AL AKHAWAYN UNIVERSITY IN IFRANE
SCHOOL OF SCIENCE & ENGINEERING

Real Estate Agency Mobile Application

CAPSTONE FINAL REPORT

Sawsane Ouchtal
Supervisor: Dr. Nasser Assem

Spring 2015
ACKNOWLEDGEMENTS

Before going any further in the description of the process of achieving the requirements of the capstone design project, I would like to thank two people whose contribution in the realization of the project was very remarkable. First, Dr. Nasser Assem who suggested the idea of the project and provided me with useful information about the field of real estate and the steps involved in such a business. Second, my teammate Ismail Qarchli who was in charge of the web based part of the project, and who was always available, creative and collaborative in making the whole project successful. Last but not least, I would like to thank www.stackoverflow.com users who provided concise answers to the issues I was facing in the implementation phase.
# TABLE OF CONTENTS

0. Abstract ........................................................................................................ iii
1. Introduction .................................................................................................... 1
2. Process and Methodology .............................................................................. 1
3. STEEPLE Analysis .......................................................................................... 2
4. Application Requirements ............................................................................. 4
   4.1. Functional Requirements ....................................................................... 4
   4.2. Non-Functional Requirements .................................................................. 6
   4.3. Domain Requirements ............................................................................ 6
5. Entity Relationship Diagram ......................................................................... 7
   5.1. Description of the Entities ..................................................................... 8
   5.2. Description of the Relationships ............................................................. 10
6. Sequence Diagram ......................................................................................... 13
7. System Architecture ....................................................................................... 15
8. Technology Enablers ..................................................................................... 15
9. Synchronization between the mobile and web applications ....................... 16
10. Implementation Process ................................................................................. 27
11. Implementation Results ............................................................................... 24
12. Future Work ................................................................................................. 30
13. Conclusions ................................................................................................. 31
14. References .................................................................................................. 33
15. Appendix ..................................................................................................... 34
1. Abstract:

The aim of this project is to manage the different operations done in real estate business. Like the web application, the mobile application is named ‘DIORR’, obviously referring to houses that are the main concern of any real estate agency. As stated ineptively above, this capstone project is divided into two main parts, the web application, or the website that has been designed and implemented by my capstone’s teammate Ismail Qarchli, and the android based application which I was in charge of during the last four months. In one word, ‘DIORR’ is meant to help people find their future house or sell their property just in a few clicks. This report summarizes the whole process that I went through during the selection, study, design and implementation of this real estate agency’s management system. It involves different documents and specifications that were discussed with my supervisor and my teammate, contains details and justifications for which I used the incremental model for my application and the tools with which the implementation was done. At the end, the appendix contains screenshots of the running application and ideas about the future work related to this project.
2. Introduction:

‘DIORR’ is an application for a real estate agency allowing on one hand property owners to publish their real estate for sale or rent, and on the other hand people to view these goods and to choose the ones that most suit their needs. My contribution in this capstone project involves designing and implementing the mobile application that allows clients to make real estate transactions from their phones/tablets. The real estate agency will make profit thanks to the transactions between clients, who make most of the transaction steps through the application. The mobile application is meant to make the clients’ interaction with the real estate agency and what it allows as functionalities (that are discussed in later sections of this report) very smooth and possible whenever they wish and wherever they happen to be. This final report contains the summary of the work I did during this semester so as to make this application happen. It involves the analysis, design and implementation phases.

3. Process and Methodology:

In order to produce the final application, we followed a software engineering model which makes the process of the project development easier by dividing it into distinct phases on which the focus will be more effective rather than working on the whole project at a time. The model we chose to use in our project is the incremental development model, because it gives a high level of flexibility regarding the change in requirements of the application. In other words, we decided to base our implementation on the division of the project into small increments which would be coded and tested separately. This model provides the right to change requirements or
add some, and this is exactly what we needed as the requirements were not stable throughout the lifecycle of the project.

The progress of the project went as follows:

- Requirement gathering and specifications analysis
- Technology enablers choice and installation
- Implementation and testing
- Deployment

4. STEEPELE Analysis:

4.1. Social:

This application serves a social issue which is finding an intermediary who is loyal and qualified enough to help clients find a buyer/renter for their properties and to guide them through the legal process. Many property owners waste a lot of time and trust in intermediaries who lack the expertise. Our solution reduces the risks of conflicts and misunderstandings before getting their transactions done.

4.2. Technology:

‘DIORR’ will introduce a real estate agency which is accessible just by a click from its own device. The real estate business that is done traditionally, and so is time consuming, will be automated and taken to a new level of robustness and accuracy. Moroccan property owners will be encouraged to get more in touch with technology in this domain more than before.

4.3. Economic:
On one hand, this application is subject to be effected by the economic situation of Morocco. Any possible inflation or crisis would make a significant impact on the real estate agency. On the other hand, the application would also affect Moroccan economy by speeding up the real estate activity in the country.

4.4. Environment:
This application would have an indirect effect on the environment. In fact, people won’t have to look for intermediaries and real estate agencies in the streets to buy, rent or sell their properties. Our application will make them find solutions for their needs while reducing their mobility, and so the environment pollution.

4.5. Political:
This application depends on the policies and regulations of the real estate business in Morocco. For example, if the government changes the commission that the real estate agency gains from every transaction, then the income of the real estate agency is affected by the political environment.

4.6. Legal:
This application is based on respecting the real estate law in Morocco, that is the percentage that the application will gain from every real estate transaction made through it should not exceed what is allowed by the government.

4.7. Ethical:
This application should preserve the confidentiality of the clients’ personal information. It should be a space where both buyer and seller are satisfied while their profiles are not being used without them being aware about it.
5. Application Requirements:

After studying the feasibility of the project and making sure that its realization is possible within the duration of the spring semester and using the tools and facilities which are available, I had to collect the requirements of the application, and this is a task that was accomplished in collaboration with my teammate Ismail Qarchli, and under the supervision of Dr. Nasser Assem. Below is the list of requirements classified into functional, non-functional and domain requirements.

5.1. Functional Requirements:

5.1.1. Manage Users:
   5.1.1.1. Add user
   5.1.1.2. modify user
   5.1.1.3. delete user
   5.1.1.4. search for user
      By First Name
      By Last Name
      By Date of Registration

2. Manage Announcements:
   2.1. Add announcement
   2.2. modify announcement
   2.3. delete announcement
   2.4. search for announcement
      2.4.1. By Date
      2.4.2. By Type

3. Manage Testimonials:
   3.1. Add Testimonial
3.2. Delete Testimonial
3.3. Modify Testimonial
3.4. Search for testimonial
   3.4.1. By User
   3.4.2. By Date

4. Manage Transactions:
   4.1. Add Transaction
   4.2. Modify Transaction
   4.3. Delete Transaction
   4.4. Search for Transaction
      4.4.1. By Date
      4.4.2. By Registration

5. Manage Payments:
   5.1. Add Payment
   5.2. Modify Payment
   5.3. Delete Payment
   5.4. Search for Payment
      5.4.1. By Date
      5.4.2. By Type
      5.4.3. By Transaction

6. Manage Visitations:
   6.1. Add Visitation
   6.2. Modify Visitation
   6.3. Delete Visitation
   6.4. Search for Visitation
      6.4.1. By Registration
      6.4.2. By Employee
      6.4.3. By Date

7. Manage Employees:
   7.1. Add Employee
7.2. Modify Employee
7.3. Delete Employee
7.4. Search for Employee
   7.4.1. By ID
   7.4.2. By First Name
   7.4.3. By Last Name

5.2. Non-Functional Requirements:

The mobile application must be Android-based, and its data is to be managed using SQLite. Also, the application should be easily accessible to users, secured, scalable and well performing. Those criteria are what would make the application stand up from a simple application to a product that can be delivered and used by real life clients. However, the focus is more on implementing the application using the new tools before considering these enterprise application features.

5.3. Domain Requirements:

The application must be well designed to serve the needs of different users in the real estate business. In other words, it should care about the clients’ satisfaction regarding finding the properties that suit their needs. Also, it must be aware of what is legal and what is not in real estate business, and what the application gets as profit should be calculated according to the Moroccan law.

6. Entity Relationship Diagram:
Specifying the requirements of the application and analyzing them allowed us to get to the second step which is designing the database. The diagram above represents the tables that form the database of our application. As shown, there are 13 table that will embrace all the data needed to make the application alive. We represented the relationships between these tables and enumerated the attributes of every entity. Drawing this diagram required a number of meetings between me and my teammate and our supervisor before reaching the final diagram which is the basis of the implementation of our database.

6.1. Description of the Entities:

As the ERD shows, the database of our project “DIORR” contains the following tables:

User, Account, Announcements, Registration, Property, Property_Description, Price_History, Registration, Visitation, Employee, Transaction, Payment, and Testimonials.

The following part is a description of each entity:

**User**: This table refers to the users using the system of the real estate agency. Each user is defined by his/her CIN, which will be the Primary key of this entity since it should be unique between users. Moreover, a user is defined by a first name, last name, gender, a date of birth, and date of registration.

**Account**: This table will keep track of all accounts in the system. Each user will have an account that will be defined by an Email, which will be its primary key. Therefore, if a user wants to get connected, he/she has to enter his/her Email and password. For security issues, an account will be defined as well by a secret question and its answer in case the user forgets the password. The last attribute is registration_date, which will help to know the date of registration of each user.
**Property**: This table is used to refer to the different users’ properties that are ready to get rented or sold. It is defined by an ID, type, name, city, and a specific address. Each property is going to be linked to a specific user.

**Property Description**: This table specifies the characteristics of the property. It has an id to identify it from other descriptions, the type of the property (apartment, villa, field…), the area of the property, its number of rooms, of façades, garages and floors, and a description for further details about the property.

**Testimonials**: This table is for gathering the testimonials of different users who used the application. Each testimonial will be recognized by an ID, text, and the date of release. Each testimonial will be linked to the user who made it by using his/her CIN.

**Announcement**: This table will keep track of all announcements made by the users of the application. In fact, each user having a property to sell or rent, can create an announcement that will be defined by an announcement_ID, date of the announcement, the type of the announcement, as well as the status and of course the CIN of the user to identify who published the announcement.

**Registration**: This registration table is known by a registration_ID, date, and description. This table is going to be used by the users of the application who are interested in a property posted by another user.

**Visitation**: Each visitation is identified by an ID, date, and type. During each visitation, there will be an employee who’s going to be present as well in order to take remarks.
**Employee:** An employee will be present during the visitation date in order to take remarks about the visitation between the two users. Each employee is identified by a CIN, ID, First and last names, date of birth as well as his/her status.

**Remarks:** A remark is a note identified by a primary key Remark_ID, date, and its content.

**Transaction:** This table is going to be used in order to keep track of all transactions made between users of the application. Each transaction is identified by an ID, an amount, as well as the date of the transaction. This table contains registration ID as a foreign key in order to keep track of the users behind these transactions.

**Payment:** The payment table is identified by an ID, amount, date as well as its type. The table contains as well transaction_ID that represents a foreign key coming from the transaction table. This foreign key is the one that will help us identify the commission to take from every transaction.

6.2. **Description of the relationships between the entities:**

1. Account-User:

   This is a one-to-one relationship since every user should have exactly one account by which he/she will perform the different possible operations on the application.

2. User-Testimonial:

   Each user can write many testimonials where he/she expresses their opinion about the application and to report any bug or unsatisfaction.

3. User-Announcement:
A user can create an announcement to find buyers/renters for his/her property, and create as many announcements as he/she needs. A user can also be registered just to view the announcements of others, which justifies the cardinality of many side; zero or more.

4. User-Registration:

Once an announcement is created and posted on the application, users who are interested in the property being announced can show their interest to the owner by creating a registration into that specific announcement. A user can create as many registrations as he needs, and might create none if he is a property owner rather than a buyer or a renter. This justifies the cardinality of the many side; zero or more.

5. Announcement-Property:

As an announcement serves to allow a property owner to publish this property for sale or rent, every announcement should be related to a property. Also, a user can create as many announcements for the same property depending on his/her needs.

6. Price_History – Property:

A property has many price histories as the price is changeable by the owner depending on how other users respond to his/her announcement. The property should have at least a price history which is the one set at the beginning in case of no change.

7. Property – Property_Description:

Every property should be characterized by one property_description where every detail about it is set by the owner of that property. This is why there is a one to one relationship between these two tables.

8. Registration-Transaction:
A registration might lead to a successful transaction between both parties, or might not lead to any transaction at all if the parties do not agree on its details. This is why there is a one to one relationship between Registration and Transaction entities with a cardinality of zero or more.

9. Registration-Visitation:

Once a user registers into an announcement, a visitation is scheduled to the property so that they can examine more what they are about to buy or rent. A registration leads to at least one visitation, or many if the client is not sure yet.

10. Visitation-Employee:

Each visitation should happen under the supervision of an employee from the real estate agency. But not all employees are supposed to supervise visitations, this is why there is a zero or many in the many side.

11. Visitation-Remarks:

A visitation should be followed by a remark where the employee writes down the description of the meeting between the transaction parties. Every visitation has a remark; which justifies the one to one relationship between the two entities.

12. Transaction-Payment:

When one transaction is successfully completed between a buyer and a seller, the real estate agency should record exactly one payment, which is the commission that corresponds to the type of the transaction (sale, rent...).
7. Sequence Diagram:

We chose to draw a sequence diagram for our project, because it would describe clearly the functionalities of the system and their sequence when performing a specific service by the user. The purpose behind drawing such a diagram is to help us visualize the application as a whole, which will make the implementation easier.

The following diagram shows that users (buyers or sellers or renters) need to register first into the application by creating an account. Then, the property owner creates an announcement to publish his/her property in an attempt to look for buyers. Users can show their interest in that announcement by registering into it. Then, the administrator schedules a visitation depending on the availability of both parties and specifies which employee of the real estate agency will accompany them to the property. Once the visitation is over, the employee has to record any remarks related to the meeting between the buyer and the seller (did they agree on a specific price? do they need another visitation to the property? etc ). If the real estate transaction happened successfully, the employee has to record the transaction and any detail about the legal process between the parties of the transaction. These have to pay the commission in favor of the agency which made their transaction happen. This payment is calculated according to the percentages set by the AMAI, Association Marocaine des Agences Immobilières.[2] In fact, real estate agencies make a profit out of every real estate transaction depending on the type of that transaction; sale, goodwill, rent for a short term or long term. Payments are recorded by the administrator or the employee who supervised the transaction to keep track of the income of the agency. The percentages of the commission are included in Appendix A.
create account() → account created

create account() → Account created

create announcement() → announcement created

register to announcement() → Registration acceptance

Create visitation() → visitation created

Add remarks() → remarks added

Add transaction() → transaction added

Payment Invoice for the seller

Payment Invoice for the buyer

Add payment() → payment added
8. System Architecture:

A system architecture describes the organization of the system in terms of structure and behaviors by representing the different components and the relationship between them. The architecture of this system is composed of a mobile client that allows the application’s users to make use of the different functionalities of the system. This mobile client makes use of the Auto-complete Google Places API, for which an HTTP protocol is needed to allow requests and responses to happen. As this is the only API I used in my mobile application, the next component is the web server and finally a DBMS which role is to manage the raw data of the system.

9. Technology Enablers:

Now that the requirements are analyzed, and the entity relationship and sequence diagrams are drawn, the step of choosing the technology enablers and installing them came. As an Android application developer, I chose to write my application’s code and compile it with the help of Android Studio as it was specifically build for Android development, while Eclipse for example was built to be a general all-purpose IDE that can be used with any platform. Android Studio provided a friendly interface that makes the implementation easier compared to other IDEs. For the database, I did not need to install anything, I used SQLite as it is a software library that implements a self-contained, serverless, zero-configuration,
transactional SQL database engine. SQLite is a good choice for my mobile application as it is fast and light as the name implies.

The tools I used in the development of the mobile application were new to me, even though the language is Java which I am familiar with, coding in Android platform required discovering the different classes and packages that form the basis of any Android application. Therefore, before starting the implementation of the application, I had to go through some tutorials and to practice on a sample application to finally start implementing the final software product.

10. Synchronization between the mobile and web applications:

As ‘DIORR’ is real estate agency that offers its functionalities to its clients either via the web application developed by Ismail Qarchli, or by the mobile application developed by me, it is mandatory that both of us synchronize our databases so as to make data concise and consistent. The following solutions are what was possible to synchronize data of both sides:

- Access MySQL database remotely so that I can read from it or write on it whenever there is an unsynchronized datum. However, as I am using SQLite I tried changing my database and use the very same MySQL database as my teammate. Although he grants me privileges from his side, but the connection fails to be established between us for some reasons related to the firewall of the university.

- Keep using SQLite on the mobile side and sending unsynchronized data between both applications using HTTP requests and in JSON format. This involved the usage of two external libraries: gson and android-async-http. Android app makes HTTP requests to the web application which responds by sending back unsynchronized data in json format.
These data are finally fetched by the mobile app and added to SQLite db. This solution is costly in terms of the heavy code required to add to both PHP functions and Android classes.

11. Implementation Process:

The implementation took almost three months, and involved coding every single day to finally arrive at the final product which will be presented during the capstone defense. The implementation was about designing activities, which are similar to pages in web applications, creating the database inside my project and linking the graphical user interface with the functionalities and the database. Every time I implemented a functionality, I tested it directly on my Android mobile phone so as to get a real life representation of what the application would look like on our clients’ mobile phones. Also, I had a problem with the mobile device emulator that is included in the Android SDK [1], and this is another reason for which I was testing on a real device. I also managed to make the mobile application compatible with phones supporting different versions and with tablets by setting the compatibility features of the application using the code below:

```
<supports-screens
    android:anyDensity="true"
    android:largeScreens="true"
    android:normalScreens="true"
    android:smallScreens="true" />
```

This piece of code is located in the AndroidManifest.xml file. As this file presents essential information about Android applications to the Android system such as permissions, classes, activities, and content providers.
The screenshot above is for the code responsible for granting the application the permission to send short messages (SMS). I had to enable this permission as the application sends messages to users who forgot their passwords. When the user hits the “forgot password” button, they are redirected to an activity where they are asked to answer the secret question that they set the day of the registration into the system. If they answer it correctly, a short message is sent to their phone as a new password that they can change later on as it is randomly generated as shown below.

```java
public void forgotPassword(View view) {
    EditText editText = (EditText) findViewById(R.id.Email); 
    String email = editText.getText().toString();
    String cin = db_user.retrieveCin(email);
    String phone = db_user.retrievePhone(cin);

    Random rand = new Random();
    int randomNum = rand.nextInt((999999 - 10000) + 1) + 10000;
    String activation_code = String.valueOf(randomNum);
    sendSMS(phone, activation_code);
}

private void sendSMS(String phoneNumber, String activation_code) {
    SmsManager sms = SmsManager.getDefault();
    sms.sendTextMessage(phoneNumber, null, "Bienvenue sur Diem, votre nouveau mot de passe : " + activation_code, null, null);
}
```

The screenshot above represents the code that is responsible for granting the application the permission to open network sockets. This permission is essential to make the application use Google places autocomplete API, which completes strings typed by the user in the City text field. This was implemented in the intention of avoiding wrong city names entered by the user.
The screenshot above is for the insertion and retrieval functions of the Account table. `addAccount` uses the insert function provided by the SQLiteDatabase instance “db”. `getAccount()` retrieves accounts that are in the cursor returned by the execution of the query. [3]

The screenshot above is for the display of all the active announcements in the system. It is returns an arraylist customized into strings containing information about every announcement. These announcements are used to fill the ListView of the Announcements activity.
The following screenshot shows the application being executed from a real device rather than the emulator.

Also, from the design point of view, I integrated a sliding menu illustrated in the figure below.
The selected component on the right is the ListView that represents the sliding menu. It is represented outside of the device as it does not show on the screen until the user slides the screen to the right.[5]

This menu shows when the user clicks on the sliding menu button in the upper left side of the screen or by sliding the screen to the right. This involved using some new components compared to the normal activities such as DrawerLayout. Every item in that sliding menu corresponds to a fragment that is generated as an ActionListener handler to that item. Note that a fragment in Android represents a particular interface running within a larger activity, and so they cannot run independently from that activity.
Moreover, I integrated one of the cryptographic hashing algorithms to provide security to the application users. This algorithm is MD5, Message-Digest 5, which is one of the quickest and simplest ways to add security to the users’ credentials. [4]. In fact, when the user choose a password during their registration into the application, their password is being hashed using MD5 algorithm before being inserted into the database. Later on, whenever they log into the application using their credentials, the verification of identity is done by comparing the hashed password and the one that is already hashed and stored in the database.
```java
public class MD5Encryption {
    private static String convertToHex(byte[] data) {
        StringBuffer buf = new StringBuffer();
        for (int i = 0; i < data.length; i++) {
            int halfbyte = (data[i] >> 4) & 0x0F;
            int two_halfs = 0;
            do {
                if ((0 <= halfbyte) && (halfbyte <= 9))
                    buf.append((char) ('0' + halfbyte));
                else
                    buf.append((char) ('A' + (halfbyte - 10)));
                halfbyte = data[i] & 0x0F;
            } while (two_halfs++ < 1);
        }
        return buf.toString();
    }

    public static String MD5(String text) throws NoSuchAlgorithmException, UnsupportedEncodingException {
        MessageDigest md;
        md = MessageDigest.getInstance("MD5");
        byte[] md5Hash = new byte[32];
        md.update(text.getBytes("Base64"), 0, text.length());
        md5Hash = md.digest();
        return convertToHex(md5Hash);
    }
}

public class Account {
    private String email;
    private String password;
    private String secret_question;
    private String secret_answer;
    private boolean is_confirmed;
    private boolean is_active;

    public Account(String email, String password, String secret_question, String secret_answer, boolean is_confirmed, boolean is_active) {
        this.email = email;
        this.password = password;
        this.secret_question = secret_question;
        this.secret_answer = secret_answer;
        this.is_confirmed = is_confirmed;
        this.is_active = is_active;
    }
}

public class DatabaseManager {
    public void addAccount(Account account) {
        ContentValues values = new ContentValues();
        // Add values to the database
        values.put(Account.KEY_EMAIL, account.getEmail());
        values.put(Account.KEY_PASSWORD, account.getPassword());
        values.put(Account.KEY_SECRET_QUESTION, account.getSecretQuestion());
        values.put(Account.KEY_SECRET_ANSWER, account.getSecretAnswer());
        values.put(Account.KEY.IsActive, account.getActive());
        values.put(Account.KEY.CONFIRMATION, account.getConfirmation());
        // Insert the account into the database
        db.insert(TABLE_ACCOUNTS, null, values);
    }
}
```
12. Implementation Results:

This part is meant to describe the final output, that is the perfectly implemented mobile application that meets the functional, non-functional and domain requirements set at the first step of the capstone project. I made some snapshots of the most important activities of the application and described the utility of every one of them.
Figure 1: This is the very first activity that shows up on the screen once the user accesses it.

Figure 2: This activity allows the user to proceed to the functionalities of the application by sliding the left menu, or they can make a search by clicking on the search icon on the upper right corner of the screen.
Figure 3: This menu allows the user to view all the announcements, access to their profile, their DIORR account where they can find their own announcements, payments and visitations, view the testimonials of the application users or have more information about it, mainly the contact information of the developers.

Figure 4: This activity is filled dynamically with all the announcements of the application. Once the user clicks on one of the announcements, they are redirected to the activity in figure 5.
Figure 5: This activity shows the detailed information about a specific activity. If the user is interested in it, they can register by clicking on the button “Abonnez-vous par ici”

Figure 6: This activity contains the contact information of the owners of the application.
Figure 7: This activity is for the logging that is requires to identify the user in case they want to perform some specific actions such as registering into an announcement, accessing their own account, etc.
Figure 8: This activity is for searching properties based on the chosen criteria.

Figure 9: This is the activity that the user gets in case they hit the “Mon Espace DIORR” item in the sliding menu.
Figure 10

Figure 10: This activity contains testimonials that were inserted by the application users. It shows what people think about DIORR.

13. Future Work:

As explained earlier in section 10, we could not implement the synchronization functionality which we deem very important in such a capstone project with two related applications. Being aware of its criticality in this context, we are planning to synchronize our databases so as to make the project realistic and ready to be deployed. Also, I should work on
making the application satisfy the features stated in the non-functional requirements. As I mentioned earlier, I focused more on developing a mobile application using the tools that are new to me rather than making the application secure and scalable.

We would finalize the online payment functionality and create an account in Paypal so that we can make the Paypal code we integrated in both applications work.

We might as well make our application involved in the following steps of the real estate transaction by recoding the sales agreement and any other important step with the notary.

We would empower our search functionality and allow the users to search the properties published for sale or rent which are in the nearby of the user’s location using the GPS function on their phones. Once I manage to accomplish these tasks, I would subscribe to Google Play Store account and pay an amount of 25$ to publish my application on Play Store so that anybody can download it and use it for real.

14. Conclusions:

The development of this capstone project represents one important step of my academic career as I stepped away from what I am capable of doing which mainly developing web applications and moved to an interesting field of development using Android. I got to enrich my capacities of self-learning when it comes to learning about new platforms such as Android in a duration of less than 4 months and working with it in a real life application. Working on this project would widen the areas in which I can work in future internships or real jobs. On the other hand, I developed my teamwork skills as this capstone project was realized with the collaboration of Ismail Qarchli. I trained myself to be more careful with deadlines when it comes to the diaries I had to write and show to my supervisor every week.
Being easily driven by stress, I managed to take control over my uncertainties and work pressure, and I gained some confidence in my computer science capacities as the final result met my expectations.

All in all, I am satisfied with this experience in terms of what I have learnt technically, professionally and also personally, and I emphasize on thanking the people who made this happen; my supervisor and my teammate.
15. References:


Appendix A:

The table below shows the commission that our application would make out every real estate transaction depending on the type of the transaction (sale, rent...)

<table>
<thead>
<tr>
<th>Transaction</th>
<th>Commission</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vente/Achat</td>
<td>2,5% H.T (3% TTC) sur la valeur totale du bien payable par chacune des parties : Vendeur et Acheteur.</td>
</tr>
<tr>
<td>Location</td>
<td>1 mois H.T du montant de la location payable par chacune des parties : Locataire et Bailleur.</td>
</tr>
<tr>
<td>Location Courte Durée</td>
<td></td>
</tr>
<tr>
<td>3 mois</td>
<td>25% H.T du loyer mensuel.</td>
</tr>
<tr>
<td>6 mois</td>
<td>50% H.T du loyer mensuel.</td>
</tr>
<tr>
<td>12 mois</td>
<td>1 mois H.T du loyer mensuel.</td>
</tr>
<tr>
<td>Gestion locative</td>
<td>10% H.T de la valeur mensuelle du loyer</td>
</tr>
<tr>
<td>Fond de commerce</td>
<td>10% H.T de la valeur du fond de commerce</td>
</tr>
</tbody>
</table>