, thus making it difficult to get along with other and split everything correctly.

1.2 Scope of the Project

The scope of the project is to help people who have roommates organize themselves more easily when it comes to trivial matters such as chores, or splitting the money for the cleaning supplies. For this reason I was thinking of develeping a mobile application, the reason for developing a mobile application is because nowadays people are on their phones all the time, and almost everybody has a smart device. Because of this reason it is easier to check the application if something has came up or something has changed or not.

The reason why I did not develop a web page is because users prefer native application instead of web applications. There are a lot of reasons why this is the case, one of the reason is that it is faster and native application are more easily personalized, and users tend to like personalizing their application. Another reason for prefering application is that is more secure, and they can work up until a point without internet connection.

The mobile application is called split the bill, this application helps with spliting the chores and cleaning supplies between the people living in the same flat, and it also has a billboard page for putting announcements for people to see. In short the application will be useful for people who want to keep track of the things above and help them organize themselves more easily.

2. Use cases

Use cases help in organizing system requriments, and services that the software product will provide. They help undestanding how the user will interract with the system, and how the modules of the system will interract with each other.

2.1 Application Description

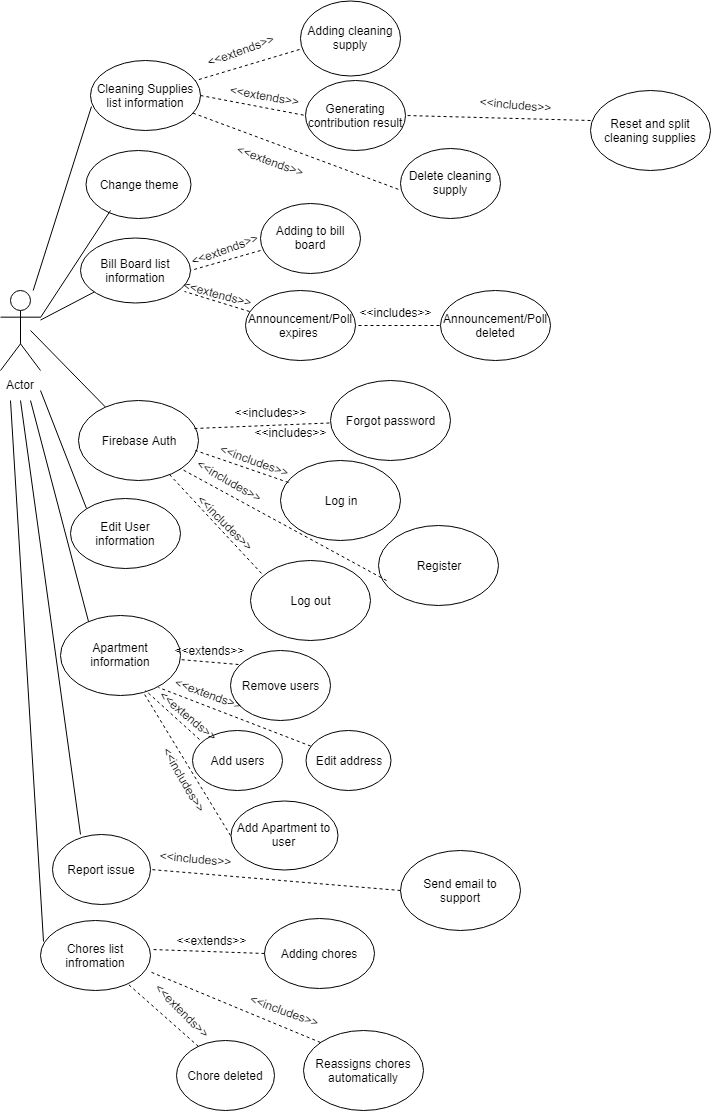
The application is called Split the Bill, it’s main focus is to help mange money resources for people who live in shared accomodation, thus making their life easies when it comes to splitting equally the money for cleaning supplies.

The application offers more than this, it also helps the people living togheter organize themselves when it comes to spliting the chores, and again it help with time mangment. The chores will be reassgined automatically to different people after a certain amount of time that the users in the apartment set.

It is also useful because the people who use this application can also make announcements and polls that the other can see. The announcements are for example when the landlord is coming over to collect the rent, and people need to be home, and someone writes this announcements in the bill board page of the application. People can also ask their roommates question if they can have some friends over, or if they can host a party, and the other people can vote with yes or no.

The application will be a native application, for both Android and IOS devices.

These are the main features of the application that the application is trying to solve and help people get along more easily, and helps them organize themselves more efficently.

2.2. Use case

An important use case is the authetication of the user this is done using firebase. When a user register they will have to provide the following information: first name, last name, nickname, phone number, a valid email address and a password. When sigin in the users will have to use their email address, because this one of the way you can autheticate using firebase. The users will be able to log out from the application. In case the user forgets to their password they can complete a form with their registered email address, to where the user will recieve a message with a link where they can reset their password.

Users will be able to view their user information, they will also be able to edit some of the information if it is outdated or it was misstyped. The followind data can be edited: first name, last name, nick name, phone number.

Another use case is the personalization of the application, the user can enter a dark theme mode, or if they choose the light theme mode they can edit the color of the application.

Reporting some issues is also possible by completing a form, the form contains three fields, a title, a description field where the user will be able to detail the what is wrong with the application, and a type. The type can be as follows: crash, bug or other.

The cleaning supplies information represent a list for all the cleaning supplies for the apartment that the user is apart of. The elements of the list will contain the following information:name, price, the nickname of the userthe type which represent where the supply will be used, the types can be of kitchen, bathroom, living or other. The user can add and delete from this list, and they can also reset the list meaning all the information will be deleted. The deletion will be possible by swiping the element from right to left. Another important use case is the generation for the contribution of each user, the generated data will be as follows: the total amount spent will be calculated and also each user how much paid from the total amount. The calculated result for each user will be the amount that user spent and the precentage from the total amount.

Billboard announcement infromation use case will provide the user the possibility to put an announcement or a poll for the other users in that apartment to see. The billboard will contain a list of announcements, each element regardless of their type will contain a title, a description, the author of the post. If it is a simple announcment it will only have the above information, on the other hand if it is a poll type than the users will be able to vote with yes or no, and the result of the application will be shown and updated each time a user votes. The user will be able to add and delete from this list as in the cleaning supply page. When addin a new element the user must specify the availability of the announcemnts, because after the time specified by the user the element will be deleted automatically by the system.

The chores list informatio list is another important aspect, here the user will be able to view the list of chores that are attached to that specific apartment. The users can add and delete from this list at any given time, deletion will happen the same way as in the bill board use case. Each element of the list will show the following informtation: the name of the chores, a description of the element, the current assigne. After the given amount of time the chores are reassigned to different people, this can be set in the page, and for all the element in the list the changing interval will be the same.

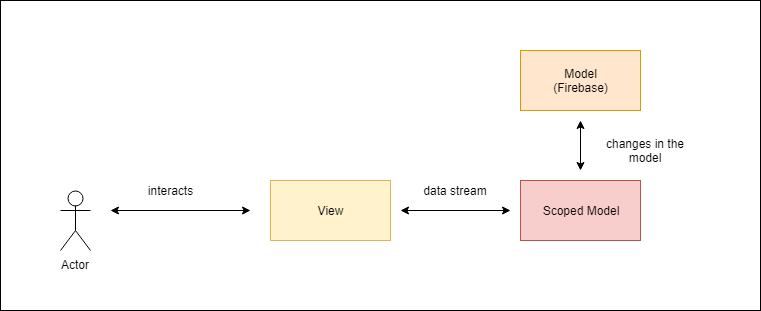
2.3 Application Features

In the table bellow there is the list of features that the application should contain for it to be able to function well.

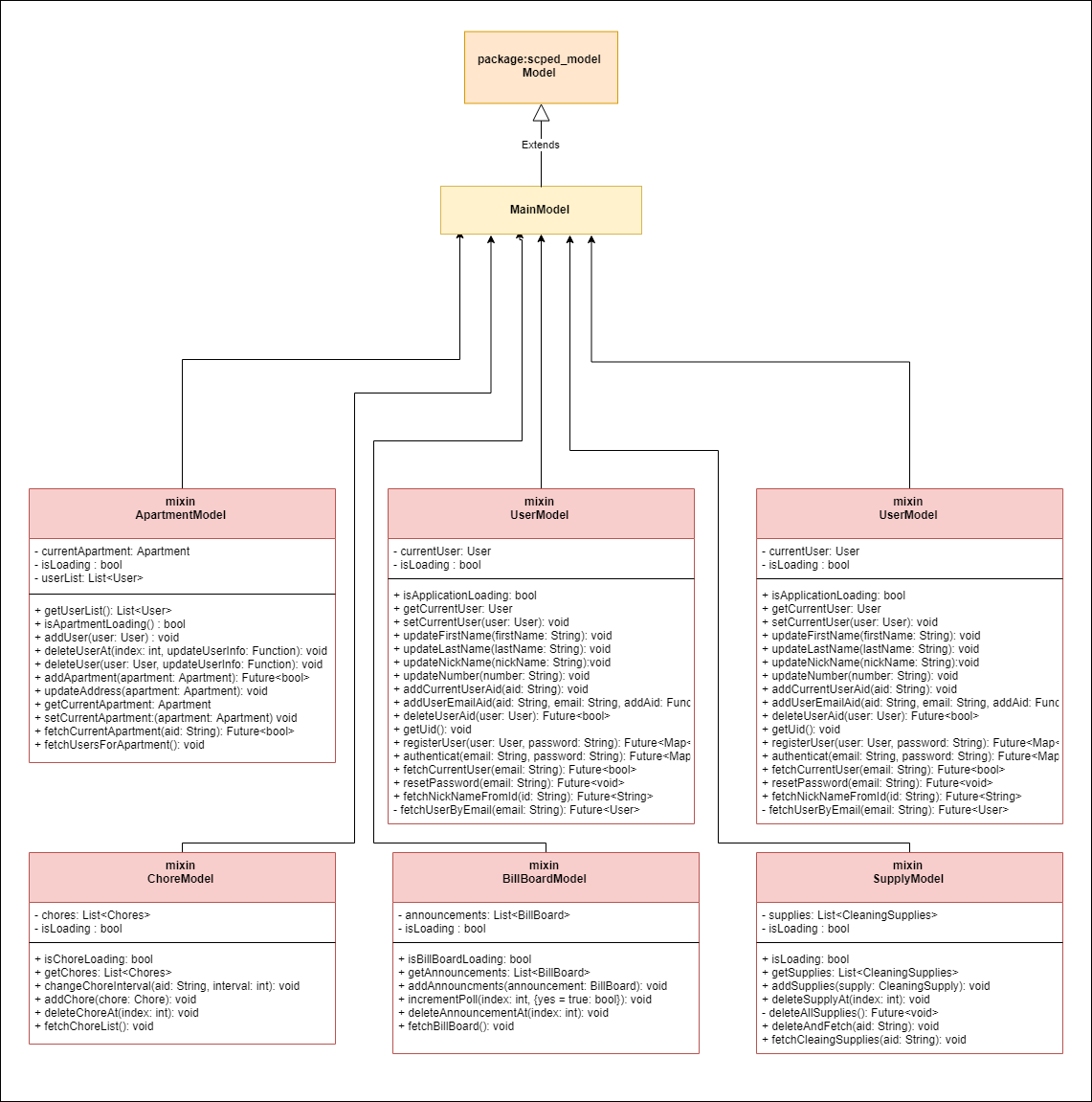
1. Users should be able to register to the application, and when doing so they should give some information about themselves.
2. Users should not be able to use the application without being autheticate
3. In case users should forget their password they are able to reset it.
4. Whe user encounter some problems they can report via gmail.
5. Personalizing the theme of the application.
6. Editing user information is editable in some of the information changed.
7. The application has a bottom navigation drawer, and has multiple pages.
8. The application is functional on android as well as ios platform
9. The application allows the user to add and delete chores, cleaning supplies and billboard announcements
10. The application also generates the contribution of each user in percantage and calculates it is total contribution
11. The billboard announcements can be of two types, simple or a poll, where users can vote with yes or no.
12. The chores are automatically reassigned to the users after a give amount of time.
13. The billboard announcements expires after a given amount of time and they are automatically deleted.
14. Interface should be simple and user friendly.

3. Application design

3.1 Architecture

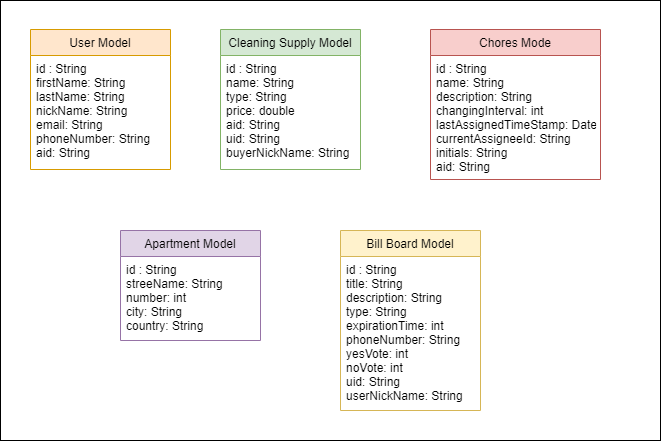
For this application I used layer architecture, in the application I have three layers, a presentation layer, a business logic layer and for the backend I have used firebase, the architecture is shown in the diagram bellow:

The presentation layer repestins the previously mentioned widget tree, that rerenders each time some changes appear in the business logic. This is possible using Scope Model and having references to function using the Scope Model Descendant object that wraps around the widgets. In the business logic section I handle the fetching and writing of the data to firebase. In this layer I also implament some functionality like . The backend layer is completly handeled by Firebase.

Split the bill also takes advante of event-based programming, which is useful when objects change their state. In case of flutter, as mentioned before we have Scope Models and Scope Model Decendants, and using these I can update the state of the model, from the presentation layer and redrawing the layout. This is possible because Scope Model notifies the Scope Model Decendants that can found lower down the widget tree that some changes occoured.

3.2 Structure of the data model

The data that is saved to the Firebase data model is in json format, but in the application there are some object that represent by some custom objects. The classes that define these models have a function that converts the data into a Map, that is accepted by Firebase.

There are five such models throughout the project. In the diagram bellow you can see the models and their fields.

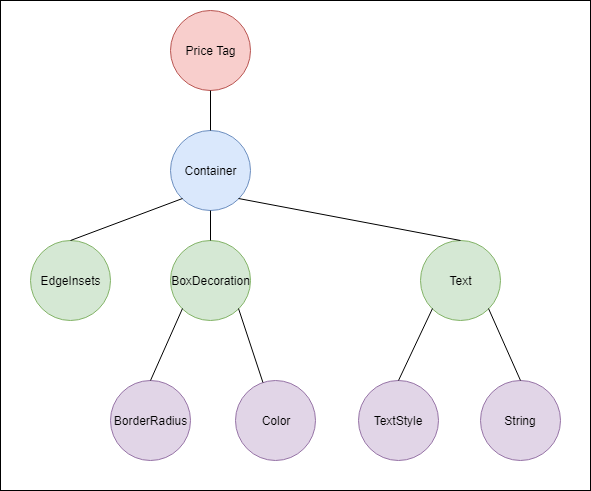
1. Technologies

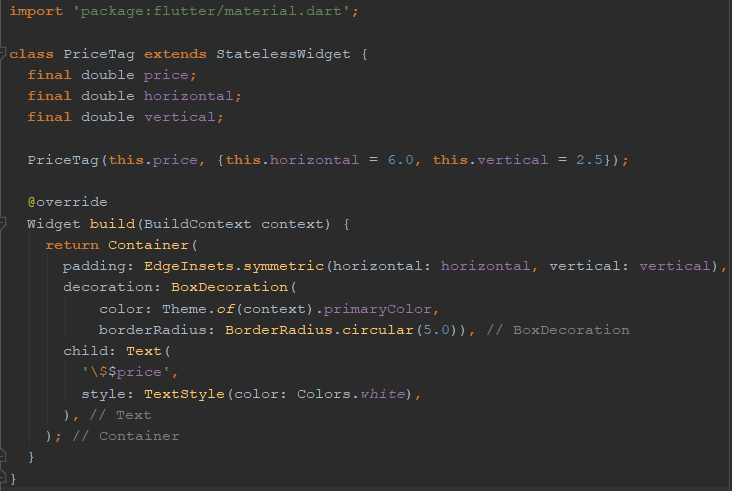
4.1 Flutter SDK

Flutter is an open source  cross-platform mobile app development framework, with which the developers can create native applications  for both the Android and IOS platform, thus the framework has one codebase from which it generates two native application. For Android the generated code is in Java and for IOS the generated code is Swift. This framework is created by Google. The programming language used by Flutter is called Dart.

The framework doesn’t depend on the widgets that are used by the platforms, because it only needs the canvas to draw it’s own rendering.

Flutter implements the OOP principle of the composition over inheritance. Composition and inheritance are both fundamental concepts of object-oriented programming. Composition means that a class has a reference to some other class, thus establishing a HAS-A relationship between these two classes. Inheritance on the other hand is a process where the class that inherits the super class and it’s public and protected methods and fields, this principle establishes an IS-A relationship between the parent and the child class.

A flutter application consists of a widget tree, so the application consists of small widgets which creates a complex user interface. A widget can be of two types. Stateless Widget which doesn’t change thus it is a final widget, and the Stateful Widget, this type of widget changes states by calling the function setState() which notifies the listeners that the state of the widget has changed thus the widget is redrawn.Bellow you can see an example of a widget tree and the code sample for which the tree is built

Because flutter is open-source there are a lot of packages that the developer can use to build the application faster, these packages can be added in the pubspec.yaml file of the projects, thus these packets will be added only to the current project not to flutter. In my diploma project I used packets such as these, for example the scope\_model, percent\_indicator, http, url\_launcher packets. In the above mentioned you can also add other dependencies such as the assets. Assets are considered to be images, or fonts etc.

* + 1. Dart

Dart is an object oriented programming language that uses both JIT(Just In Time) compilation and as well as AOT(Ahead-Of-Time) compilation.

Just In Time compilation is a technique that converts the bytecode that was generated by the compiler into native machine code at run time thus getting the name just in time compilation.

Ahead of Time compilation is a different compilation technique this one compiles the code into native machine code before the code starts to run in the environment.

* + 1. Scope Models

As mentioned before the Scope Model is one of the third-party packages that can be installed by adding the dependency into the pubspec file. This package helps pass data within the widget tree from the parent to the child nodes. And when the data is changed these child nodes will be updated and redrawn.

The Scope Model can be accessed in two ways, by using the widget ScopeModelDescendant, in which case the widget that is the descendant of the model will be rebuilt, or by statically accessing the ScopeModel.of(context) method.

Firebase

Firebase is a platform created by Google, that can be categorized as BaaS (Backend as Service). Firebase offers a lot of features, these features help developers to create application that respond fast, thus creating a great user exprience.

This platform offers a lot of analytics, these are categorised into tree group the development group, which consists of a NoSql real time database, authetication, cloud messaging, storage, hosting, test las and crash reporting. Another of the grpups is the grow froup which focuses on the notifications, remote config, app indexing, dynamic links, invites, adWords. The last category is the earn group which consists of AdMob, so that developers can earn money from in-app adds.

4.2.1 Authentication

For the diploma project I used the authetication, for users to be able to log in to the application, using firebase so that i do not save the user passwords into the database, and letting firebase handle user authetication.

Split the bill tries to autheticate the user via a login page, the credential of the user in our case email and password are sent to firebase, firebase then sends a response wheter the action was succesful or not, the token that is sent back to the application is encoded and if this token is manipulated firebase will know that the user that is trying to autheticate is fake or someone is trying to steal data.

4.2.2 Real Time Database

Another feature that was used in my diploma project is the real-time database, this type of database is a NoSql database, this means that the data is stored in a file with the JSON format.This mean that the datamodel is structured as a tree, and the data is saved by adding children to the parent node. The actual data that the user saves can be found at the leafs of the data model.

One of the main benefit of this type of database is that the data is synchonised in a few ms, thus making the mobile application responsive. When users are using a moblie application they like it to be fast, andto get the data as fast as possible. If an application is slow users tend to search for other options, ones that are faster.

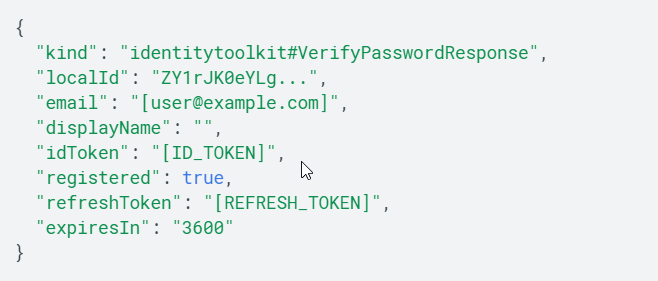
Another great feature of firebase is when that device looses it’s connection to the internet the events are still fired, so when the device goes back online the data will be merged to the database.

4.2.3 RESTful API

REST is short for Representational State Transfer, this API is used with http requests, such as GET, POST, PUT, DELETED. This method is used mostly for communication with web services. The reason is user RESTful API is that this was the server doesn’t exactly know about your application, it doesn’t have concrete infromation about the application itself, unlike with Android development where the user has to connect the application directly to firebase. The application uses some endpoint to firebase for adding, chanding, fetching and deleting data.

GET is used for fetching data from the data model, on the other hand POST is used to created or add new data, while PUT is called when an already existing data needs to changed. DELETE as the name suggests it is used to delete data. The data used by these requests can be block, file or even object, in split the bill the data is encoded in a map, that is formated so that it fits the real time database data model of json format.

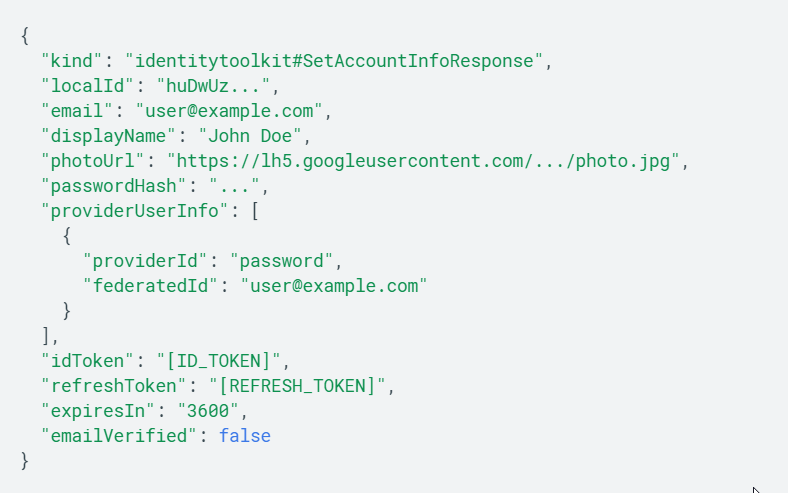
One of endpoint used in split the bill is when user is regitering, signing in, or requesting password reset is provided by google. This http link is an https which means it’s a secure link, the link contains an API key that can be found in the firebase project, that is created for this specific mobile application. After login in the user, we get a response if the authentication was succeful or not, if not we get the following error codes: EMAIL\_NOT\_FOUND, INVALID\_PASSWORD, USER\_DISSABLED. A sample of the response body:



User registration has a similar procedure, but it has different error codes: EMAIL\_EXISTS, OPERATION\_NOT\_ALLOWED, TOO\_MANY\_ATTEMPT\_TRY\_LATER. A sample for registation response body:



Another feature that uses that uses a similar link provided in the documentation is the email reset that is sent to the email of the user. The response body of this later feature is more complex than the one preveiously discussed. This service provided by firebase send the user an email with a link, if the link is accessed the user has to complete a form with the new password, and thus changing his or her password. Below you can see an example of such response:



Another enpoint that is used in split the bill application is the https link for the real time database that is part of the firebase project, from where we got the API KEY mentioned above.

1. Application Implementation

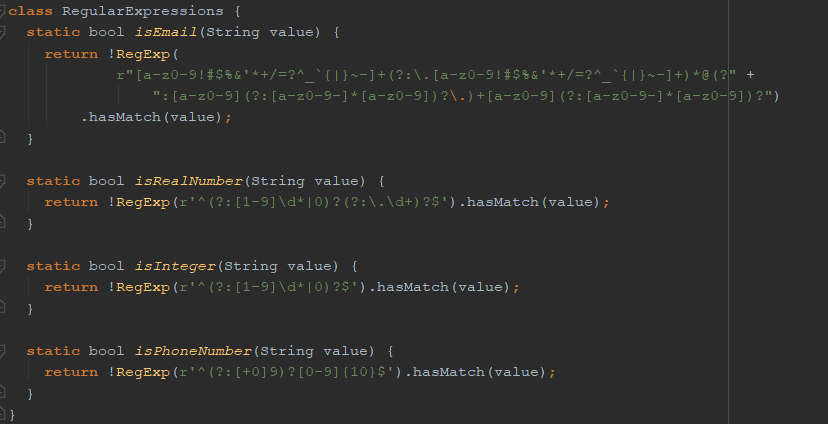
In this chapter I will explain in more detail about how the application was implemented.

Account

Each user has to have an account, they cannot use the anonymous authentication. This means that the user have to register and then later on authenticate before being able to access or view the application.

* + 1. Registration

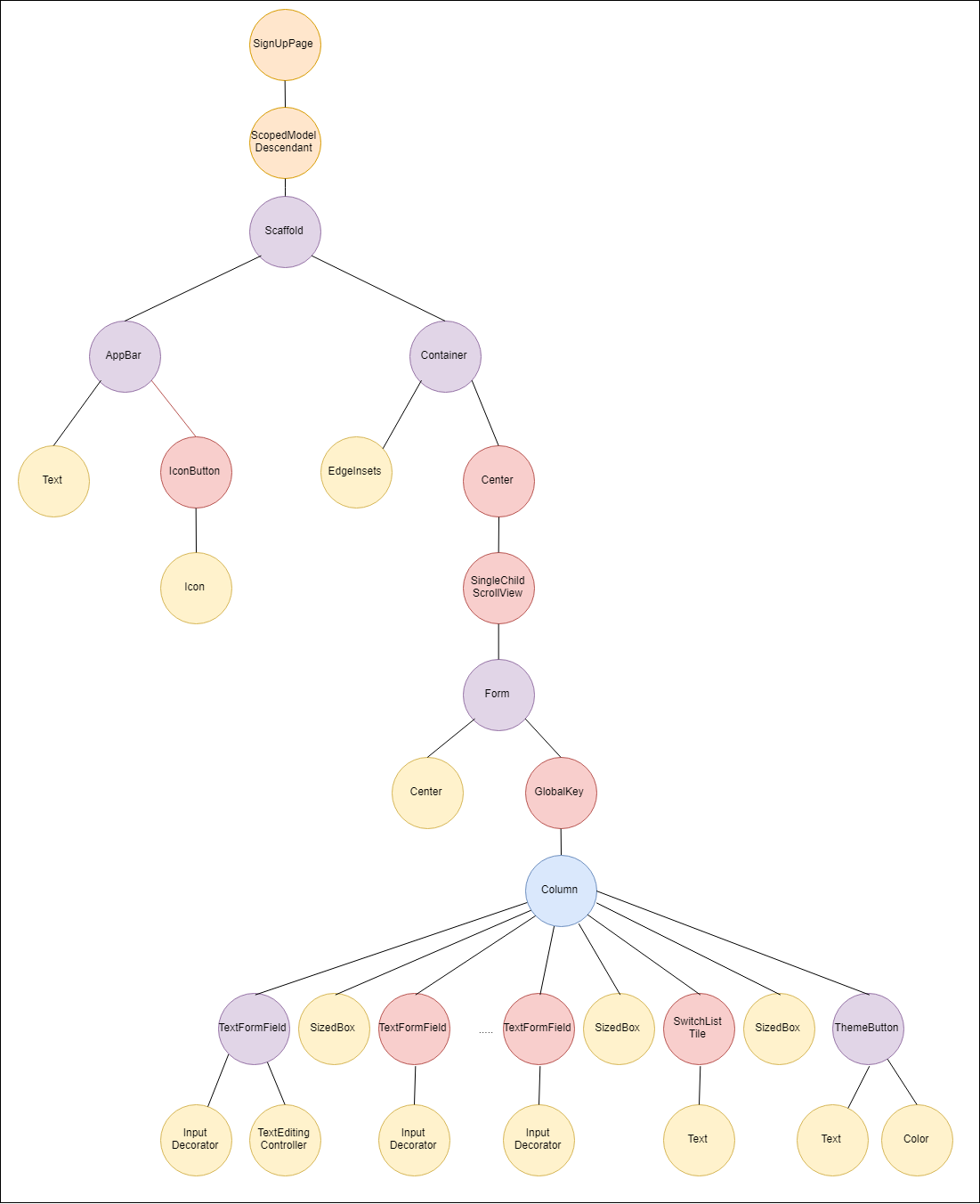
The registration page consist of a form that the user has to complete, each element from the page is part of a single form, in my implementation I do use the verification of the field after the user entered all of the data. In flutter there is a widget called TextFormField, for eache element of a form a new widget of this instance will be added. TextForm field has a validation parameter, and when the user submits the whole form the validator will check if the input is a valid one.

The validation in split the bill is done using regular expression, for this reason I created a class where there are several static functions for different verifications, bellow you can find a code for the verifications:

These verification are also reused in other parts of the project for verification of other forms.

In the previous chapter we talked about RESTful API and how registration works with it, we also mentioned when a user attempt to register there is a response body, and if it was unsuccessful then some error codes are sent as well, in this page some of the most common error messages are handled by creating a pop up window with the error message, so that the user knows whether registration failed or was successful.

As mentioned before the user interface is build from a widget tree, in the bellow image you can find the widget tree for the registration page:

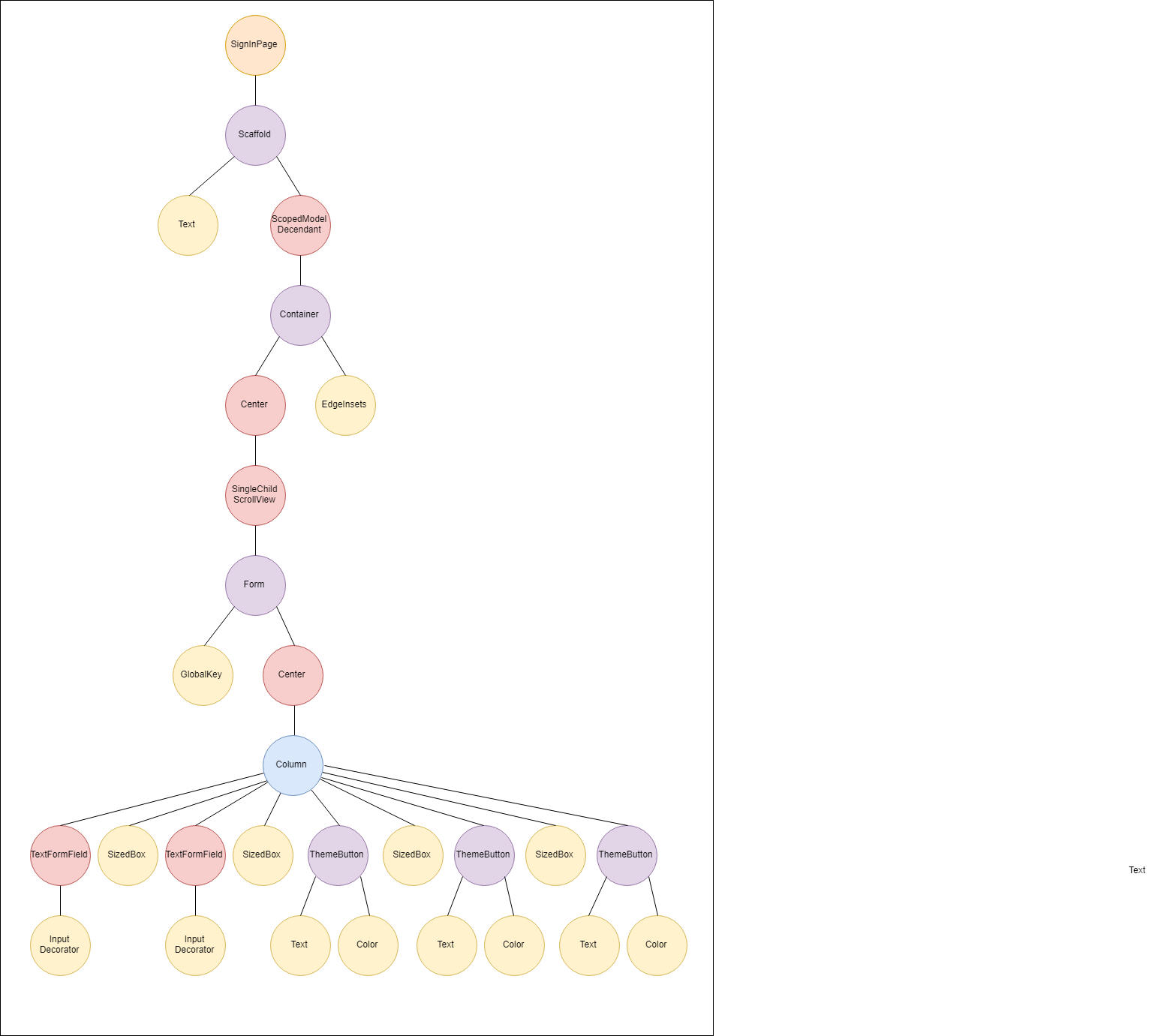


* + 1. Authentication

As mentioned in the previous chapter the authetication is done using RESTful API and using Firebase. Similar to the registration page, there are some error codes that REST can send back if the authentication fails, similarly to reigstration these are handeled using pop up windows, so that the user will know what went wrong.

In the sign in page we also use forms, only this time it is a shorter form, with only two input fields for the email address and for the password, and when the validators are called the functions from the RegularExpression class will be called to check the if the user typed the input correctly.

In the image bellow you can see how the widget tree for the authetication page is build:



5.2.2 Editing User Information

Blah blah