GUARDING US

Adrian Rodriguez-Aguilar

Capstone Project Final Architecture & Design

Grand Canyon University

Instructor: Professor Mark Reha

Date: April 24, 2021

**ABSTRACT**

In this project it is to make an application for the guards specifically to those who are working in a private subdivision. This document will show how the project is planned through detailed designs and drawings and some DDL scripts. Also, it will be analyzed in which resources are going to be used and investigated for the web application.

|  |
| --- |
| History and Signoff Sheet |

**Change Record**

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Revision Notes** |
|  | Adrian | Initial draft for review/discussion |
| 4/20/22 | Adrian | Final Revision |
|  |  |  |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

**Integrated Instructor Feedback into Project Documentation**

Yes  No

**TABLE OF CONTENTS**

Design Overview 4

Detailed High-Level Solution Design 5

Detailed Technical Design 6

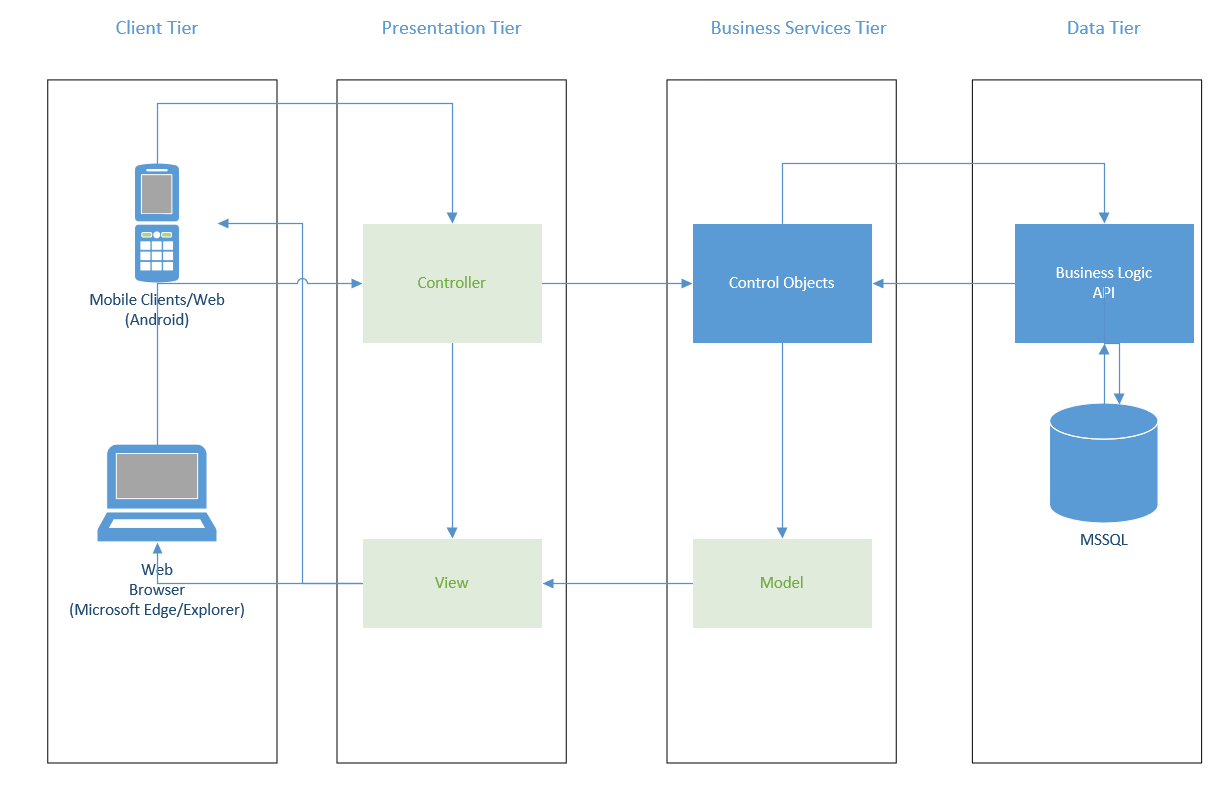
Appendix A – Technical Issue and Risk Log 7

Appendix B – References 8

Appendix C – External Resources 9

Design Introduction

This would be my high-level design for my proposed solution. This application starts using an electronic device: an Android mobile or a computer that has Microsoft Edge / Explorer. After that, they ask the controllers of the web application for a request to be able to fulfill the functions that they are asking for. After knowing the functions, the information towards the control objects and the application API, which is an API that was created especially for this. There is only one database for this project. The API is going to be your job in being able to use the necessary information in the database. After the database gives access to the requested information, the model and the view will have the job of displaying the page or the function on the Android mobile or / and on the Browser (Microsoft Edge / Explorer).



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Deliverable Acceptance Log | | | | | |
| ID | Deliverable Description | Comments | Evaluator (internal or external as applicable) | Status | Date of Decision |
| 1 | Data Dictionary | The file is called “CST-451 Data Dictionary” |  |  |  |
| 2 | API Design | The file is on the Detailed Technical Design section |  |  |  |

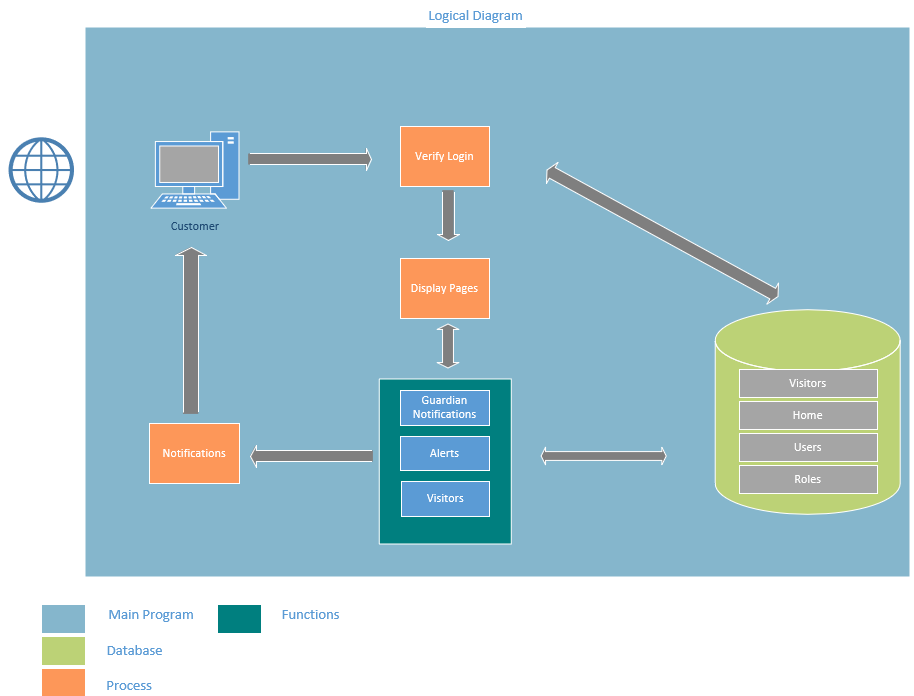
Detailed High-Level Solution Design

1. List of Frameworks
   1. .NET framework
   2. Xamarin

|  |  |  |
| --- | --- | --- |
| Proof of Concepts | |  |
| **Description** | **Rationale** | **Results** |
| 1.Write a function that can be able to submit pictures for the car plate | It would be a quick way to write the information about the visitor | Still working on it |
| 2 – Be able to display an image on the interface | This will help me with the visitor’s information so the page can display the visitor’s car plate | Still working on it |
| 3 – Get the correct connections on the database of the project | The data transfer would be very effective | Success |
| 4 – Research about coding for an Android phone | To be prepared to work on my application in the mobile perspective | Still working on it |
| 5 –Be able to display the data of the database in the correct format | It will help especially for the user to have information that is easy to read | Working on it |

|  |
| --- |
| Hardware and Software Technologies |
| 1 –Visual Studio (C#/HTML/CSS) |
| 2 – Microsoft IIS (ASP.NET/MSSQL) |
| 3 – Android Phone |
| 4 – Desktop/Laptop (Browser: Microsoft Edge/Explorer) |
| 5 -Xamarin |

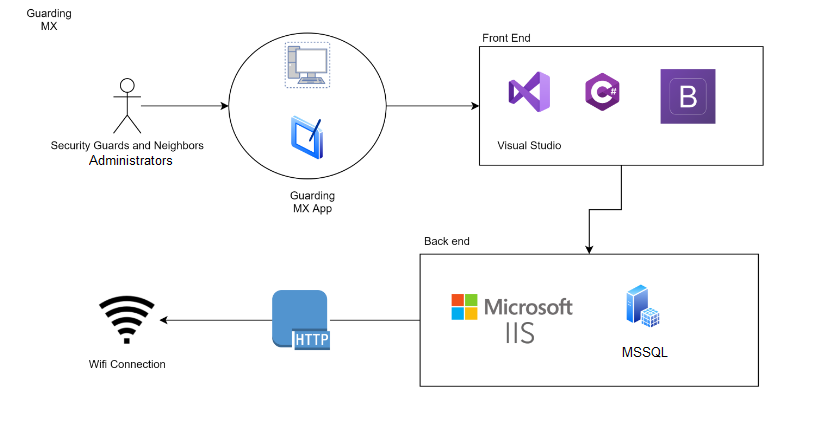
**Logical Solution Design:**

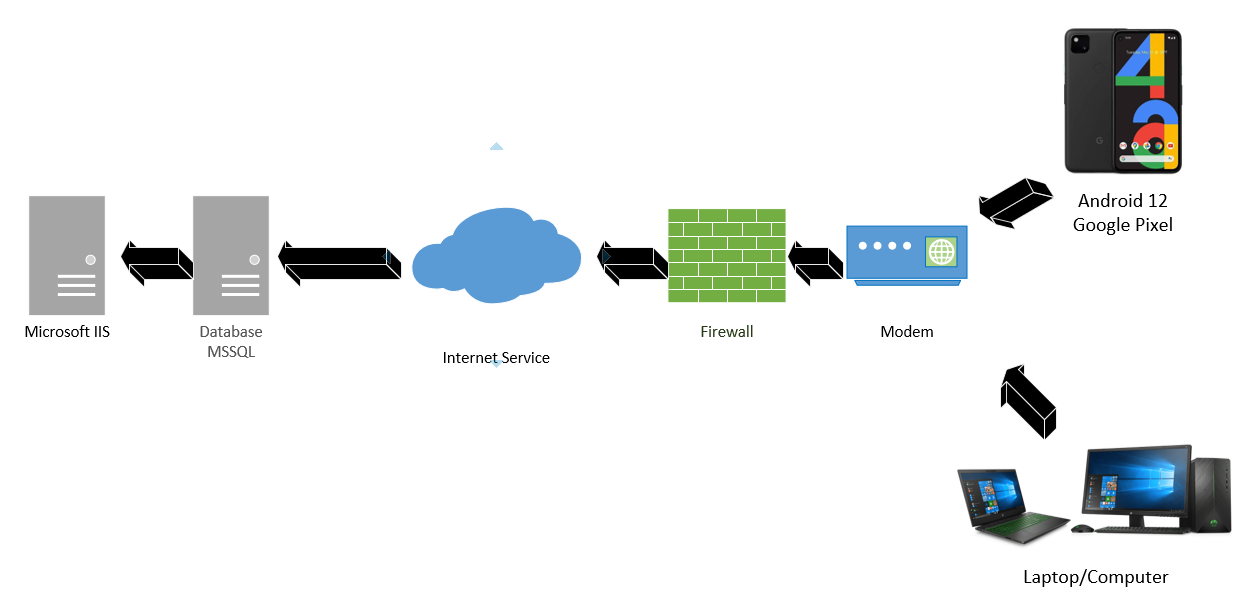


This diagram represents the logical solution design of the web application. Mainly the internet and an electronic device are required to start the application. Then you login with your account and you would get specific webpages displayed depending on what role you have. The way in which the login is verified is thanks to a database that contains 4 tables: visitors, home, users, and roles. Depending on what function you want to use, it will appear on your device in the function that the "customer" selected.

**Physical Solution Design:**

Physical Diagram





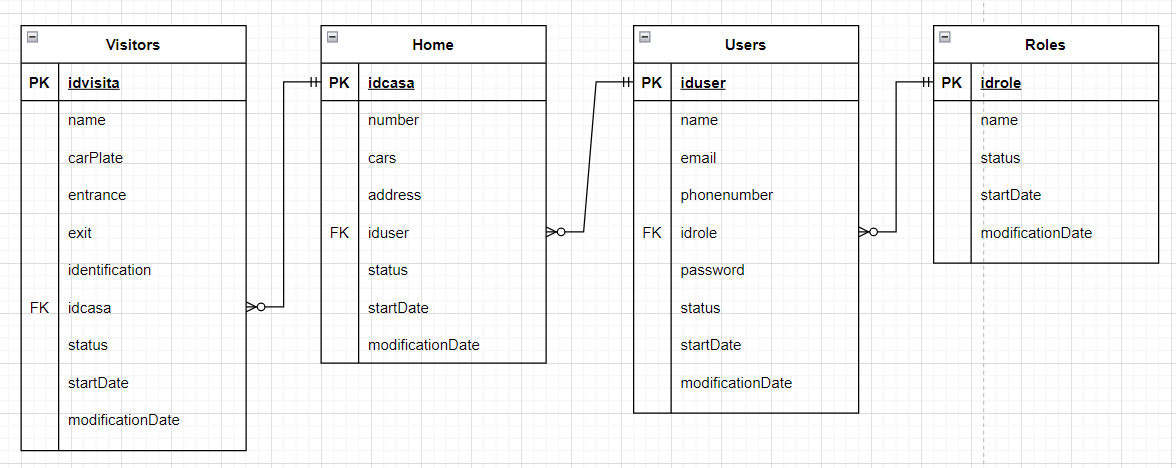
The following drawing is about two physical solution designs of the GUARDING US application. To be able to access the page, you simply need to have an electronic device, either a cell phone with the most recent version of Android, or a computer that has Microsoft Edge / Explorer as a browser. After that, a modem is used to connect to the internet and finally access the databases and Microsoft IIS.

Detailed Technical Design

**General Technical Approach:**

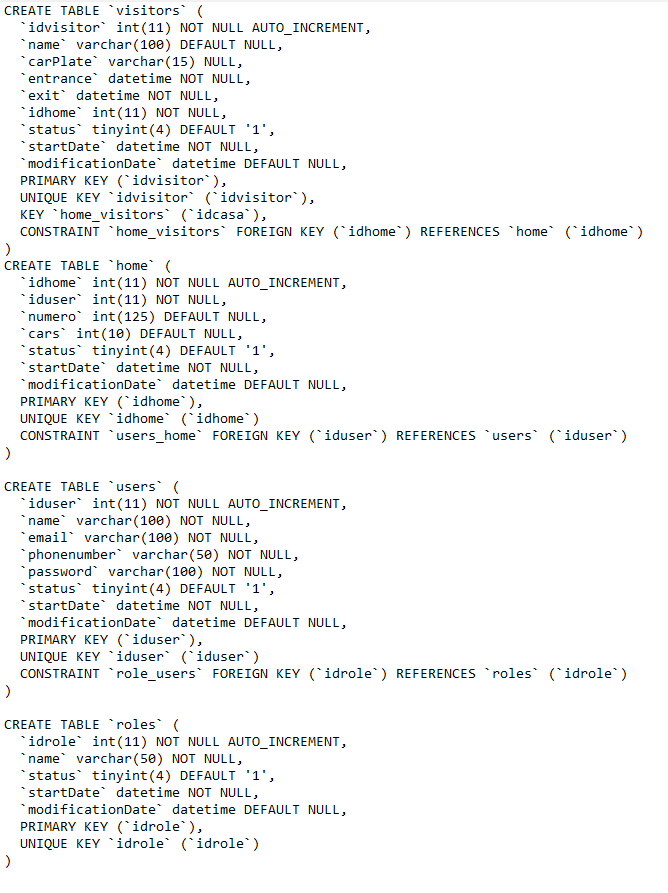
In this project, hardware devices are not required so that the web application can fulfill its functions, but it is important in how you are going to place the fields of the page design. In this application a database is used with which it has 4 tables inside. All the tables include 3 fields with which it is used to see the information of how much the table was created, when it was modified, and the status of the table is active or not. The application and the database are going to be created through Visual Studio and it's going to be displayed with the Microsoft IIS Server. The users who will be able to use this application are those who live in the subdivision, the security guards and the administrator.

**Database ER Diagram:**



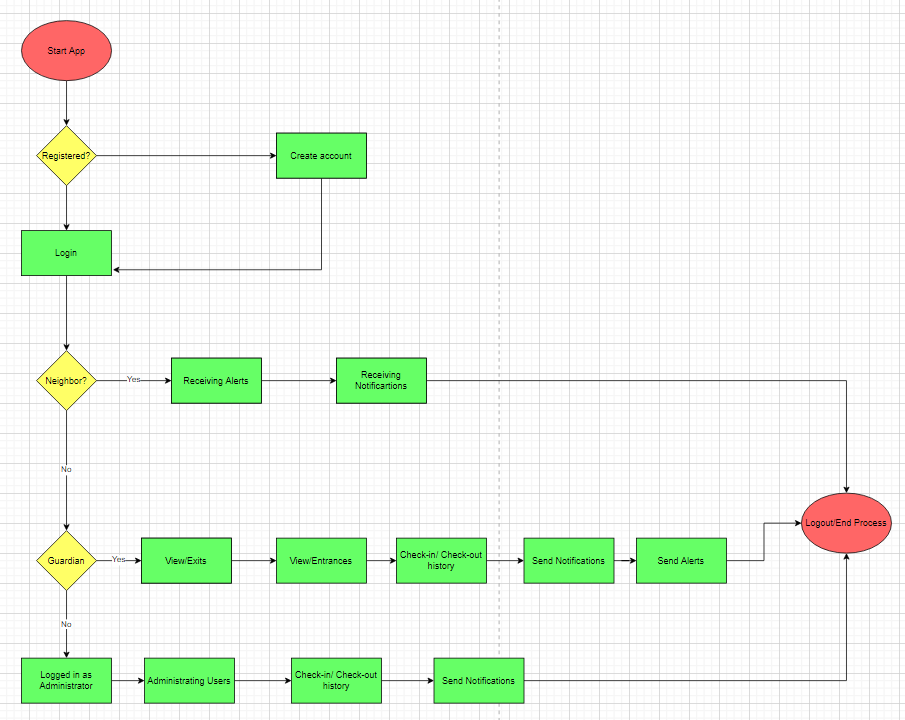
In this diagram the presentation is given of how it is planned to make the application database. It consists of four tables: visitors, home, user, and roles. Visitors connect to the home table, then from there it connects to Users and finally to Roles. The tables are connected by means of the primary keys and foreign keys.

**Database DDL Scripts:**



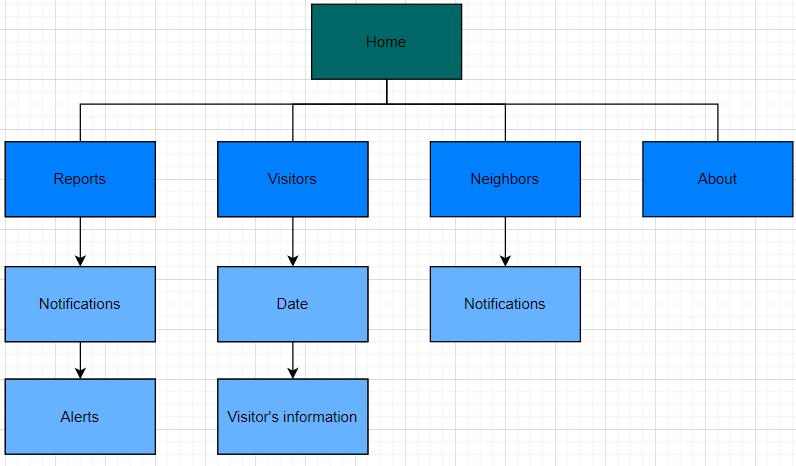
This is another presentation of the tables for the web application database. This gives more specific information in the format that is given in each field of the table. In most tables, users don't have to worry as much about their information “id”, “status”, “startDate”, and “modificationDate". These fields are to know the date the tables were created, if there was any change and if the status of the tables is active or not.

**Flow Chart/Process Flow:**



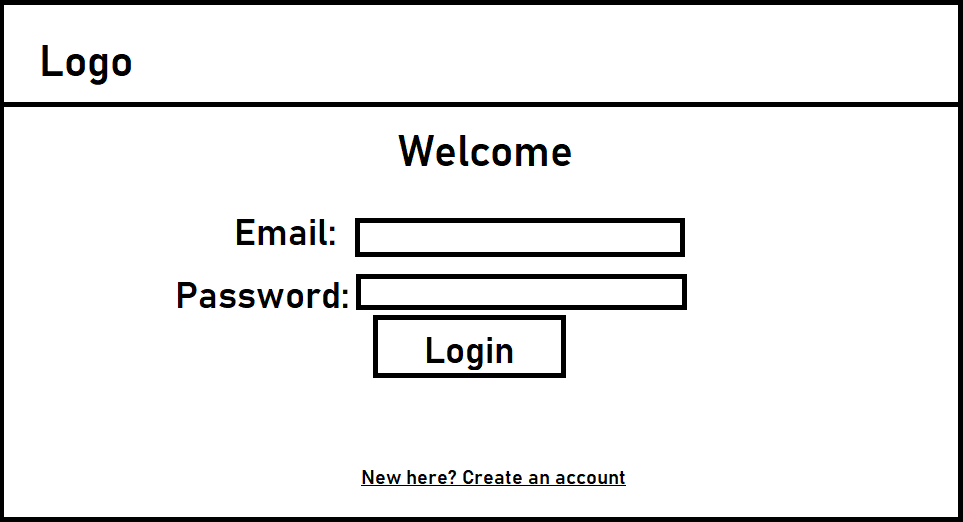
This Flow Chart represents more on how the web application will work when using it. There are certain conditions to be able to enter the page, such as having an account for example. After that, depending on the role you have of the division, you have access to use certain functions of the application. The one with the fewest functions is the neighbor, that refers to the people who live in the subdivision. The one with the most functions would be the guardian, since it is the role that will have much more activity in this project. He oversees giving security to those who live in the subdivision. If you want to terminate the application, you only have to logout.

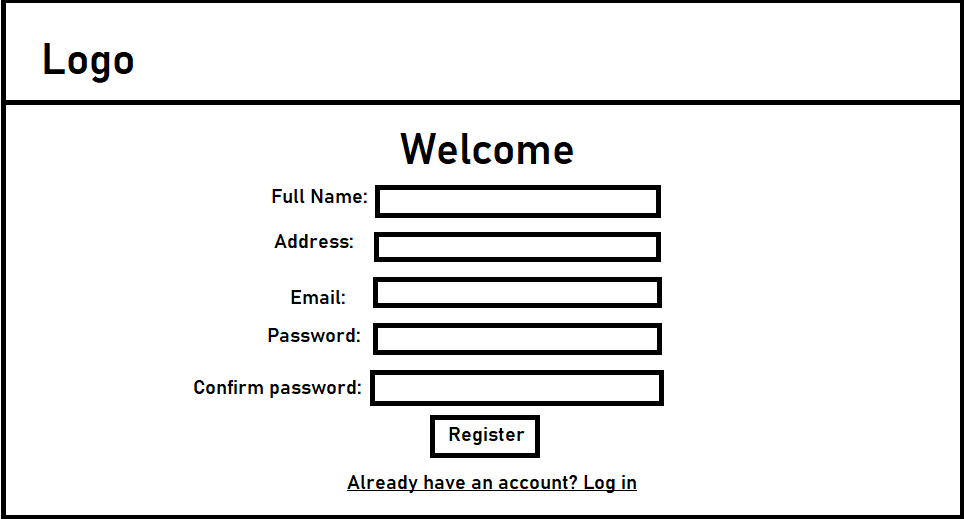
**Sitemap Diagram:**

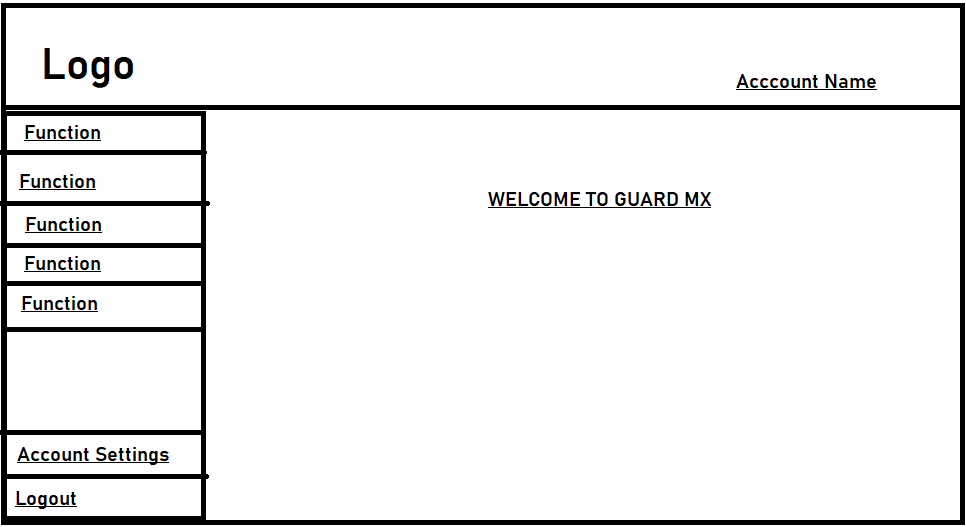


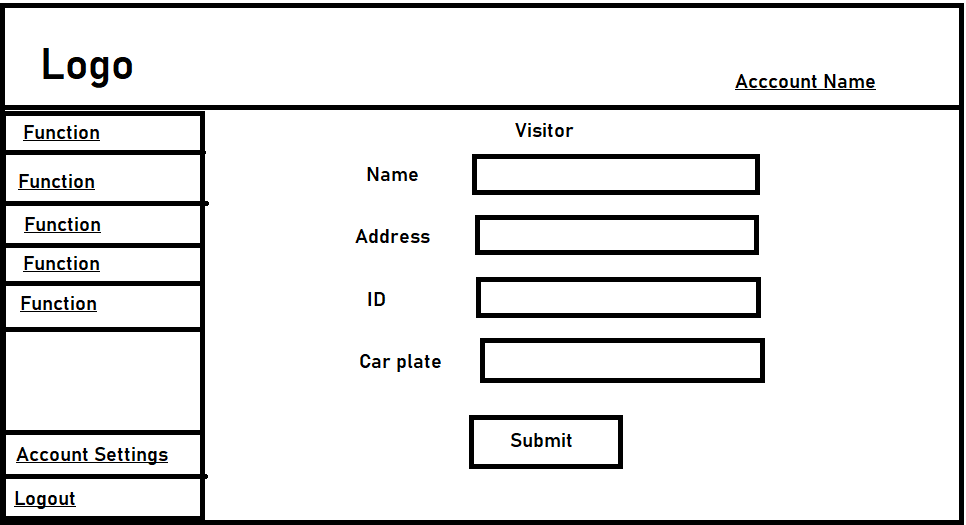
It is a simpler presentation of knowing what the web application will contain in terms of web pages and the functions that can be accessed. The users who will be able to have more access to more functions are those that contain the role of guardian and administrator. Those who have the role of neighbor, which are the ones that live in the subdivision, contain little but enough to fulfill the objective of the project.

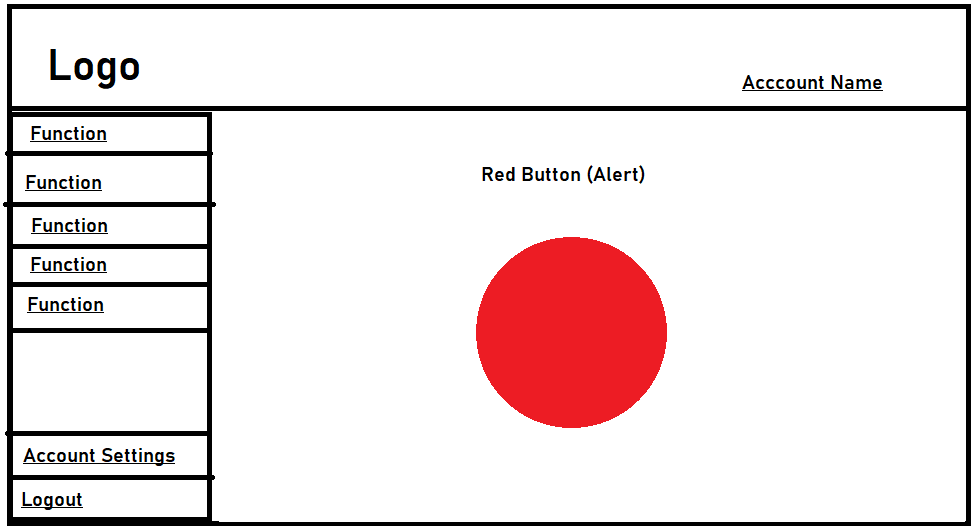
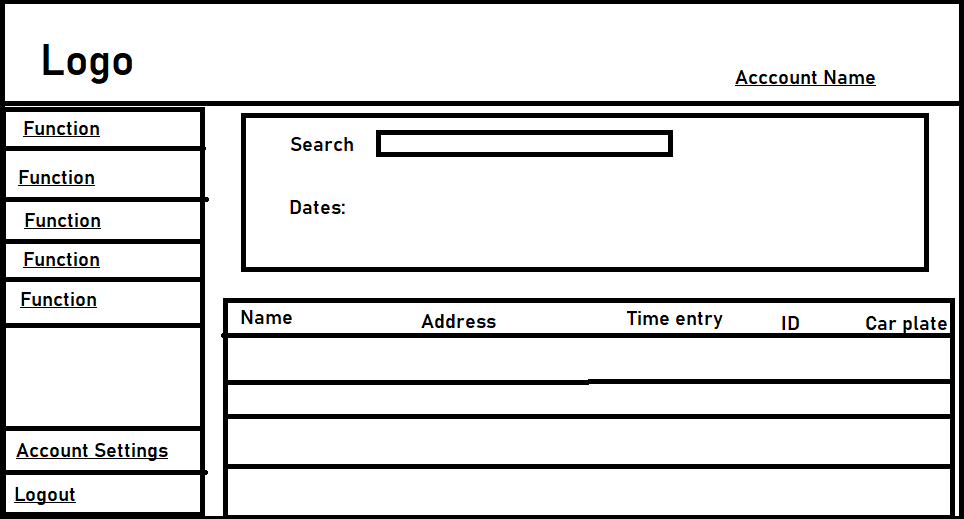
**User Interface Diagram:**

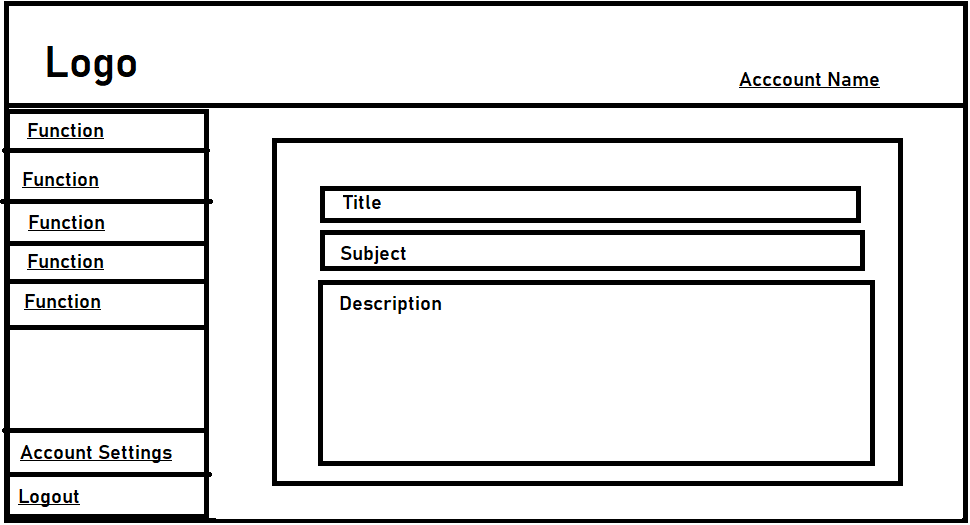


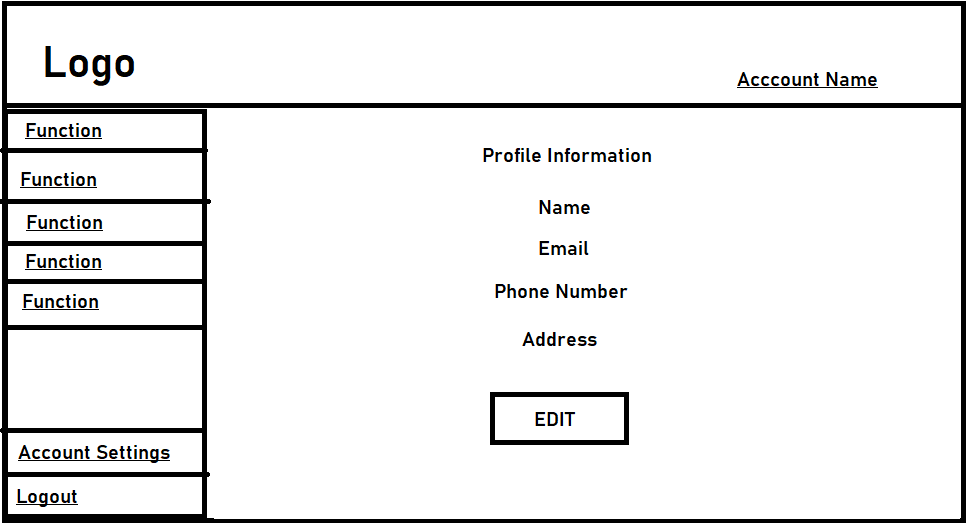




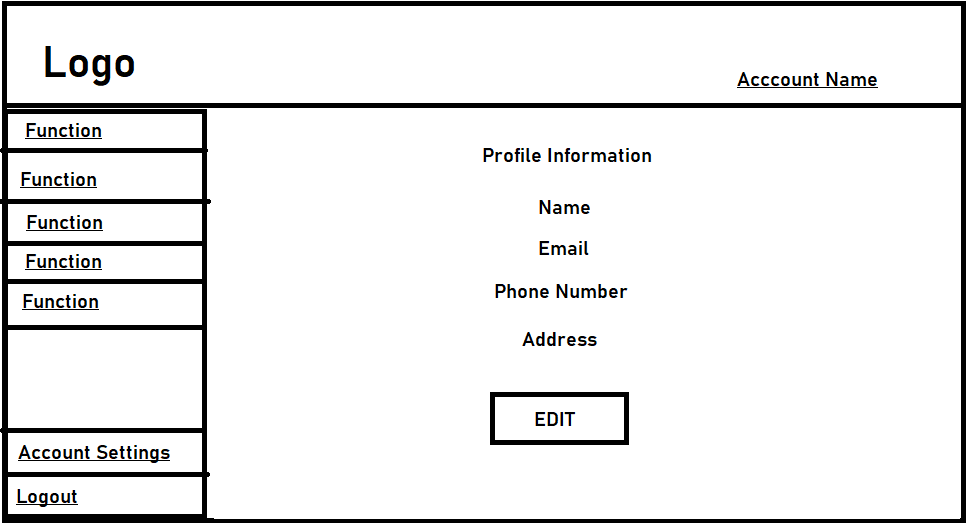


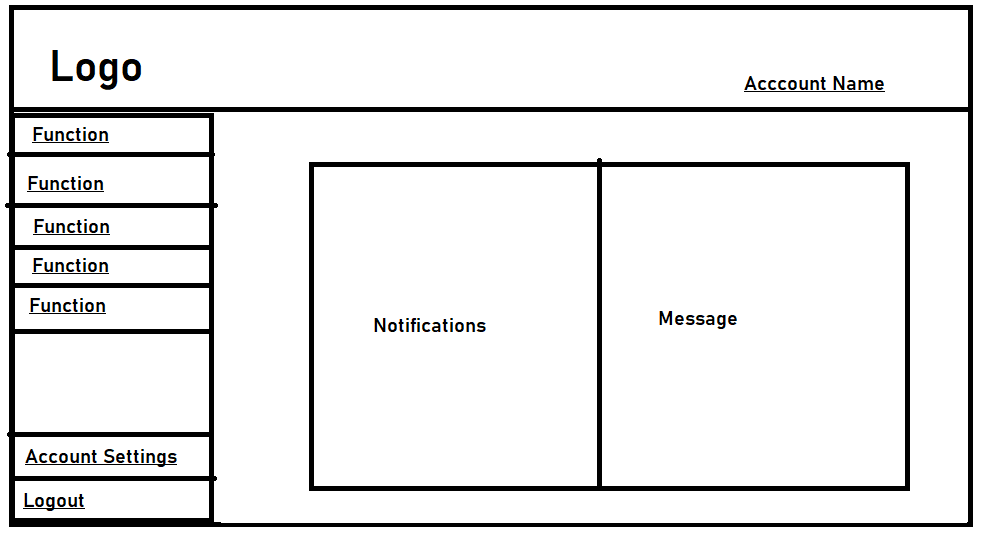


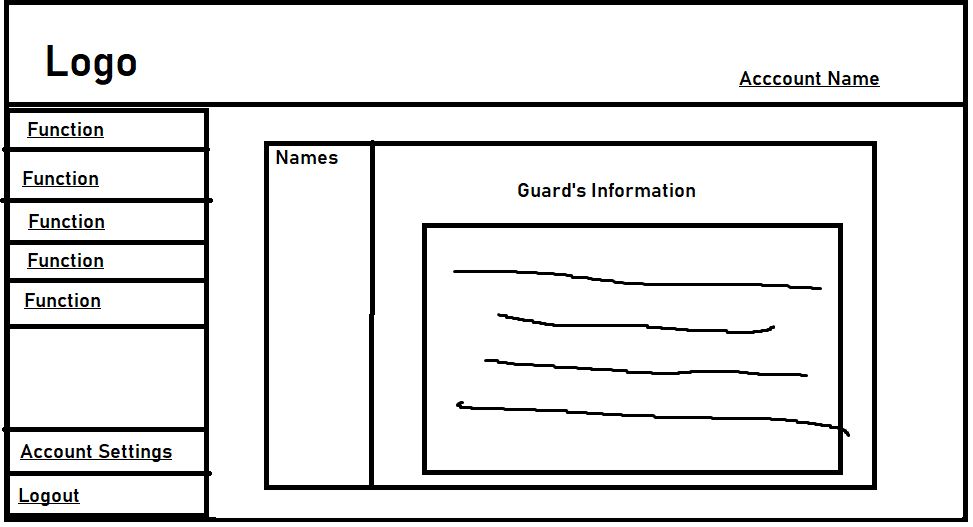


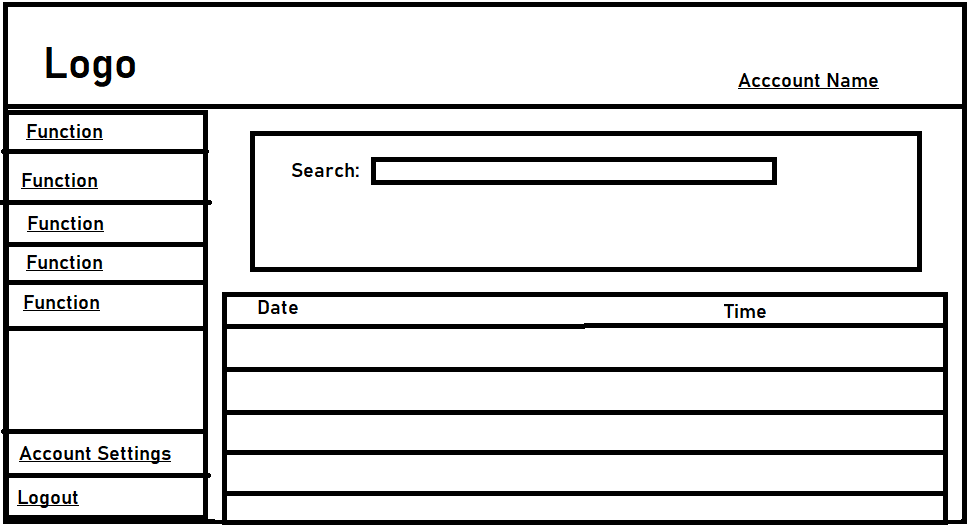


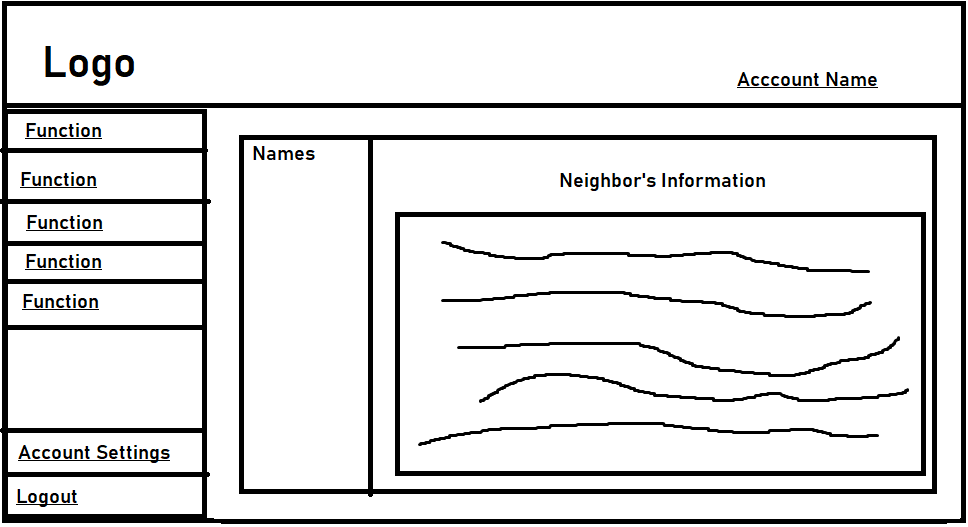
Neighbors Section

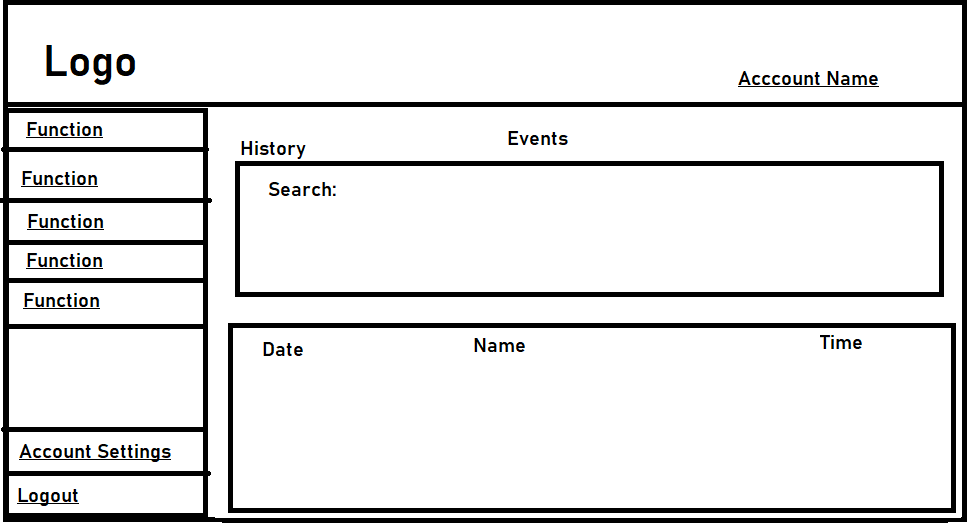












These drawings approximate how the web application will be designed. Not all the wireframes that are at the top are going to be for all types of users. Some will have different interfaces that you will be able to access. Apart from the main functions that are going to be obtained in the web application, also to see other web pages that are going to be created to obtain information from certain people. An example is that the neighbor will be able to access the information of the person who oversees taking care of the subdivision.

**UML Diagram:**

A picture containing text, map, indoor

Description automatically generated

This is another presentation of the tables in the web application database. It is not as detailed as DDL Scripts, but it gives a better picture of how the tables are linked in the data types that are each field. The way the tables are connected is as follows:

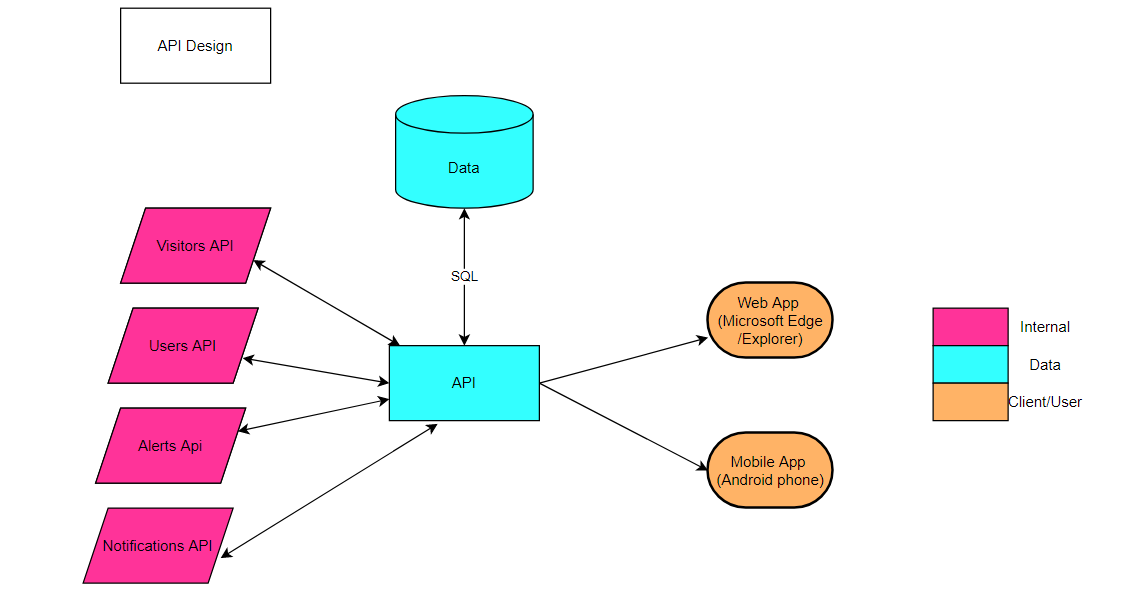
-A house can have many visits

-A User can have several houses

-A user can only have one role

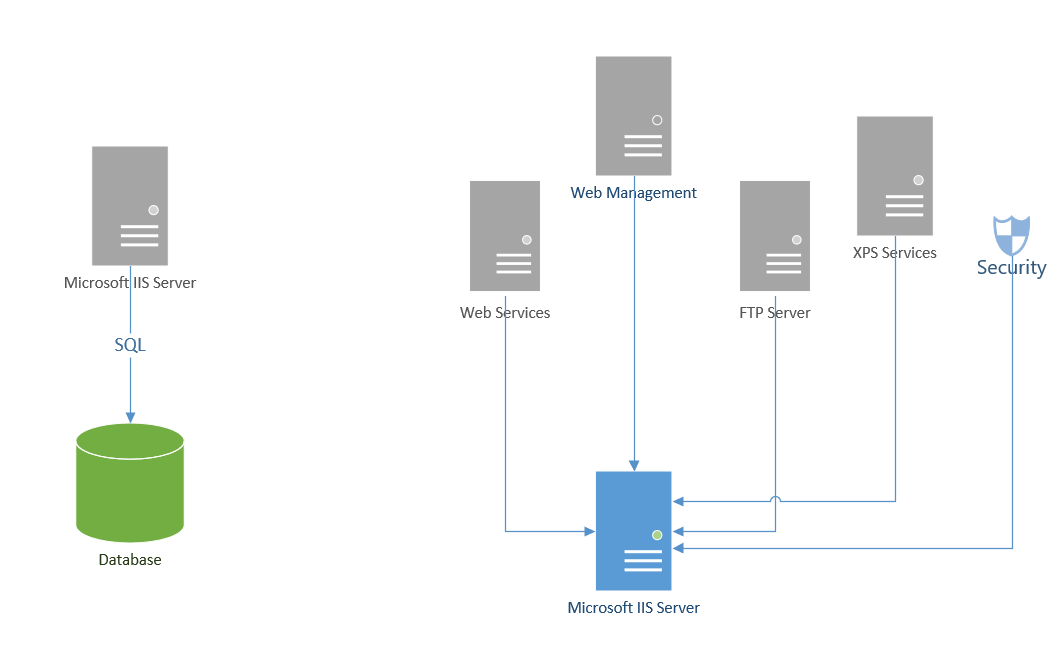
-Admin and Security can send notifications to the resident

**Service API Design:**



This is a representation of how the API that I am going to create for my web application is going to be. The only subjects that will use the API are those that have the application and have the correct electronic device. The main functions that will be added in the new API are for the section. In the end, the API will have the objective of being able to access the data in the database.

**Other Documentation:**



This is extra information about the servers and services that are being used. Only one database is used in this project. In order to be able to use the server perfectly, which is Microsoft IIS Server, you must activate certain services to avoid errors, you can have all the possible permissions to be able to make the project, and it is compatible with the web application.

Appendix A – Technical Issue and Risk Log

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Issues and Risk Log | | | | | | | | |
| **Issue or Risk** | **Description** | **Project Impact** | **Action Plan/Resolution** | **Owner** | **Importance** | **Date Entered** | **Date to Review** | **Date Resolved** |
| I/R | What is the issue or risk? | How will this impact scope, schedule, and cost? | How do you intend to deal with this issue? | Who manages this issue? |  |  |  |  |
|  | Spend an Azure server if the free services are not enough | 200dlls | Double check the free services | Adrian |  |  |  |  |
|  | Not compatible with Android to Web app | The Android users will not be able to use this | Research more about the frameworks and the applications that I am using | Adrian |  |  |  |  |

Appendix B – References

*Servicios gratuitos | Microsoft Azure*. (n.d.). Microsoft Azure. Retrieved November 21, 2021, from https://azure.microsoft.com/es-mx/pricing/free-services/

*ASP.NET Core Crash Course - C# App in One Hour*. (2020, November 16). YouTube. Retrieved November 21, 2021, from https://www.youtube.com/watch?v=BfEjDD8mWYg&t=10s&ab\_channel=freeCodeCamp.org

*Top Course - Learn ASP.NET Core 5.0 MVC - CRUD Operations, EntityFramework Core*. (2021, January 22). YouTube. Retrieved November 21, 2021, from https://www.youtube.com/watch?v=DqD-NJf7-OM&ab\_channel=tutorialsEU

*Top Course - Learn ASP.NET Core 5.0 MVC - CRUD Operations, EntityFramework Core*. (2021b, January 22). YouTube. Retrieved November 21, 2021, from https://www.youtube.com/watch?v=DqD-NJf7-OM&ab\_channel=tutorialsEU

Wikipedia contributors. (2021, November 6). *Software portability*. Wikipedia. Retrieved November 21, 2021, from https://en.wikipedia.org/wiki/Software\_portability

Appendix C – External Resources

|  |  |
| --- | --- |
| **GIT URL:** | *The GIT URL (if applicable).* |
| **Hosting URL:** | *http://localhost:90/* |