

PROPERTY MANAGEMENT COMPANY

dATABASE PROJECT

Topic M | Friday Morning | 2017

Dr.Le Thi Thanh Sang

**Member:**

**Lê Võ Thanh Duy –ITITIU15030**

**Phạm Sơn Hải –ITITIU15070**

**Lưu Minh Quân -ITITIU15033**

**Mai Lâm Trọng Nguyên –ITIUIU15036**

**Nguyễn Quốc Sơn –ITITU15054**

|  |  |
| --- | --- |
| **Name** | **Tasks** |
| **Phạm Sơn Hải** | **Write application** |
| **Lê Võ Thanh Duy** | **Design ERD, check normalization, design relational schema.** |
| **Lưu Minh Quân** |
| **Mai Lâm Trọng Nguyên** | **Report, Draw ERD, create data, write SQL query, create database table.** |
| **Nguyễn Quốc Sơn** |

Contents

[**I.** **Introduction** 3](#_Toc481357764)

[**II.** **Project progress report** 4](#_Toc481357765)

[**1.** **DESIGN ERD** 4](#_Toc481357766)

\*.[**ERD CONCEPTUAL SCHEMA** 5](#_Toc481357768)

[**2. RELATION SCHEMA** 6](#_Toc481357770)

[**3. CREATE DATABASE** 9](#_Toc481357771)

[**4.** **GUI and JDBC** 10](#_Toc481357772)

[**III. Demo** 11](#_Toc481357773)

[**IV.** **Conclusion** 22](#_Toc481357774)

1. **Introduction**

**According to the topic assigned to us, we have designed and code the appropriate software for the customer to manage its property, employee and tenant. First of all, according to the requirement in the topic our software has entities such as buildings, apartments, person, employee, tenant, manager, technician, skill, next of kin, phone number and so on. And each of the entity is in connection with others. For example, each apartment is inside a building and each employee works in one or several buildings, a tenant stays in one apartment, an apartment can have multiple tenants. A person, which can be a tenant or an employee, has one or many phone numbers. Each technician can have one or several skills. And each tenant can have several next of kin with phone numbers. And each of these information are stored in a database in normal form of 3, which will guarantee the running of the database to be sufficient for handling property management. Not only does our software store information but it also offers several functions for its 3 type of user which are manager, tenant and technician. A manager can view all the information regards to the building, apartment, employee and tenant and agreement. A manager can also update the data of an apartment and agreement as well as update, create and delete the data of tenant and technician.**

****

1. **Project progress report**
2. **DESIGN ERD**

* **The first step to designing a good ERD is a throughout analysis of the requirement.**
* **We have to analyze the entities required in the system, as well as designing the relationship between each entity in an object-oriented style.**
* **The whole group discuss the requirement together. The first versions of the ERD designed has many flaws such as not obeying the normalization rule as well as not having the correct notation for the ERD diagram, excess relationship and entities as well.**
* **The design job was mostly done by Duy and Quan. The two analyze the previous flawed ERD and redesign the current ERD that is used in our project and the normalization checking was also handled by them. The ERD was drawn by Nguyen**

****

# 

### **ERD CONCEPTUAL SCHEMA (Double-click to access diagram)**





# *ERD CONCEPTUAL SCHEMA*

## **Relation Schema**

* **The ERD conversion to relational schema was done by Duy and Quan followed by the theory provided by Miss Sang**
* To convert the ERD to relational schema we try to remove as many redundant relationship in the database as possible.
* For one to one, one to many, many to one relationship we can remove the relationship by adding the primary key of one entity to one of the connected entity. However for a many to many relationship we can’t reduce the relationship.



**Person**(SSN,firtst\_name,last\_name)

**Phonenumber**(Phone number,SSN)

-From Phonenumber.SSN to Person.SSN

**Tenant**(SSN, Bank\_account,ID\_Building, ID\_Apartment)

-From Tenant.SSN to Person.SSN

-From Tenant.ID\_Building to Apartment.ID\_Building

-From Tenant.ID\_Apartment to Apartment.ID\_Apartment

**Next\_of\_kin**(PhoneNumber,SSN)

-From Next\_of\_kin.SSN to Person.SSN

**Work**(SSN, ID\_Building)

-From Work.SSN to Person.SSN

-From Work.ID\_Building to Building.ID\_Building

**Employee**( SSN, Salary)

-From Employee.SSN to Person.SSN

**Manager**(SSN, ID\_BuildingOffice, ID\_ApartmentOffice)

-From Manager.SSN to Person.SSN

-From Manager.ID\_BuildingOffice to Apartment.ID\_Building

-From Manager.ID\_ApartmentOffice to Apartment.ID\_Apartment

**Techinician**(SSN)

-From Techinician.SSN to Emplyee.SSN

**Skill**(SSN, SkillDescription)

-From Skill.SSN to Person.SSN

**Apartment**(ID\_Apartment, ID\_Building, numBedroom, numBathroom)

-From Apartment.ID\_Building to Building.ID\_Building

**Building**(ID\_Building, Name, Adress, numRoom)

**Agreement**(ID\_Agreement, ID\_Building, ID\_Apartment, StartDate, EndDate, SecurityDeposit,MonthlyRent)

-From Agreement.ID\_Building to Apartment. ID\_Building

- From Agreement.ID\_ID\_Apartment to Apartment. ID\_Apartment

**Sign**(SSN, ID\_Agreement)

-From Sign.SSN to Tenant.SSN

-From Sign.ID\_Agreement to Agreement.ID\_Agreement

## **3. CREATE DATABASE**

- **The database was created using SQL script. This task was done by Son and Nguyen.**

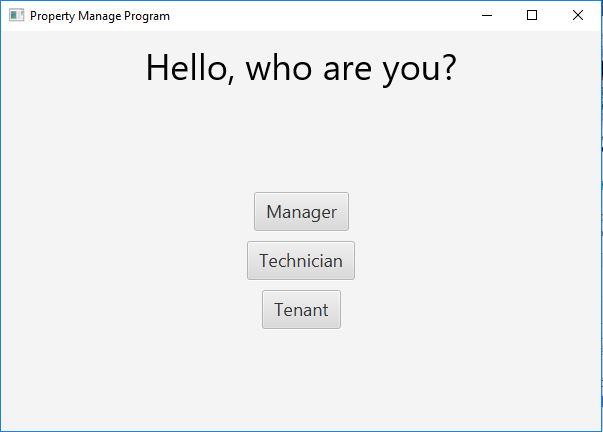
* The database has foreign keys primary key all stated out clearly
* In our database, there are also procedure which contains transaction which will check to see if the database is updated, deleted in a logical manner. For example: if a tenant is deleted some attribute connected to it is also deleted. The procedure handle task such as deleting tenant, technician, inserting tenant, inserting technician, inserting a new agreement.
* And furthermore, for our project, we coded a special program to help auto generate the data set.



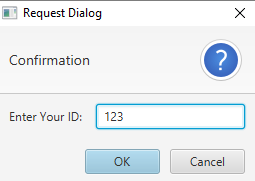
## **GUI and JDBC**

* **GUI and JDBC was mostly done by Hai.**
* In our software, we have 3 user role manager, tenant, and technician which we have separate feature for.
* A tenant can view the information of their roommate, next of kin phone number and so on. They can also edit their own information
* A technician can view the information of people who stays in the room and update some information
* A manager can view all sort of information and made some sort of edition, update and deletion.
* Our software used JavaFX as the main library. We utilized Scene Builder, a really powerful tool which makes it a lot easier.
* Our software GUI has multiple tab, each of which has different use and purposes.
* We have one class to connect to the database and get the data. And from that class we transfer the data to the other class and display them on the GUi.

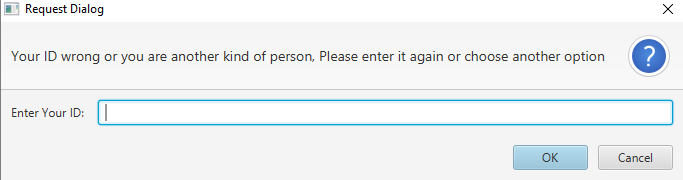
1. Demo



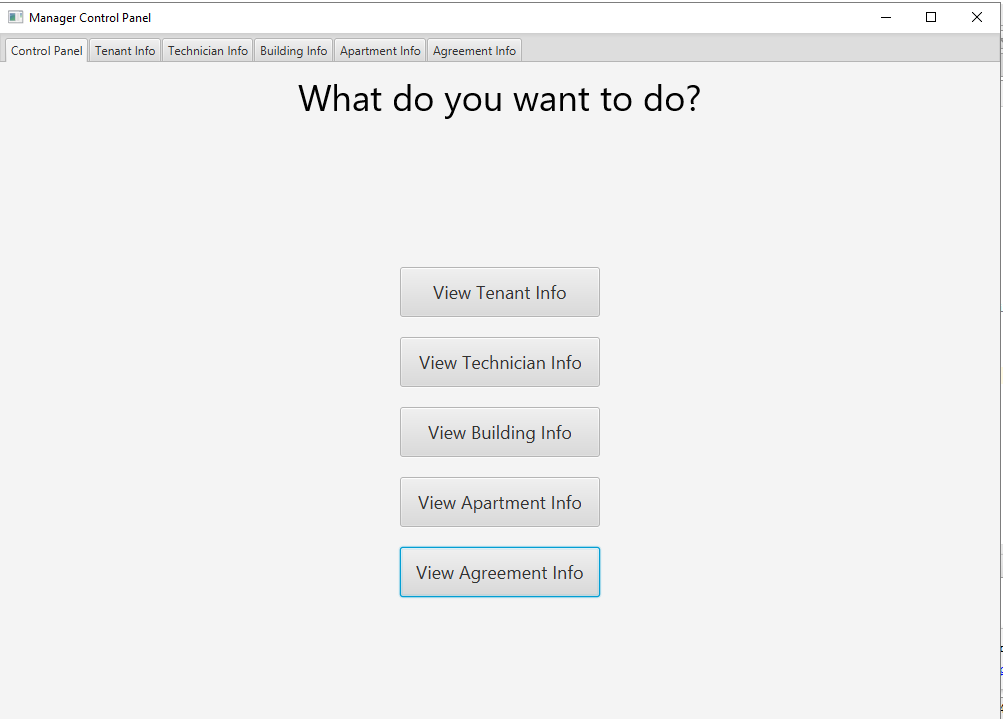
* This is the initial window of our program. You can choose from the 3 roles here which are manager, technician, tenant
* When you click one of the 3 button there is a pop up window which ask you to insert your ID



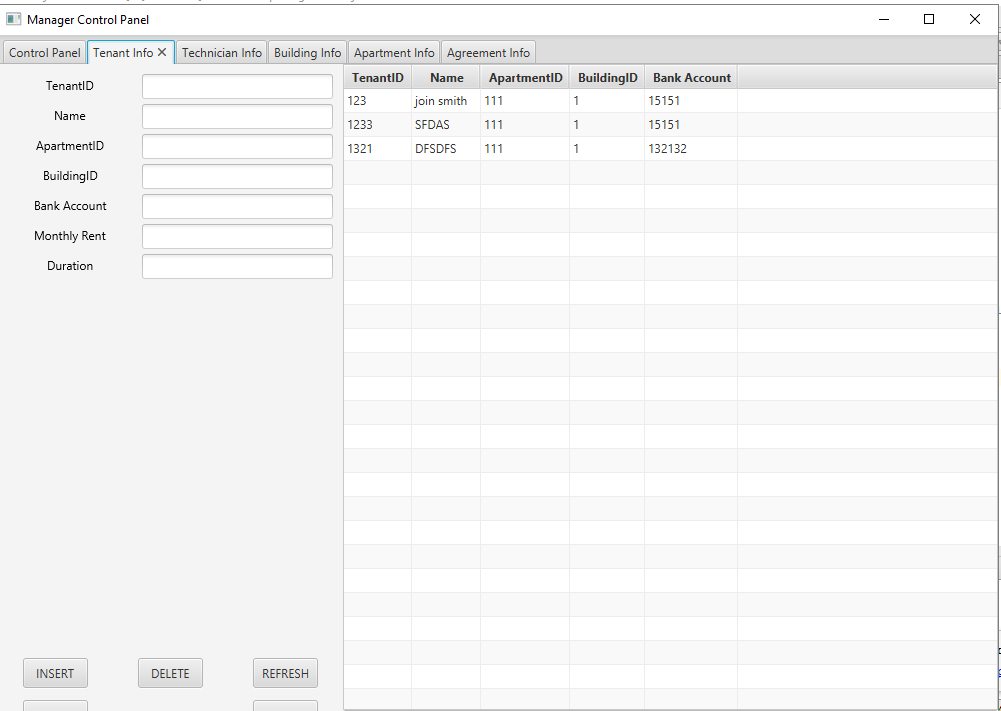
* It will only go to the next window if you input correctly, else a different box will pop up.



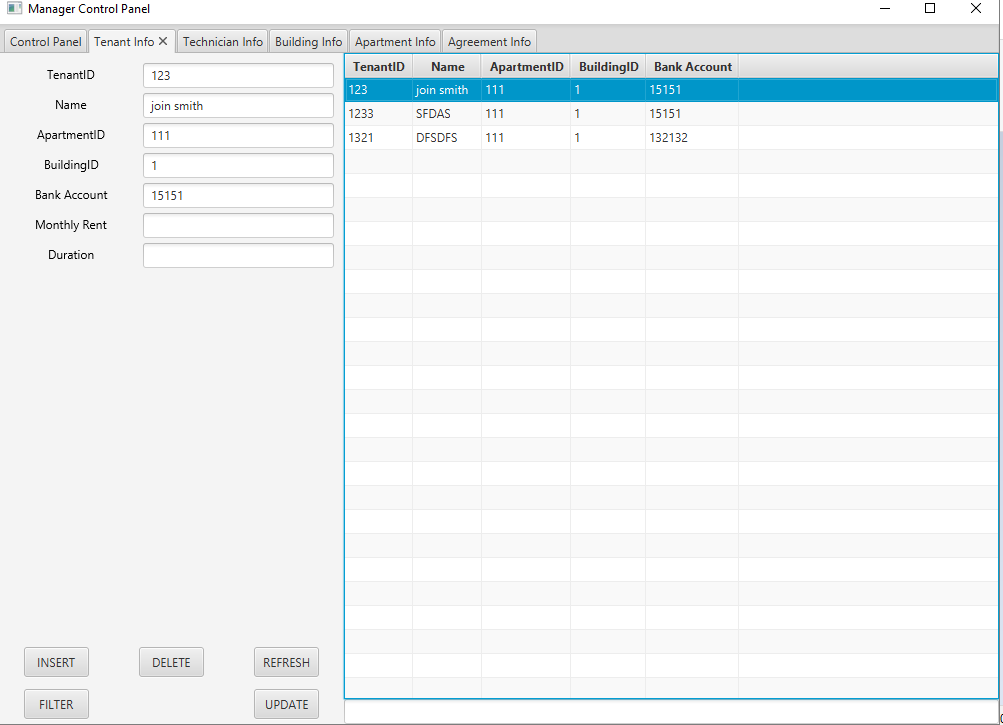
* First is the tenant window.



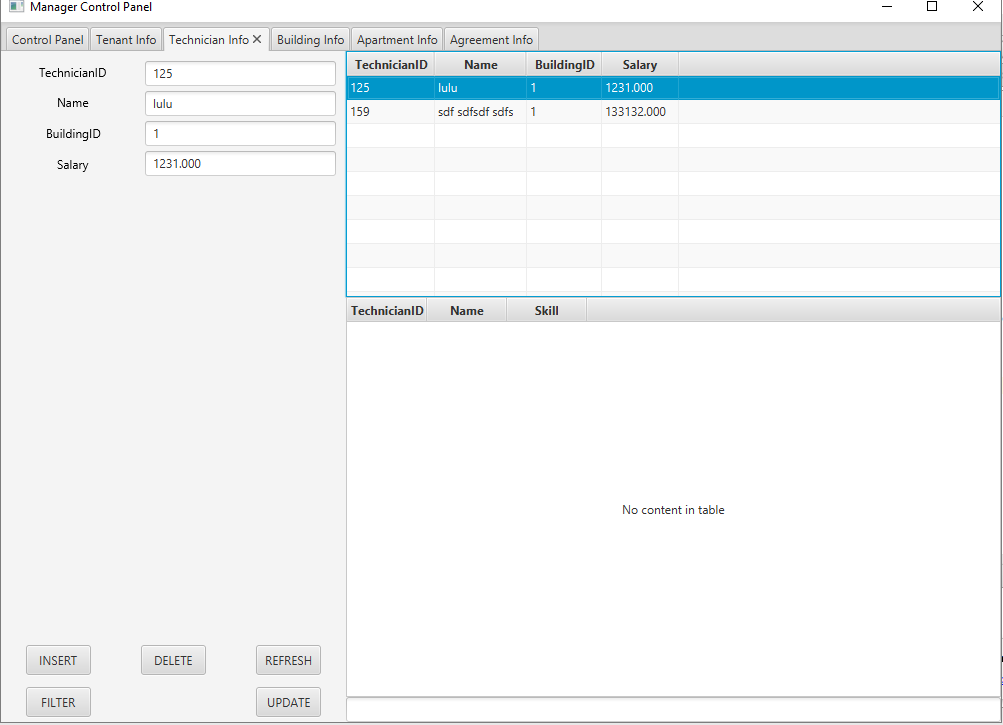
* There are 5 buttons as above. Each time you click it one tab will open.



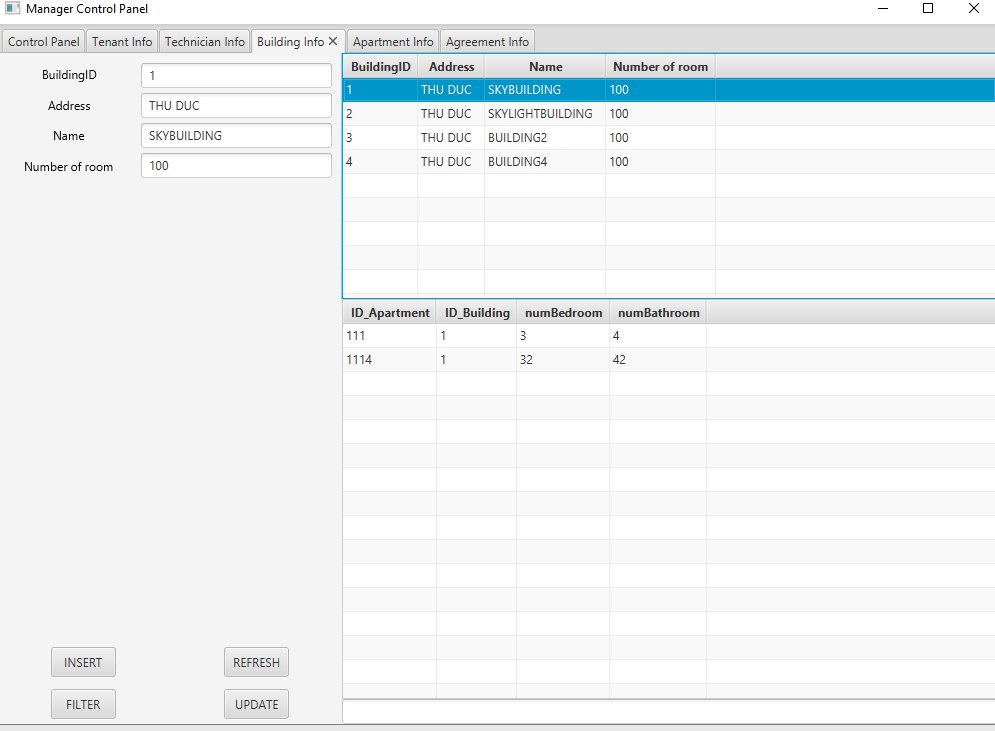
* When you double clicked a row in the table the info will appear on the textbox for you to insert, delete, update and so on. There is also a refresh button which lets you see the database after you have modify it. The filter button is for filter the input.



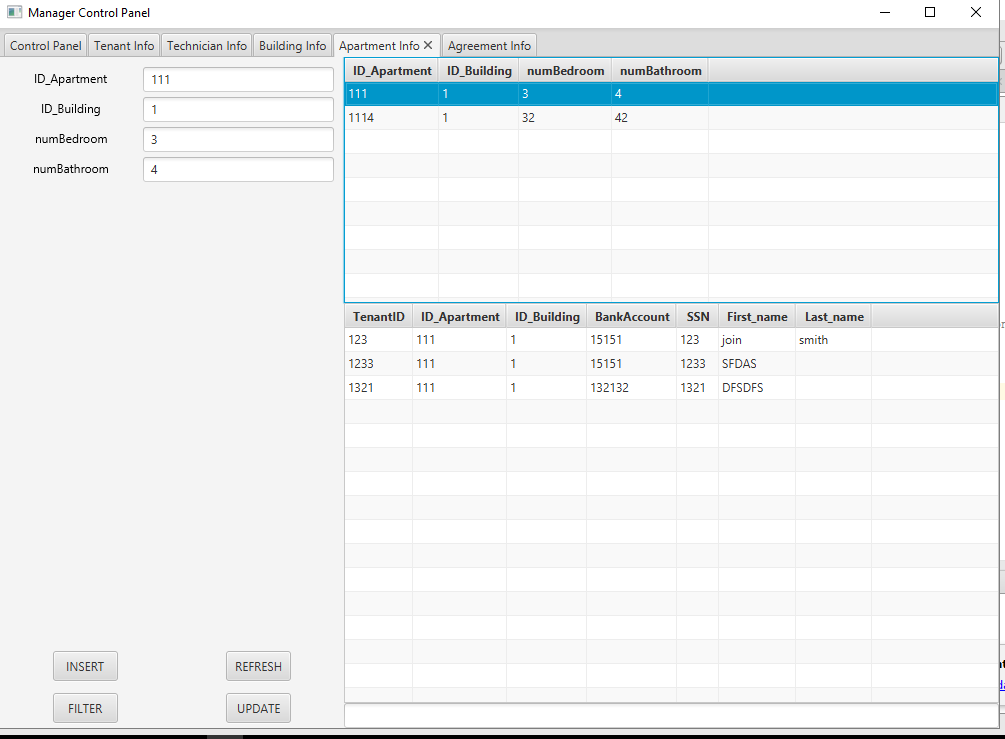
* Next is the technician tab. In the first table you can view their info. When you double click it the second table will appear a list of their skills.



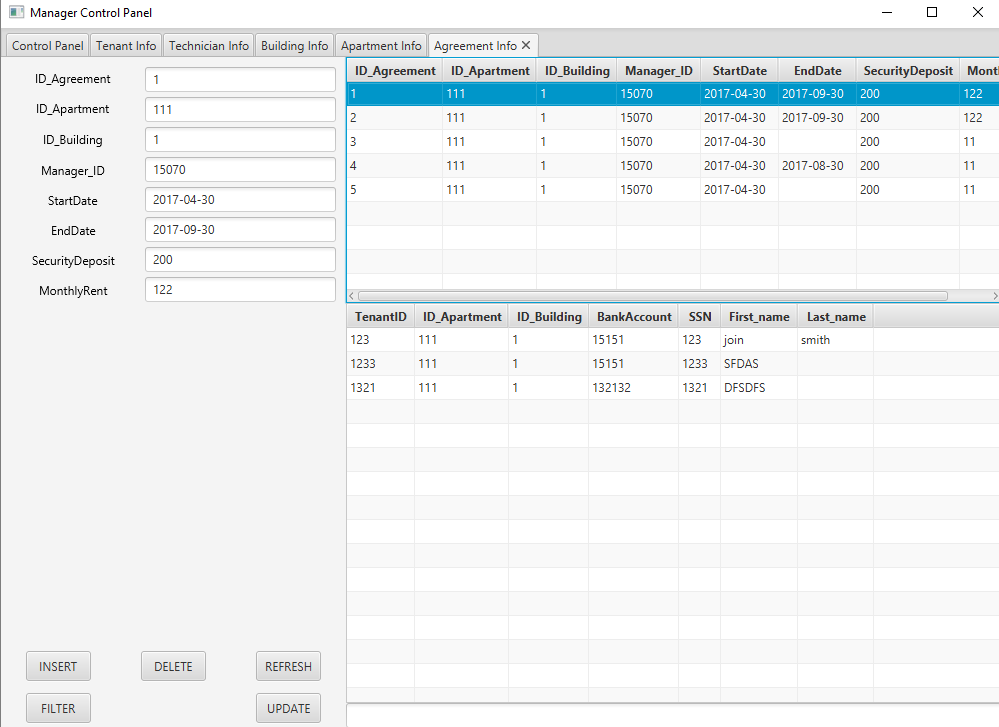
* Next is the building tab. The top table shows the building info, the second one shows the info of the apartment inside the building you double click.



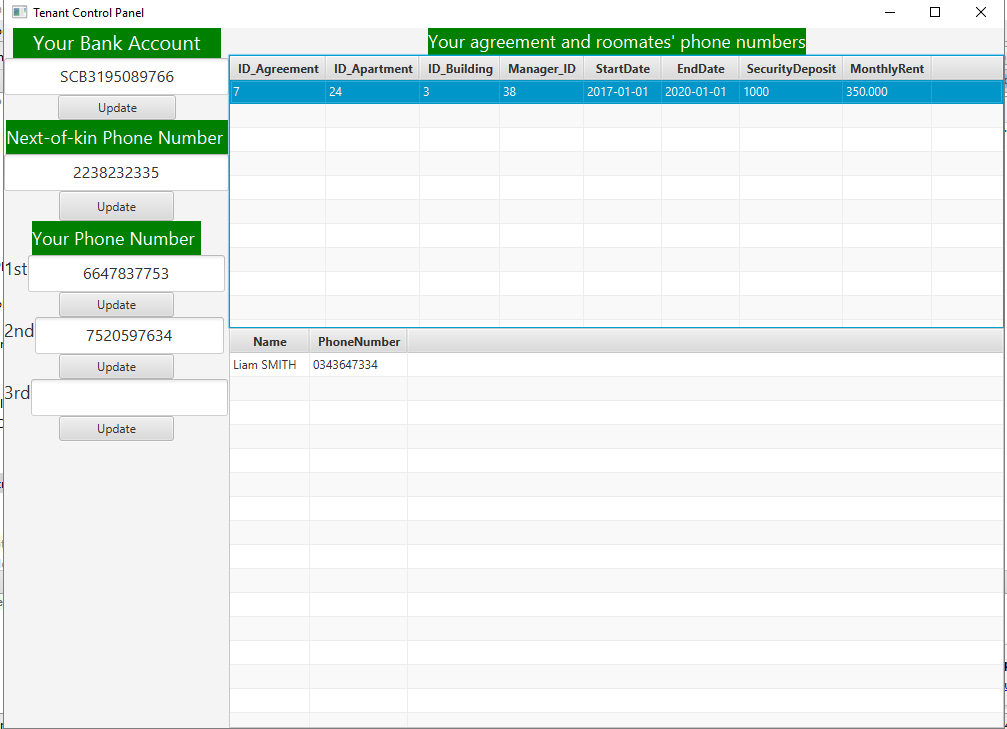
* The apartment tab has 2 table, the first one allows you to see the information of the apartment, the second one allows you to see the tenant in it.



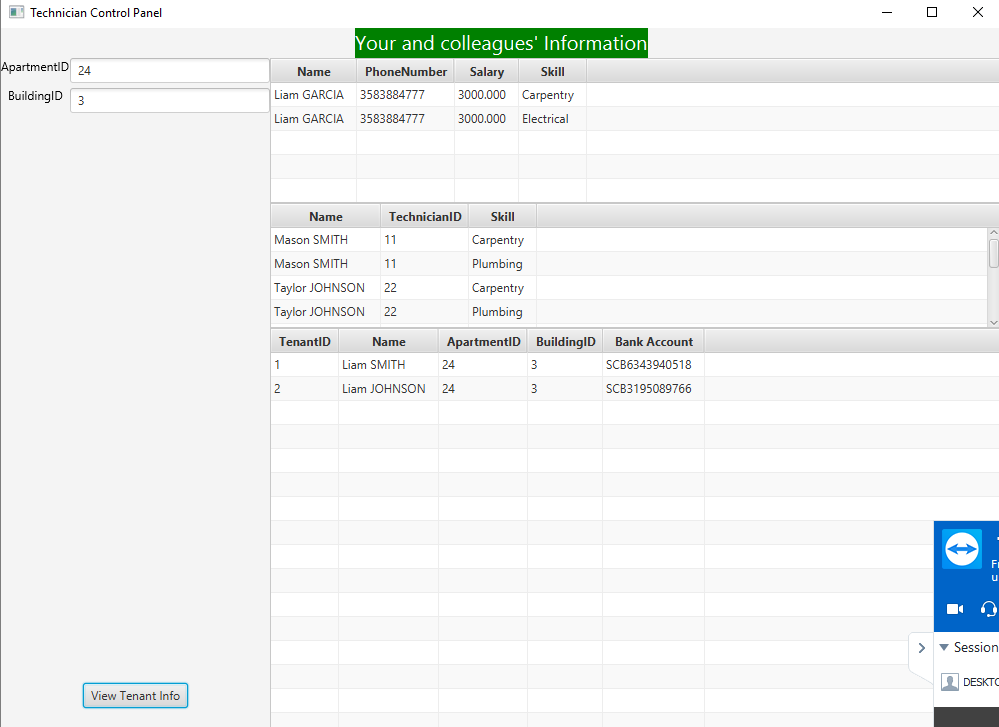
* The agreement tab 2 table as well, when you double click one of the agreement it will show who signed it.



* Next is the tenant user role in the program. A tenant can view their own information and update it. The first table shows the info of the agreement they sign and the second one shows their roommate information for contact.



* Next is the technician user role. For the technician, they can view their own information and their colleagues information. Also they can view the tenant’s name and phone number of a room.



1. **Conclusion**

* **In conclusion, for our project we have utilized many knowledge in OOP, database, GUI, JDBC, and so on. Doing this project have also help us to learn many new skills in programming, designing, etc…**
* **Even though this project is far from perfect, but it is the best of our effort. We all put our best into it.**