Politecnico di Milano

Design and Implementation of Mobile Applications

Design Document

Bressan G., de Santis S.

DBGAME

Professor Luciano Baresi

Academic Year 2017-2018

Contents

1 Introduction

1.1 Purpose

This is the Design Document for DBgame Android Application. It aims to provide an useful and complete view of the entire application system in particular it is written project managers developers, testers and other people who are interested in it. In this part will be shown and described how DBgame Android Application architectural components are integrated and their interaction with the system. Moreover we will focus also on the RunTime behaviour of the system and we will present specific design patterns.

1.2 Scope

The only way to interact with the DBgame system is only using the application that can be found on the Google Play Store, so there are no Web App to access into it, but is possible to install the application both on Smartphone and Tablets, since it support different screen size. The minimum requirement for install the DBgame application is to have minimum SDK equal to 21 so it can works on:

- API Level 21 Lollipop
- API Level 22 Lollipop
- API Level 23 Marshmallow
- API Level 24 Nougat
- API Level 25 Nougat
- API Level 26 Oreo
- API Level 27 Oreo

Once downloaded the application it will require you to access into the system in two differents ways: using the **Email** (so you have to register you account in a normal way) or you can login using your **Facebook** account. If it is your first time that you open the application you will have to wait a few seconds because the **SQLite database** is empty and you are receiving all the information from the **TheGameDB.com** website using theirs APIs. From the next access into the application you will not have to wait because the main data are already into the SQLite database so you can use normally the application while the data are updated in background. The main concept of the application is to create a collection of video games divided for different console and stay up to date on news about it. For each game you can read all the information about it like for example an overview which explain the game, the publisher of the game, the max number of players, if it is a cooperative game and also is possible to see a trailer of the game through a **Youtube** video. In the section of the game is also possible to evaluate the game with a number of stars (from 1 to 5) and leave the comment. If a game it's interesting for you, you can add it into your Wish List that consisting in to a collection of desidered games and you have the possibility to share the game on the social network like Facebook, simply clicking on the Facebook icon. Once Clicked on Facebook Icon a post that contains all the information about game will be created and posted. In this way you can increase the probability to find someone the will send you the game. In the main page of the application with the resume information of the user, you can also see the news section powered by **GameSpot.com**. Thank to this functionality you will be always updated about new game release date, important update of games and so on. Another interesting functionality available in DBgame is the possibility to find a video game shop nearest to you. So using APIs of Google Maps we have the possibility to show a map with your position and the position of all videogame shop around you. Last but not least is the possibility to have a section of the application dedicated to search specific game through 50.000.

2 Dependencies

In order to provide the best experience and performance to the final user, the Mobile Application make use of different libraries:

External Services

- Mobile application use Google Maps APIs, this let the possibility to load a Map, with a custom style, and track the User's location, up-to-date information about locations around him.
- To perform request of shops, Mobile Application use Place Search APIs, that search for places within a specified area, supplying appropriate keyword.
- Games Details and Platform Details are loaded requesting information through theGamesDB APIs, and parsing resulting XML.
- Youtube APIs that let the application load youtube Videos related to the Game Detail and stream those videos inside the application.
- News are requested to Gamespot , loading its summary available in XML.

Libraries

- Glide v4 for image contents, crop transformations, and caching to achieve best goals in terms of performance and lightweight.
- Retrofit for APIs requests, to theGameDB.net server.
- Raizlabs DBFlow, a blazing fast, powerful, and very simple ORM android database library.
- Firebase Auth that let the application log in the user, and verify authentication
- Facebook Login APIs, to let the user login into the application through its own facebook account, and share contents.
- Butter Knife to find and automatically cast the corresponding view in your layout.
- Firebase Storage and Firebase Database, to store personal images of users, comments, and favourite news, and games.

Graphic Libraries

- Recyclerview animators from wasabeef Github, used for graphical effects inside Recyclerview.
- EasyFlipView, for funny effect.
- Blurry, a library that let the possibility to add Blur effect to a background.

3 Document Structure

- **Introduction:** This section just contains a brief introduction about our Purpose and about the Scope of our Application.
- **Architecture Design:** This section explains our architectural design choices and preferences, it is organized in different chapters:
  - Overview: this section presents the logical and physical layers in our Application
  - High level components and their interaction : In this section we give a global view of the components identified and explains how they interact.
  - Component view : this sections describe each component of the system such database, application layer, mobile application and explain how these component are composed and how communicate between them.
  - Runtime view: in this section we explained using sequence diagram, the main methods used component to interact between them.
  - Selected architectural styles and patterns : there is the description of all the Architectural choices about different system layer.
- **Other design decisions:**
  - User Interface Design: System interface is described by mock-up of mobile android application both smartphone and tablet.

4 Architectural Design

4.1 Overview

In the architectural design section will be discussed all the system components, both physical and logical level starting from a general view of high level components and their distribution on logical layer. In the next section we give you more details and more specification about high level system components explaining their main functionality. Section 2.4 is entirely dedicated to explain how these logical level are divided in physical component, so we will explain the best type of physical tiers used in according to obtain robust structure. In section 2.6 we will focus on the interface between different components of the system. The last part is dedicated to explain and justify the design choices and pattern used in architectural design.

4.2 High Level Components and their Interaction

These are the high level components used in our system:

Database: which is dedicated to storage all DBgame information, so database is the data storage layer, where the system stores permanently all relevant data. Database is one of the most important high level component and for this reason that has to respect the Atomicity, Consistency, Isolation and Durability properties.

Application Layer: is the part of the system dedicated to process all the information coming from external database, required using APIs.

Mobile application: this is the frontend, able to display and dispose information and data using mobile phone interface, and also provide to the user possibility to interact with the system, so this layer contain also a small part of application logic.

Mobile application: it refers to presentation layer, and it is used directly by the customer so it provides mobile phone interface (as described in RASD document).

As we can see in the graph below the system is composed with three logical layers described above. This design permit to have the right compromise in terms of performance and application dimension in term of MB. There are also advantages in performance and security. In performance because each layer has a specific task and functionality to perform, while the in security aspect is due to an assignment of specific security credential to each layer.

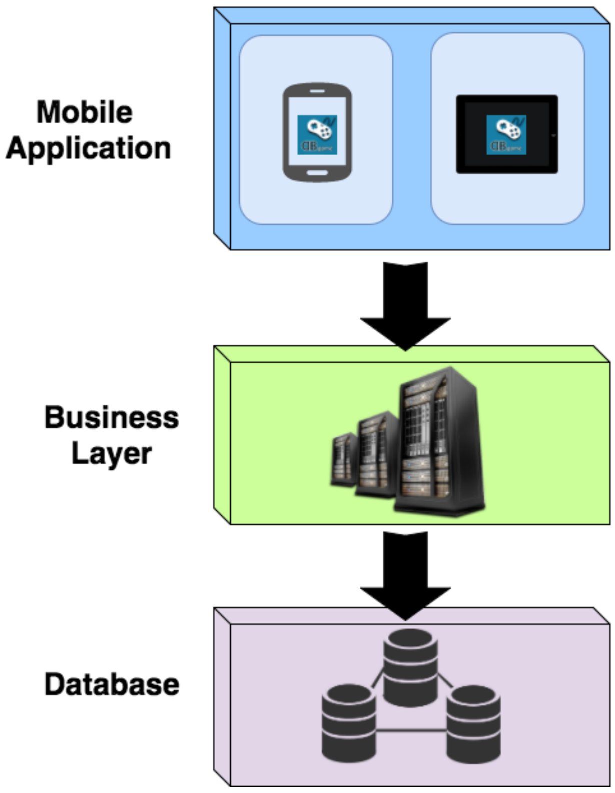


Figure 1: Layer of the system.

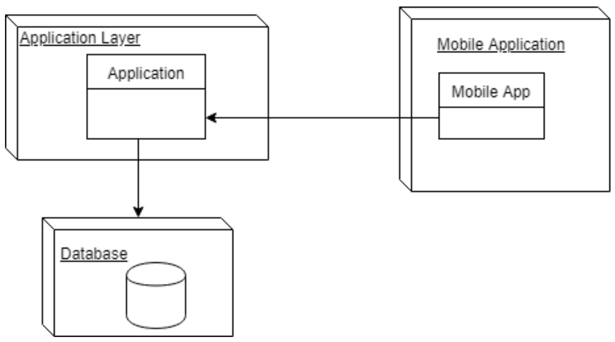


Figure 2: High level components of the system.

4.3 Component view

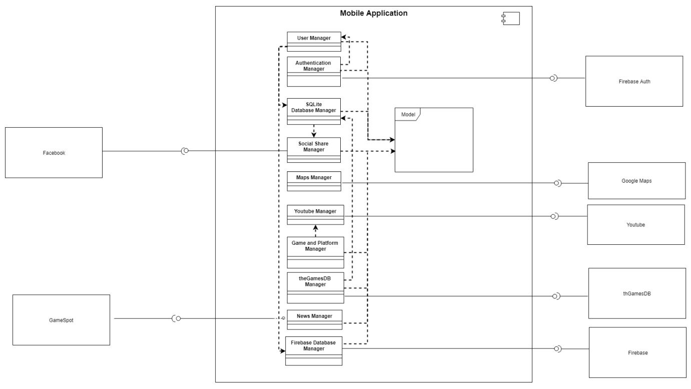


Figure 3: Main Components of the System.

4.3.1 Database

As main database of the application we decided to use Firebase to storage all the information about users profile, images, comments, and their settings. Database cares to communicate only with application layer, as described in high level description. Locally, all the data are stored in an SQLite database, through dbFlow a robust, powerful, and very simple ORM (Object Relational Mapping) android database library. In this way is possible to make the application faster and responsive because doesn't need to require data using APIs each time. used.

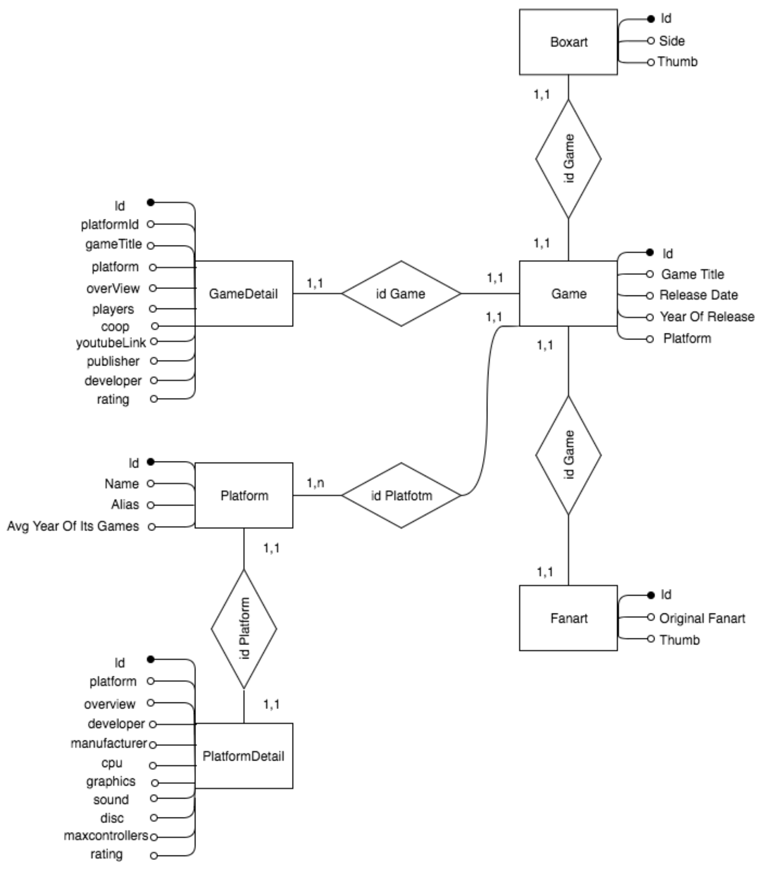


Figure 4: Entity–relationship model rappresentation.

4.3.2 Application Layer

The application layer is split in two different parts: One part is inside Mobile application, it is a lightweight logic, the other one is in the external services with which the Mobile Application interact to.They are logically divided in:

User Manager
Is the part of the application dedicated to manage all the user personal data like username and description and also provide to manage the list of favorite news and wished games. The User Manager interact directly with an external database (in our case Firebase) using Firebase Database Manager which contains proper APIs.

Authentication Manager
This is the part of the Application that authenticate an user, and let the ability to use the application with all the features. This manager let the user register, in case it isn't, or retrieve and change its password if forgot.This part is important because each user have its personalized content, its favourite games and its own read later news list. Moreover its possible to assign to each user the ability to insert a personal description, nickname and image.

SQLite Database Manager
SQLite Database was introduced in the Application to make it more fast and responsive. In fact SQLite is the part of the system dedicated to retrieve the data coming from "not very fast" external database (TheGamesDB.com) and store them following specific feature and schemas. In this case when a user need to open the application, the information are immediately available and he has not to wait the response and the fetch of the data coming from TheGamesDB.com. The important thing is that the SQLite is in any case update and the data will be shown as soon as available.

Social Share Manager
Was introduced in the DBgame application, and it take care to create and publish a post on Facebook.com. In other words, since one functionality of the application is to create a list of wished games, with social share manage that integrate Facebook APIs we can permit to the user to create a post on a specific desidered game in such a way to share it and find someone available to sell a new or used game.

Maps Manager
This part of the Application show and retrieve the view of the Google Map, showing the current position, nearby game Shops, this requires the user location. Tapping over one of the list, user can center the map and preview where it's located. In this perspective, user can see all the list of available Shops and decide to reach the most interesting one.

Youtube Manager

Since the DBgame has the functionality to reproduce a video of a specific game, youtube manger is the part dedicated to obtain the title of the game and return a specific video from youtube thanks to the Youtube APIs. Youtube manager for working properly required that Youtube application is installed into considered device.

Game and Platform Manager
In the application it's important to see all the relevant Platform available, and related games. So Game and Platform Manager has an essential role, from all the retrieved data, there is an algorithm that order games from the most recent to the least one. Also all the platform are ordered from the latest one to the oldest one. All the platform are then grouped by the manufacturer, so each section will include all the platform of the same brand Manufacturer. In this way user can have an immediate look to all the fresh stuff available.

theGamesDB Manager
TheGamesDB manager is the core of our application, in fact thanks to the free APIs offered by TheGamesDB.com we can ask for data and retrieve an XML response which contains all the information about Game, Platform and News about video games.

News Manager
To illustrate all the current updated news , this manager gets from Gamespot newest xml summary of their announcement, with related image. Opening a specific one, it's possible to see its description, and go to relative site. Off course it's possible to pin as a Read Later news.

Firebase Database Manager
Is the part of the application dedicated to interact with Firebase database, and it is used to store all personal information of the user like, name, surname, email, password, description, permit to storage also the favorite news of the users, what are his/her favorite game and also the comment and the rate relative to a specific game. In fact DBgame permit to leave a comment and evaluate a game.

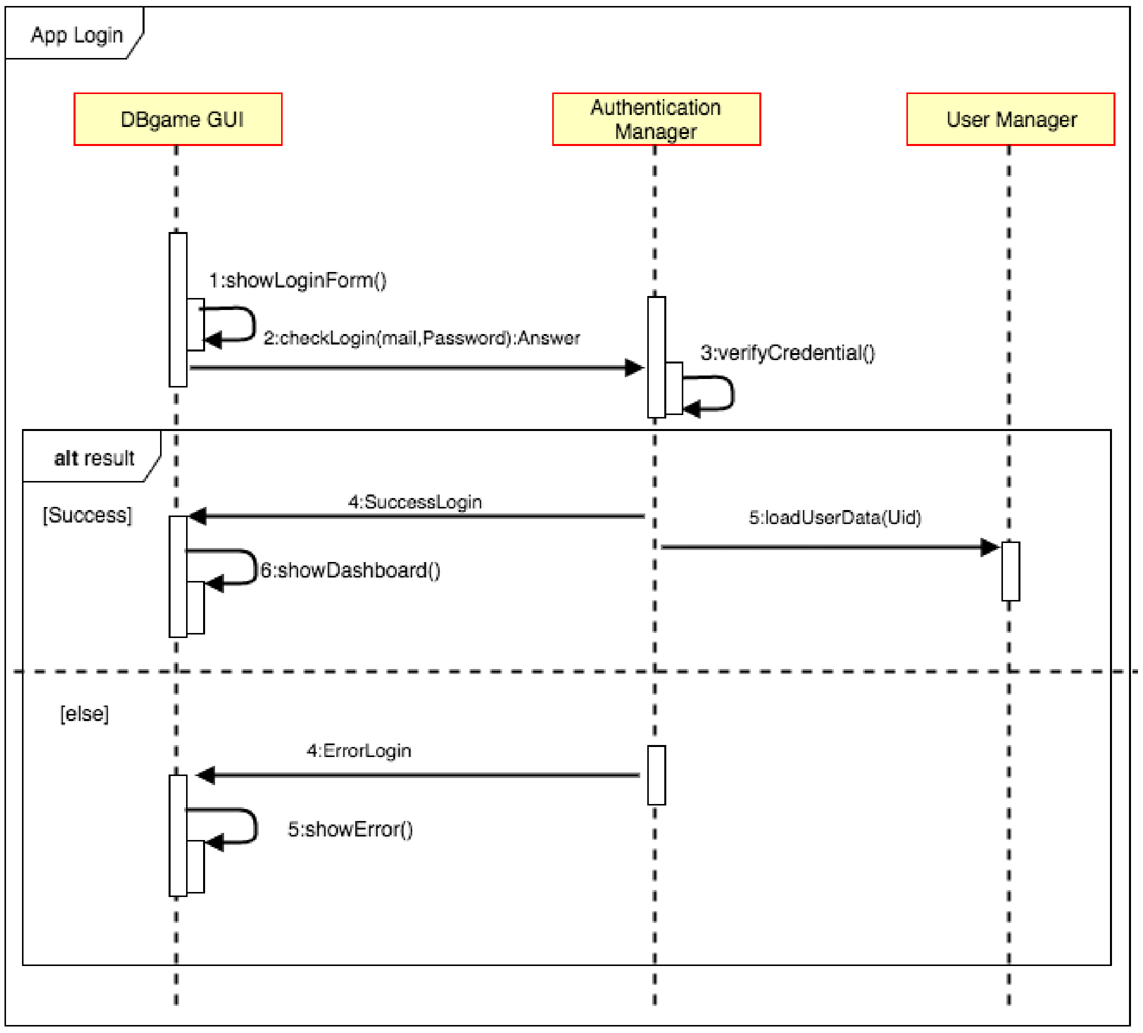
4.3.3 Mobile Application (DBgame)

DBgame Application is the part of the system directly used by customer. DBgame contains the presentation layer so is the part that once taken all the information, display them intuitively in such a way to make easy and comfortable for the customer to use it. This java application let's Android user to have a custom experience and enjoy all the contents with a nice interface. In order to give a direct interaction with user and the service, using personal location. The application let us to improve response, by caching most used data, and images. Moreover graphics element easily adapt to screen size of the specific device, giving to final user a responsive interface.

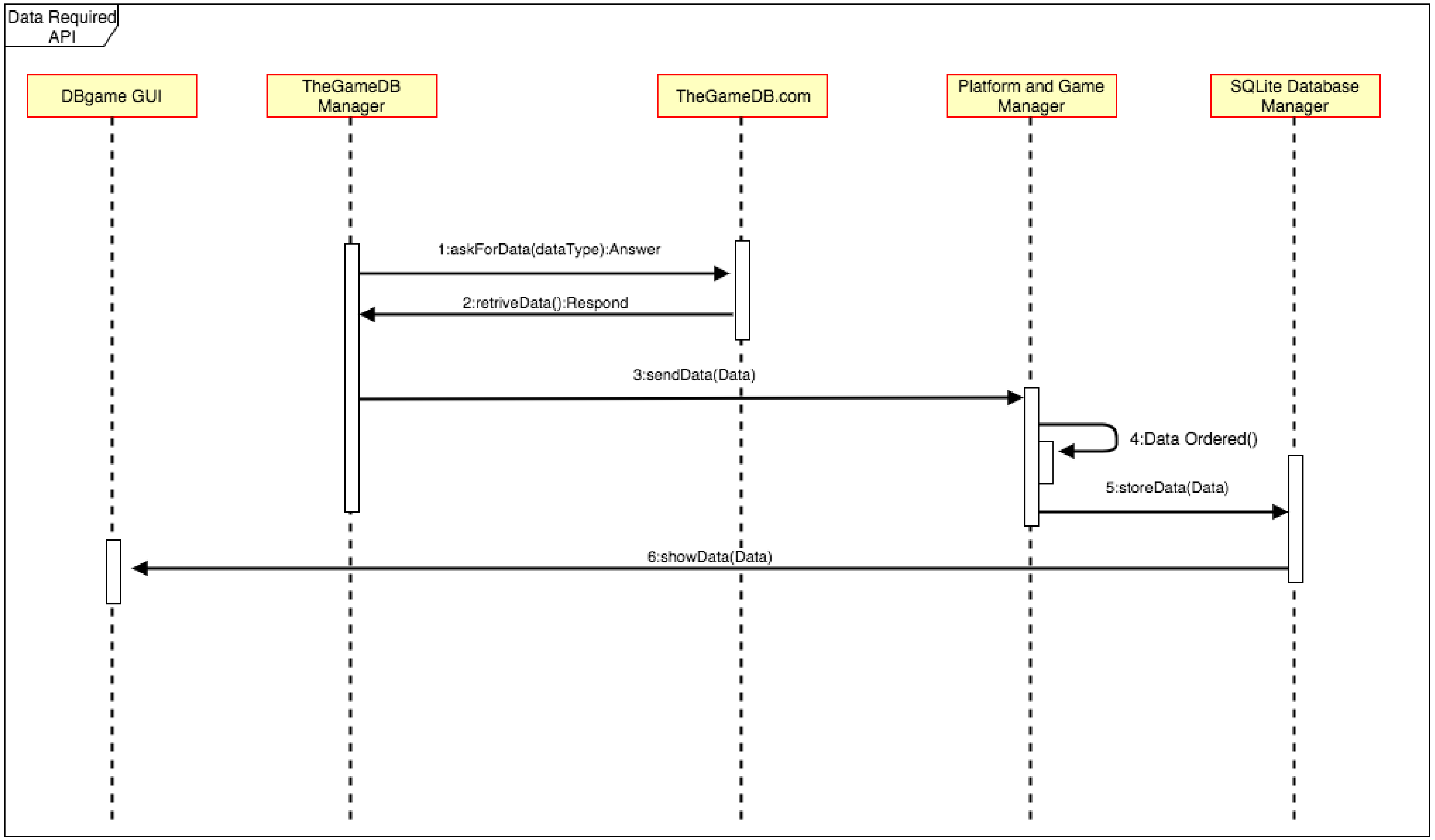
4.4 RunTime View

This section is dedicated to show the interaction between the components previously defined. We will present different use cases in which, using sequence diagrams, the dynamic behaviour of the system will be analyzed.

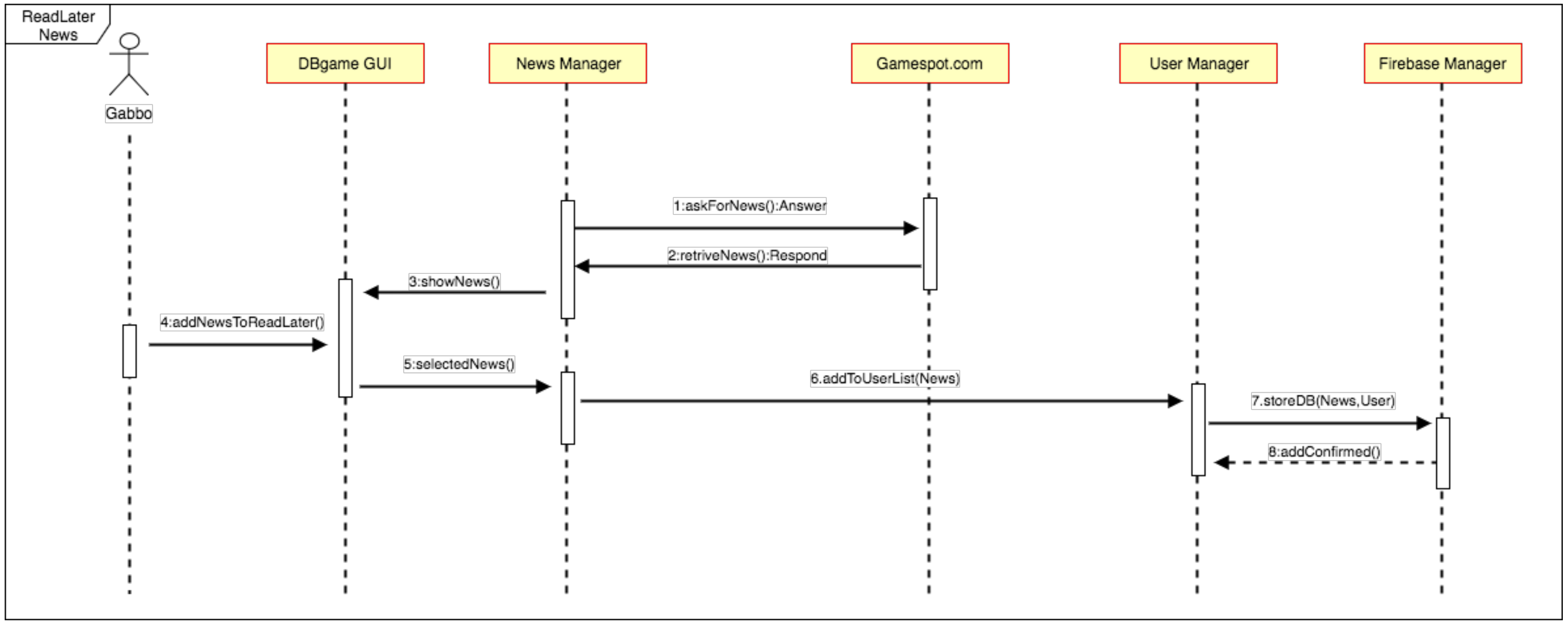
4.4.1 App Login



4.4.2 Interact with TheGamesDB.com



4.4.3 News in Read Later list



5 Overall Architecture

We decided to split our System in 2 tiers:

1. Firebase database(Application Logic and Database)
2. Application Logic (Application Logic and Presentation Layer)

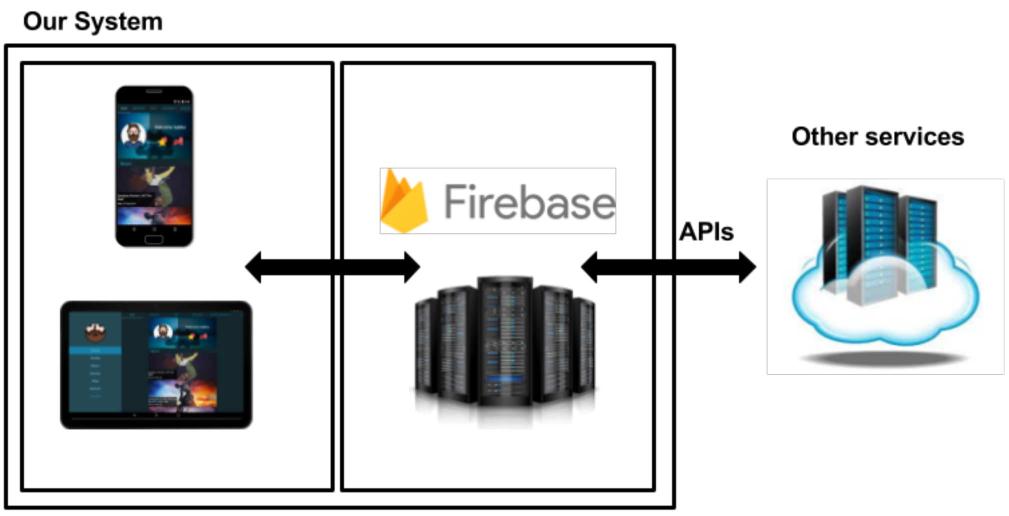


Figure 5: Architectural Overview

Architectural style for DBgame Android Application
The mobile application has been developed for android mobile operating system starting from SDK equal to 21. The style of the application (like the majority part) works taking information from an external database using appropriate APIs, manage the data retrieved and shows the information to the final user. The style of DBgame it's like the one just described in fact it ask and retrieve data from TheGameDB.com, manage the information in such a way use data that we want

and at the end the data are shown. For a better interaction with the user we also use few services, and the image below better explay what they are and what they do.

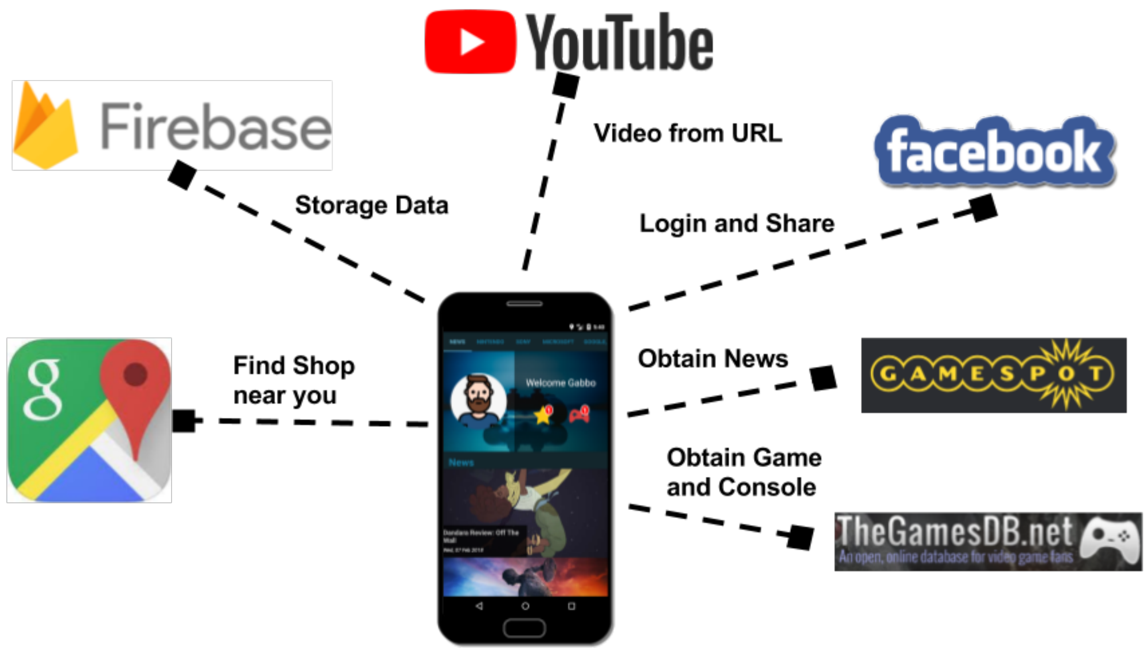


Figure 6: Schema Of API

Architectural style Firebase (Database and Authentication)
Firebase in our application is use for two different purpose. The first consisting in the authentication of the user in to the system, so once registered the application to Firebase and once obtained the KEY is possible to integrate the login and registration form (with mail or Facebook) just inserting lines of code. Firebase also permit to re-obtain the password when it is lost. For better understand how it work there is a scheme below that explain more in detail how Firebase works for Authentication.

Firebase can be also used as database in which storage data. In our case, Firebase it is used to memorize all the information about user like name, surname, mail,

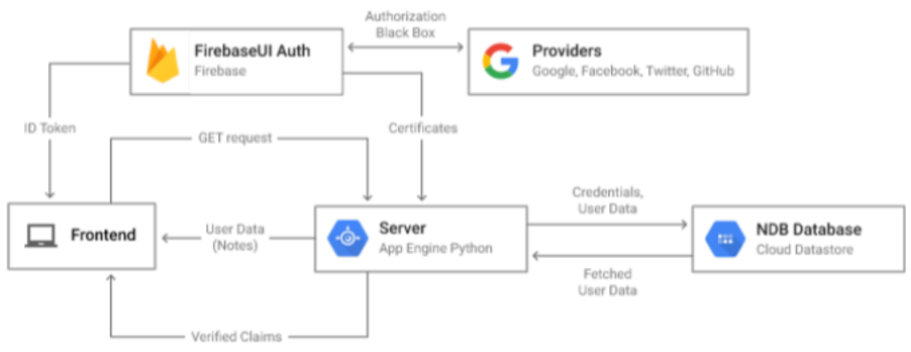


Figure 7: Firebase Authentication

description and password but also information like the preferred game, and the favorite news that he want to read later. Firebase also permit to import and export databases in JSON form. Below is reported a scheme that better explain Firebase used ad database.

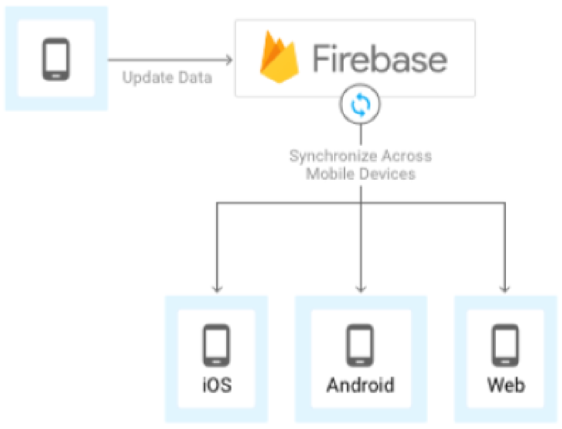


Figure 8: Firebase Database

6 Design Pattern

Adapter
A must, in order to manage ViewPager, that shows all the different games categories, even for RecyclerView, morehover in our application we need to deal with heterogeneous view type inside it.

Observer pattern
This allows the application to define some listeners that in many cases requires to be notified about changes, and even for asynchronous activities.

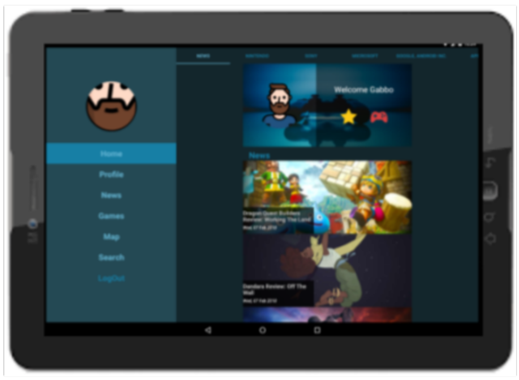
Singleton
Useful to initialize only once, and provide an useful global access to that instance.

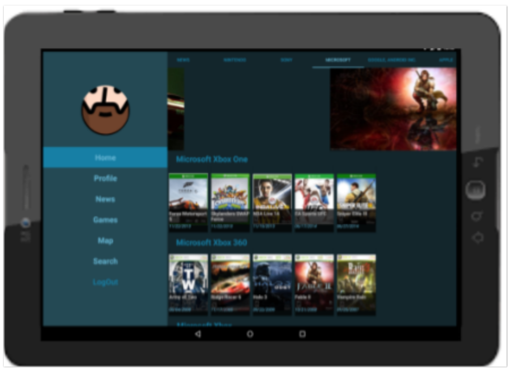
Model-View-Controller
This separation principle has been widely applied in our application, because it is the most common and the most convenient pattern when it comes to deal with complex application, and allows to design software with the concept of separation in mind.

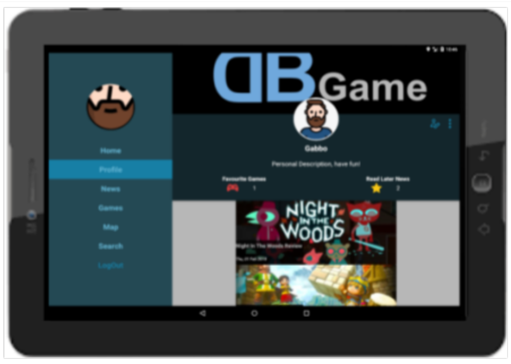
Client-Server
The Client-Server communication model is the most suitable for this application, in fact it allows us to split these two different concepts and to develop a thin client, that must work with low resources. Moreover an Application based on the Client-Server is more practical, maintainable and offers advantages in terms of security.

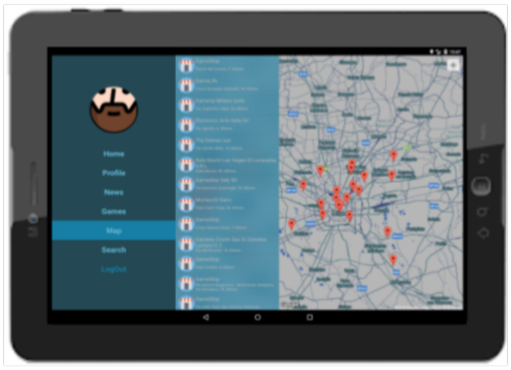
7 User interface design

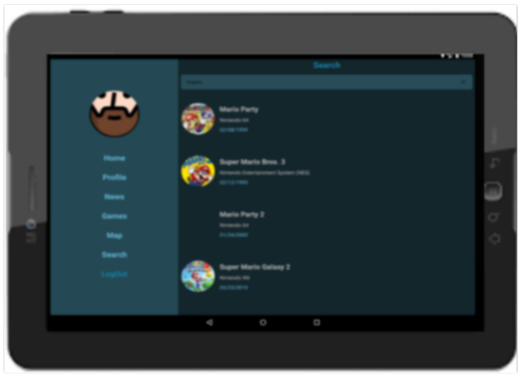
7.1 iPad System

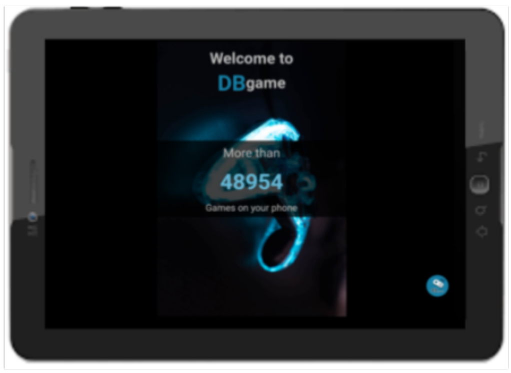




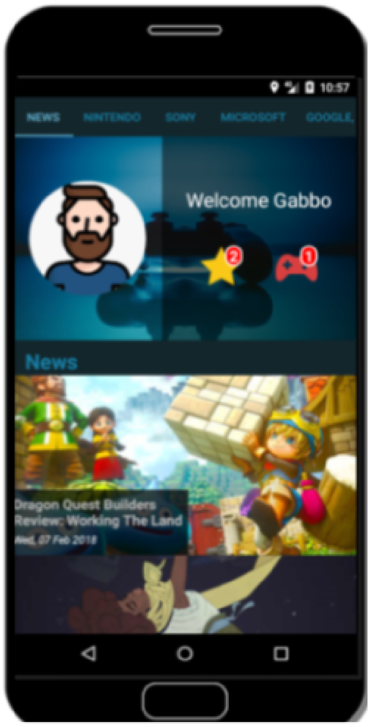


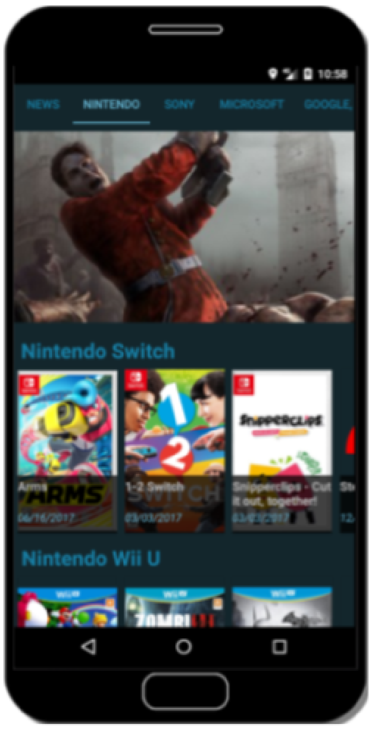


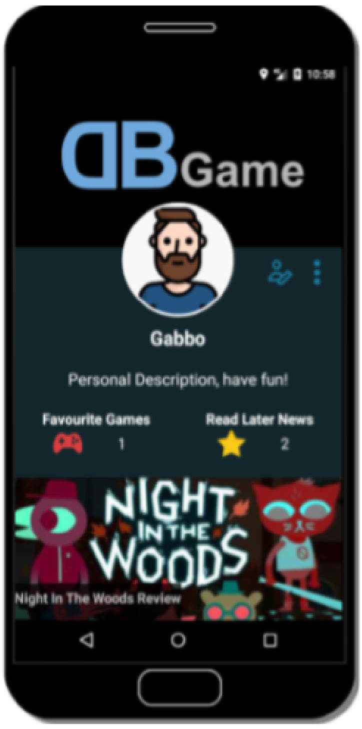


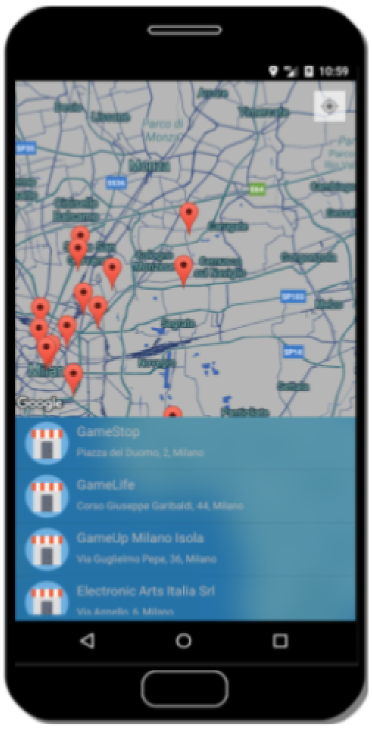


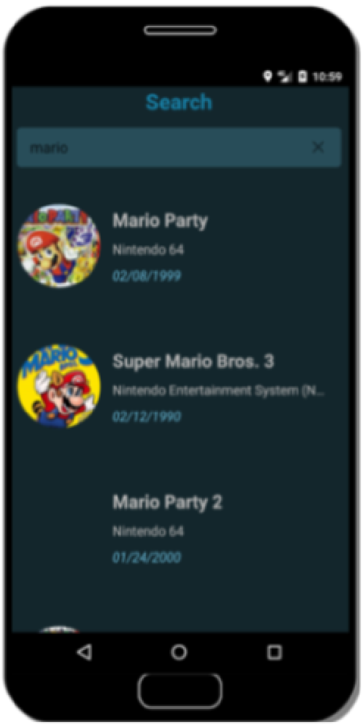
7.2 Mobile App

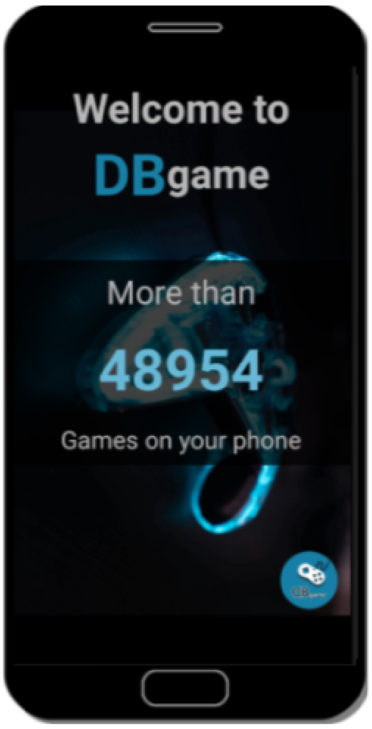







10:59
Monza
Milano
Google
GameStop
Piazza del Duomo, 2, Milano
GameLife
Corso Giuseppe Garibaldi, 44, Milano
GameUp Milano Isola
Via Guglielmo Pepe, 36, Milano
Electronic Arts Italia Srl
Via Amedeo, 4, Milano





8 Functional requirements and components

The table below show how we have combined the component with functional requirements.

- [$G1$] System checks the user's identity through the login procedure.
  - User Manager
  - Authentication Manager
- [$G2$] System allows logged users to see list of Games and Platforms.
  - User Manager
  - Game and Platform Manager
  - theGamesDB Manager
  - SQLite Database Manager
- [$G3$] System allows only logged user with Facebook to create a post and share it .
  - Social Share Manager
  - User Manager
- [$G4$] Have a easy and portable collection of Video Games and Console.
  - Game and Platform Manager
  - theGamesDB Manager
  - SQLite Database Manager
- [$G5$] System allows user to see Youtube related to Games videos.
  - Youtube Manager
  - Game and Platform Manager
- [$G6$] System allow to search a specific game in a collection of 49.000 games.
  - SQLite Database Manager
  - SQLite Database Manager
- [$G7$]System allow logged user to read latest News and their details.
  - News Manager
- [$G8$] System allow logged user to read Game overview and their details.
  - Game and Platform Manager

- [G9]System allow to insert/remove news in a Read Later list.
  - User Manager
  - Firebase Database Manager
- [G10]System allow to insert/remove a game into a Wish List.
  - User Manager
  - Firebase Database Manager
- [G11]System allow logged user to search nearby interesting game Shops and user position.
  - Maps Manager
- [G12]System allow to modify personal information like name, description and change profile image
  - User Manager
  - Firebase Database Manager

9 Used tools

The tools used to create this DD document are:

- Google Drive: used to write simultaneously text parts.
- Bitbucket: to share the code.
- Android Studio: to write code.
- Draw.io (Google extension): for schemas.
- TeXstudio: LaTeX editor.
- Skype: used to do video-conference.
- Firebase: as a servic.e
- Flaticon: web page for free icons
- Photoshop: to edit images.