

**Smart Lab System using IOT Controlled by Mobile Application**

**Supervised by:**

Dr. Haitham Safwat

Dr. Akram Salah

Dr. Shaimaa

TA Eng. Ahmed Alaa

**Implemented by:**

1. Ahmed Fouad
2. Mohamed Hatem
3. Samy Mohamed
4. Reem Magdy
5. Mona Saleh

Graduation Project

Academic Year 2018-2019

Midyear Short Documentation

**Content**

1- Introduction.

1.1- Description.

1.2- Rationale.

2- Hardware and Software Environment Requirements.

2.1- Operation Environment.

2.2- Development Environment.

3- External Entities.

4- Users and User Cases.

4.1- Defining the Users of the System.

4.2- Use Case Diagrams.

5- Interface Requirements.

5.1- Screen Displays.

6-Models

6.1- ERD.

6.2- Class Diagram.

6.3- Sequence Diagram.

7- General Model for the System.

8- Functional Requirements.

9- Non-Functional Requirements.

10- Exception Handling Specification.

11- Glossary of Terms.

12-References.

**1- Introduction:**

1.1- Abstract:

**An IOT Based system to control Devices and sensors inside a smart lab, the system is divided into 3 parts: mobile application, embedded devices and a platform to connect both of them together.**

-The mobile application will allow authorized users to access devices inside a lab using either voice commands or through a GUI interface, these devices are: A/C, Lights, Projector and PCs. As well as being able to get Readings: Temperature and Brightness, and in the case of a fire breaking out, the user will be notified through the application that a fire has occurred.

-The embedded devices will allow us to send and receive data from the devices inside the lab, as well as control them by connecting the embedded devices with the desired devices inside the smart lab that we want to control, and giving internet access to the embedded devices for them to send and receive the data.

-The Platform we will be using (FIWARE) is the intermediator between the embedded devices and the mobile application, it is an open source cloud platform that provides virtual machines with limited resources for IOT based projects.

1.2- Rationale:

The reason behind this idea is to make it easier for authorized users to have control either remotely or locally over the smart lab and being able to monitor and use devices located inside.

**2-Hardware and Software Requirements:**

2.1- Operation Environment:

1) Android Platform.

2) Android API 27.

3) Arduino uno R3.

4) Fiware lab.

2.2- Development Environment:

1) Android Studio (Java 8).

2) Arduino IDE.

3) Sqlite Database.

**3-External Entities:**

FIWARE:

-An Open Source platform that combines components that enable the connection to IOT with Context Information Management.

- Exposes standard APIs for data management and exchange, as well as harmonized data models.

**We will be using several components (GEs) from FIWARE which are:**

Context Broker, IOT Agent, IOT Discovery, Context Event Processing.

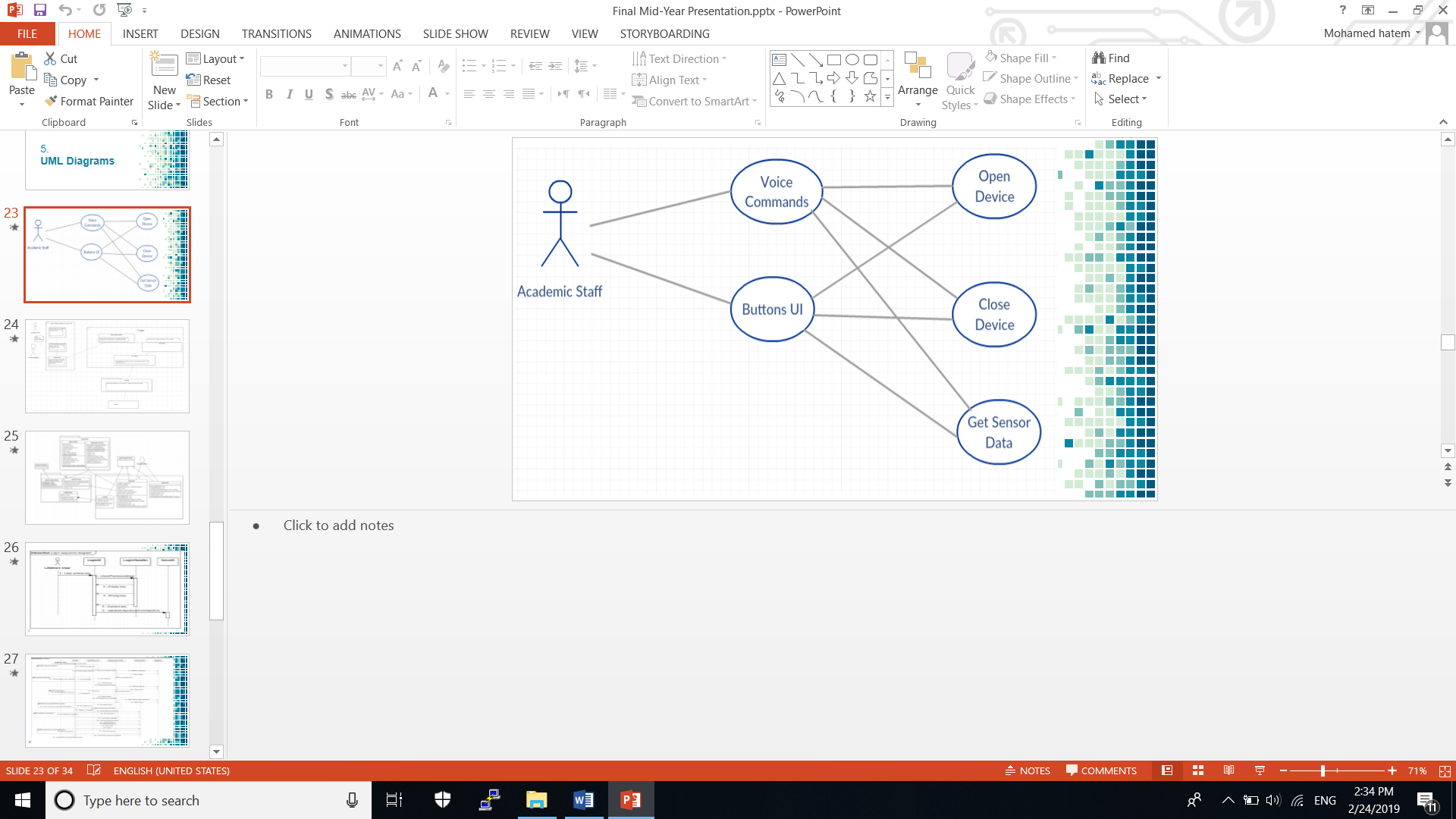
**4-Users and Use Cases:**

4.1- Defining the Users of the System:

Authorized users of the system will have access to the system through a private key which will allow them to use the mobile application.

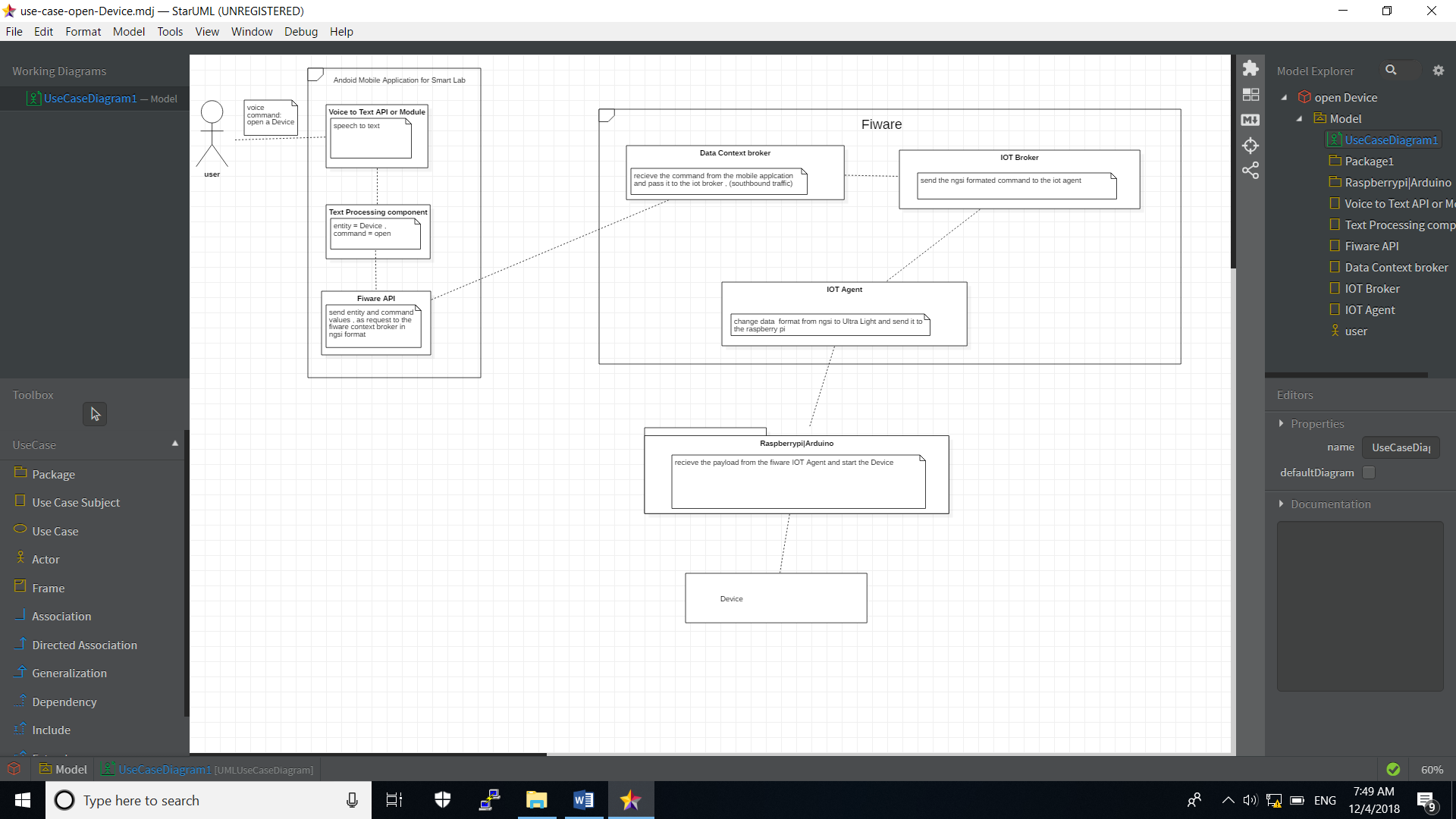
Authorized users are Figures of Authorities that use the lab, they could be TA’s or Doctors.

4.2- Use Case Diagrams:

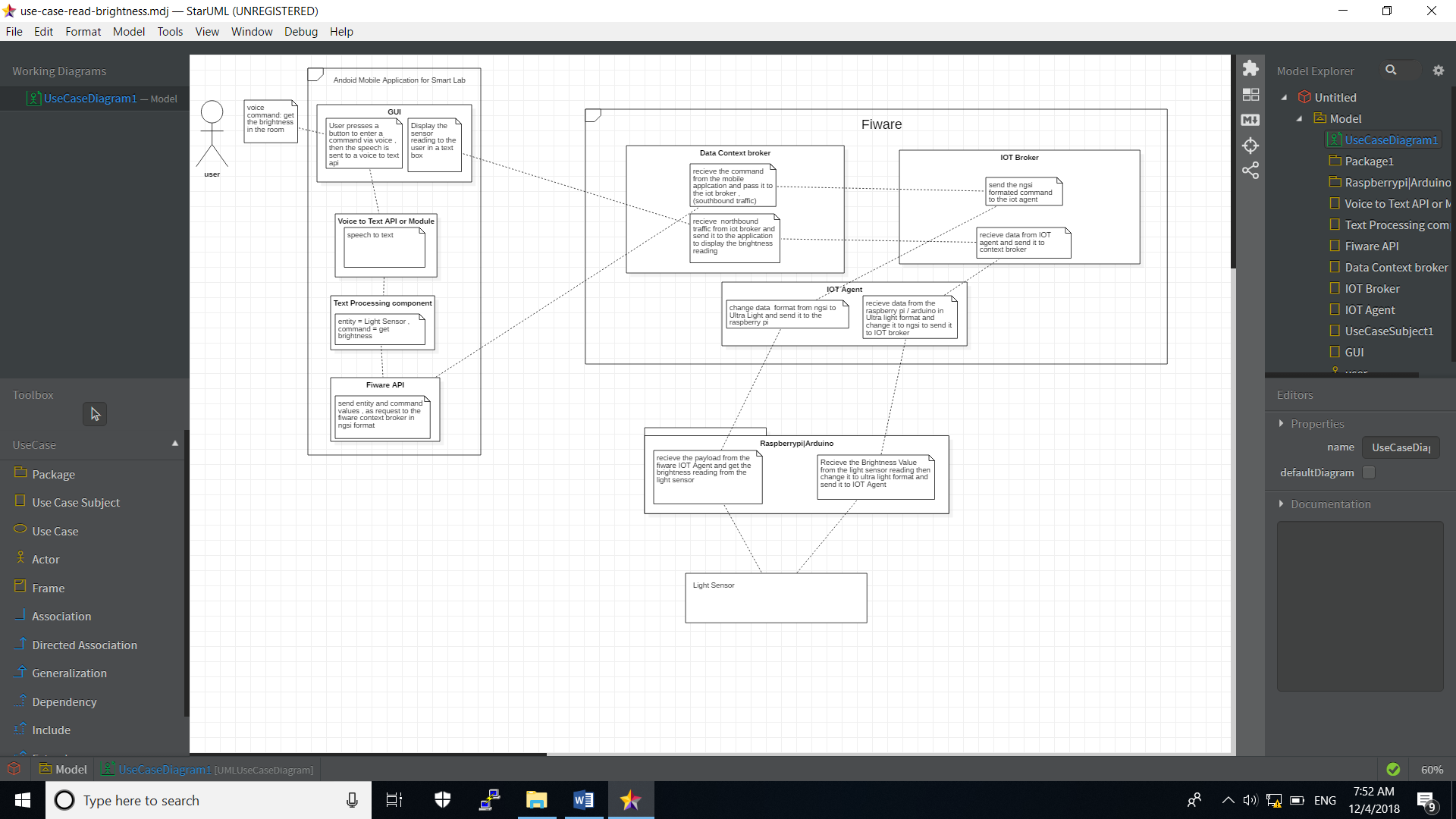


4.2.2- Sample Use Cases:

1- Open A Device.

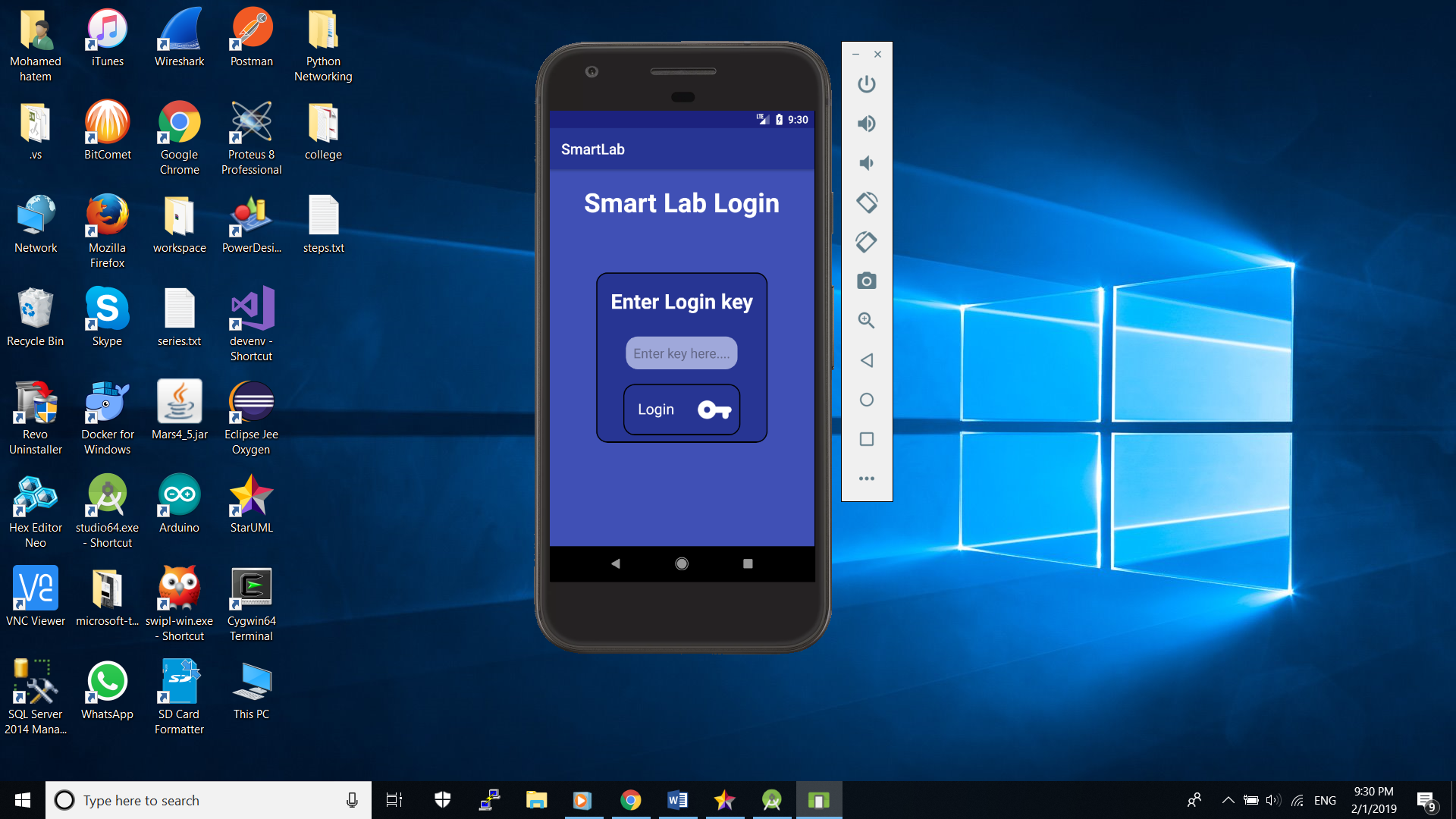


2- Get lab Brightness reading.

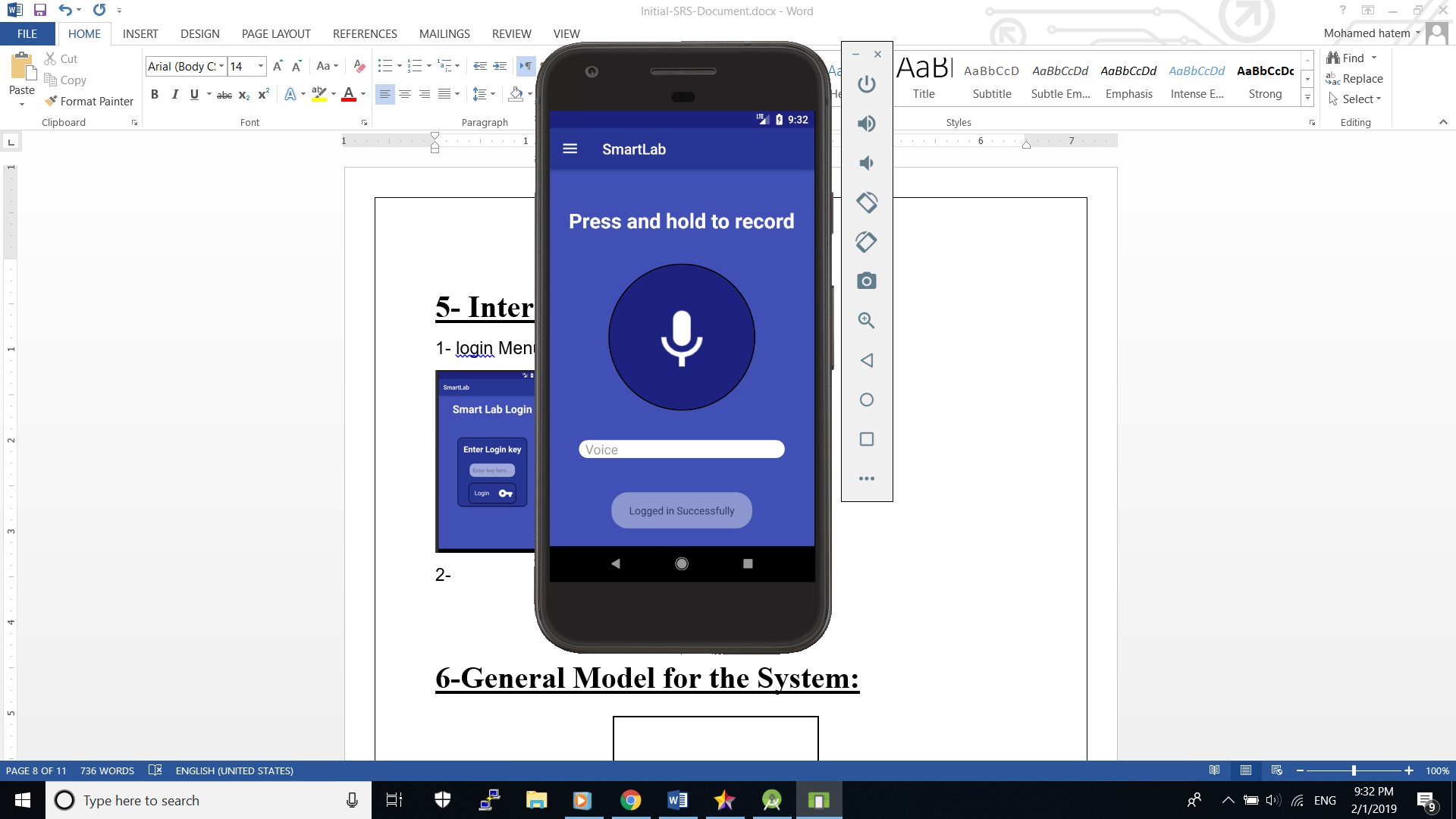


**5- Interface:**

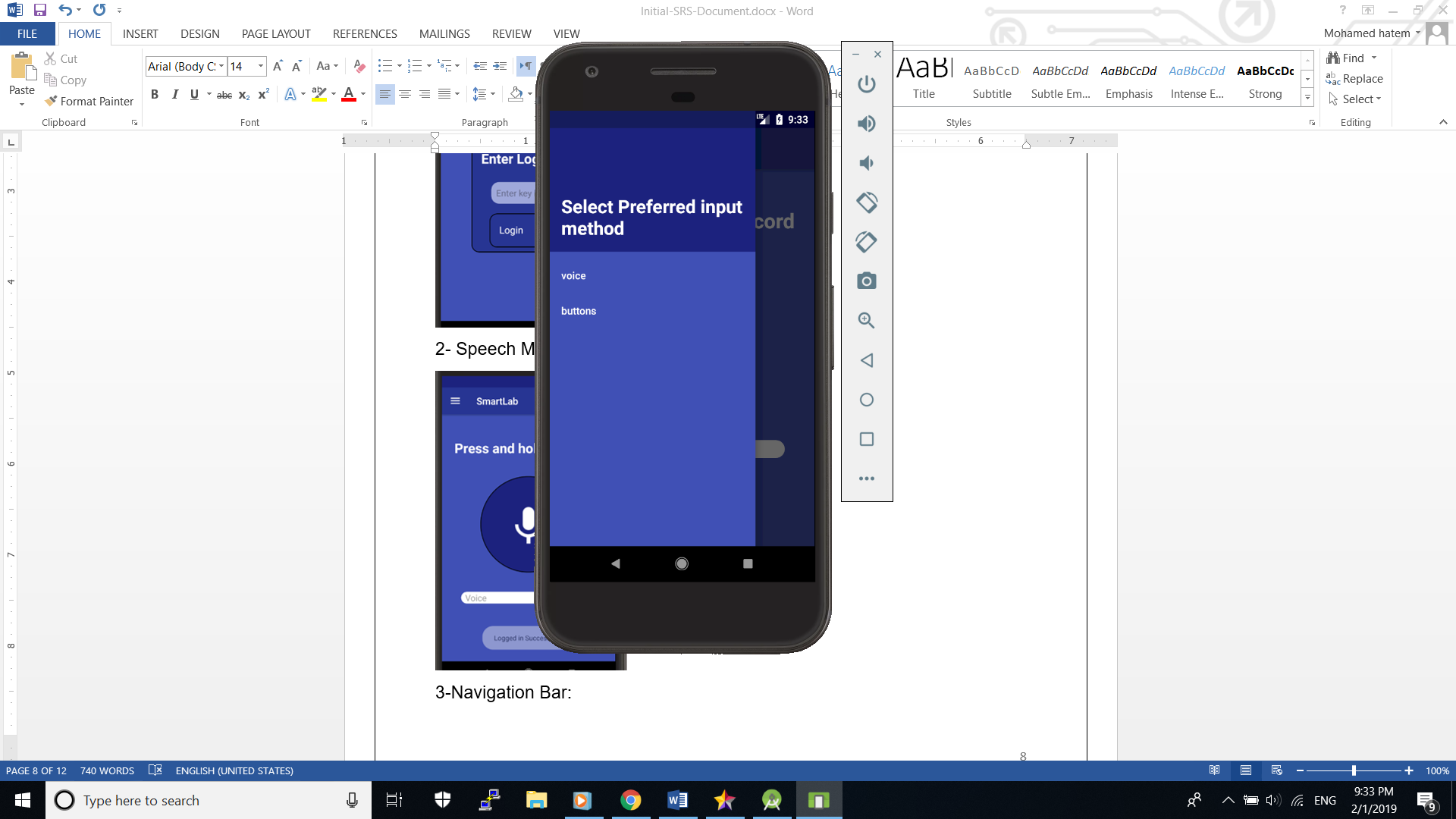
1-Login Menu:



2- Speech Menu:

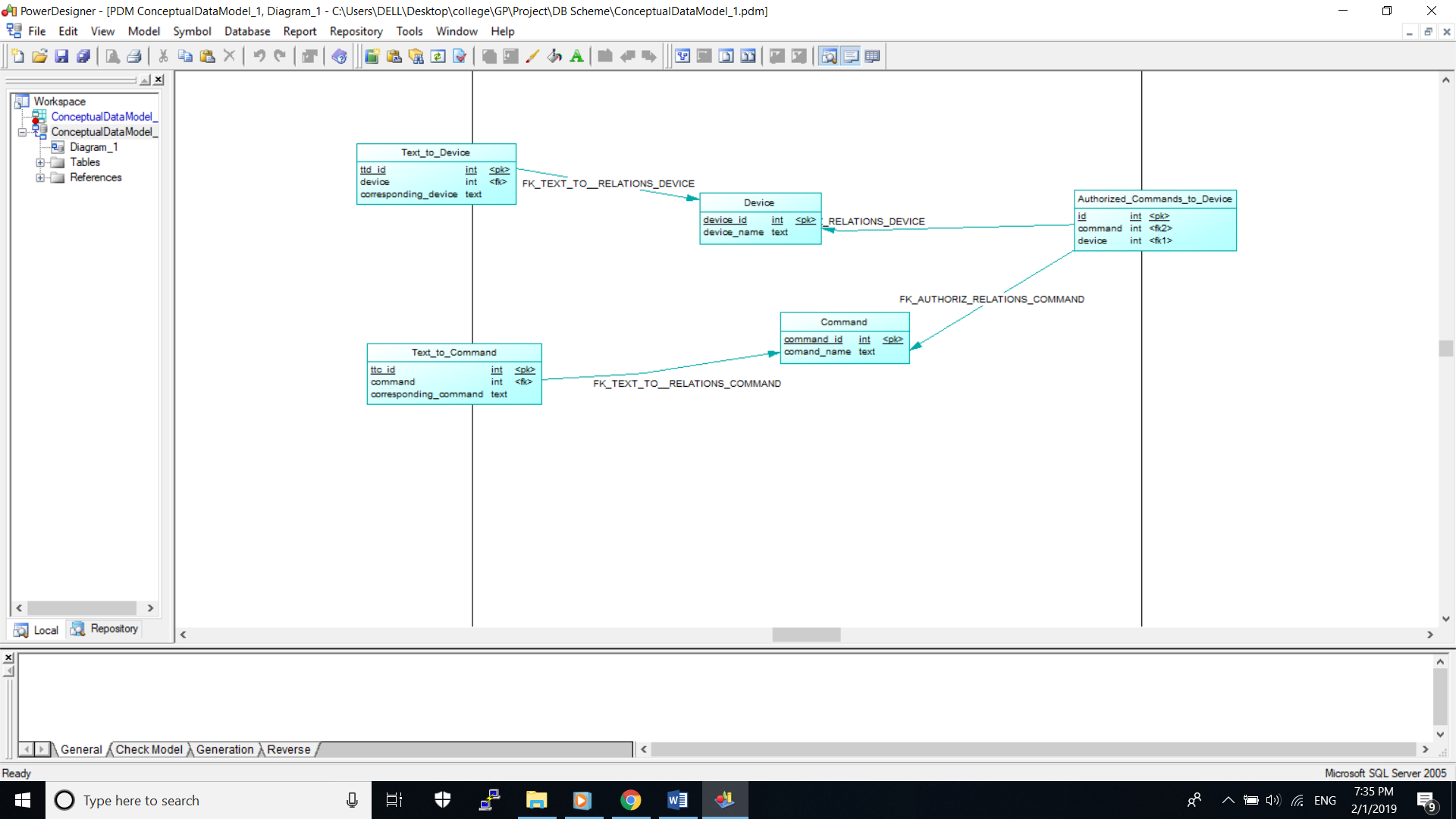


3-Navigation Bar:

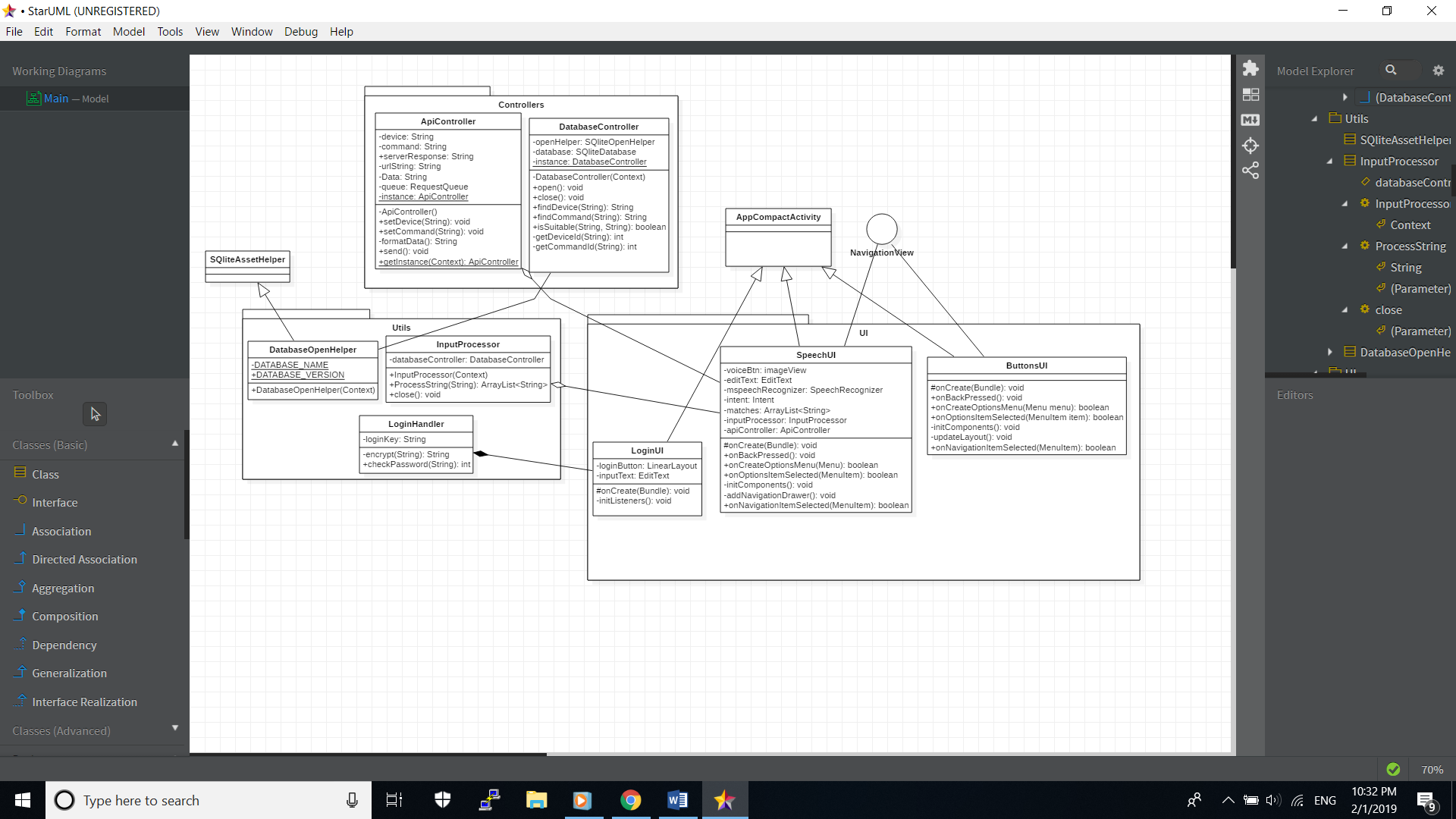


**6-Models:**

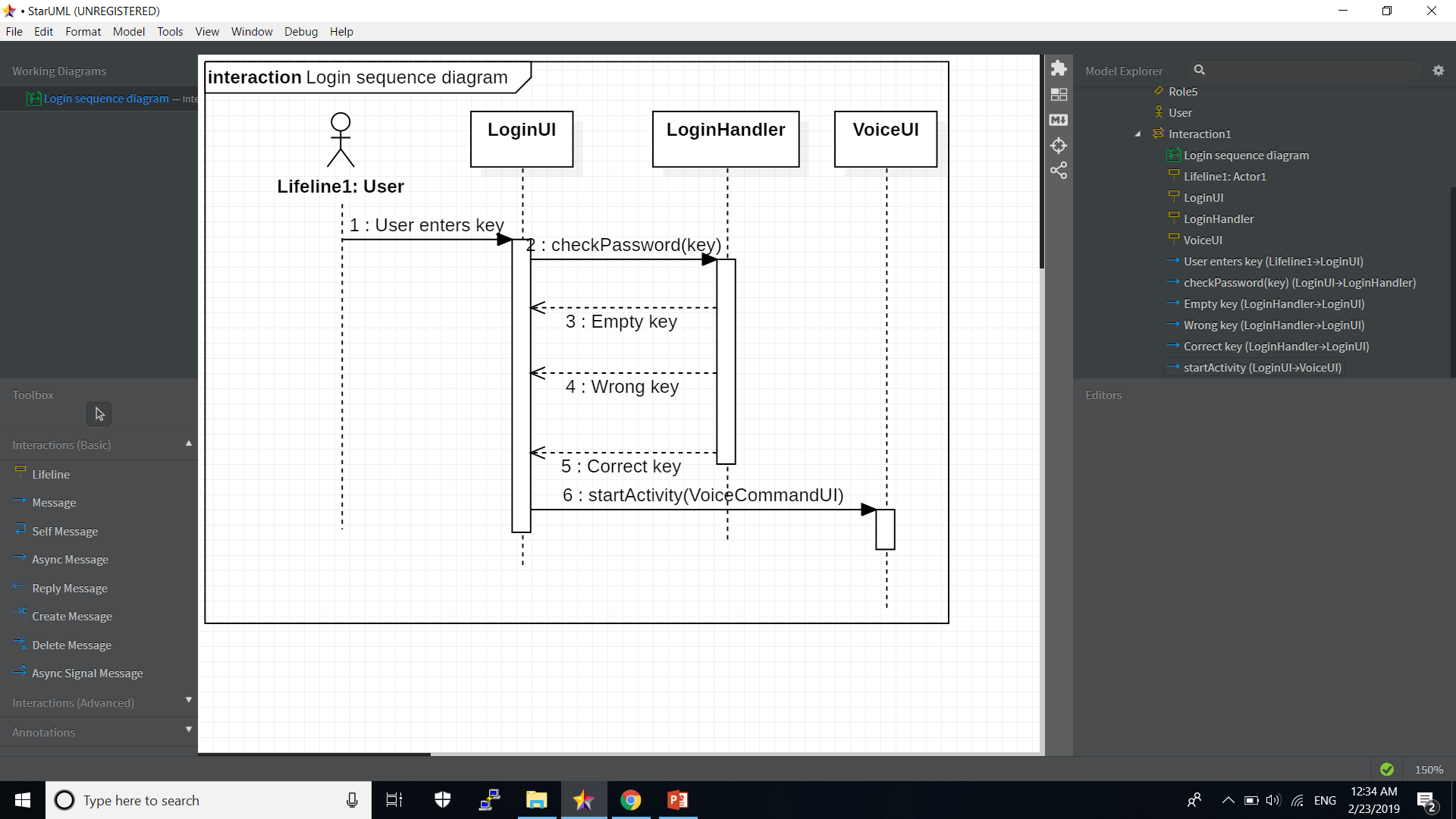
6.1- ERD:

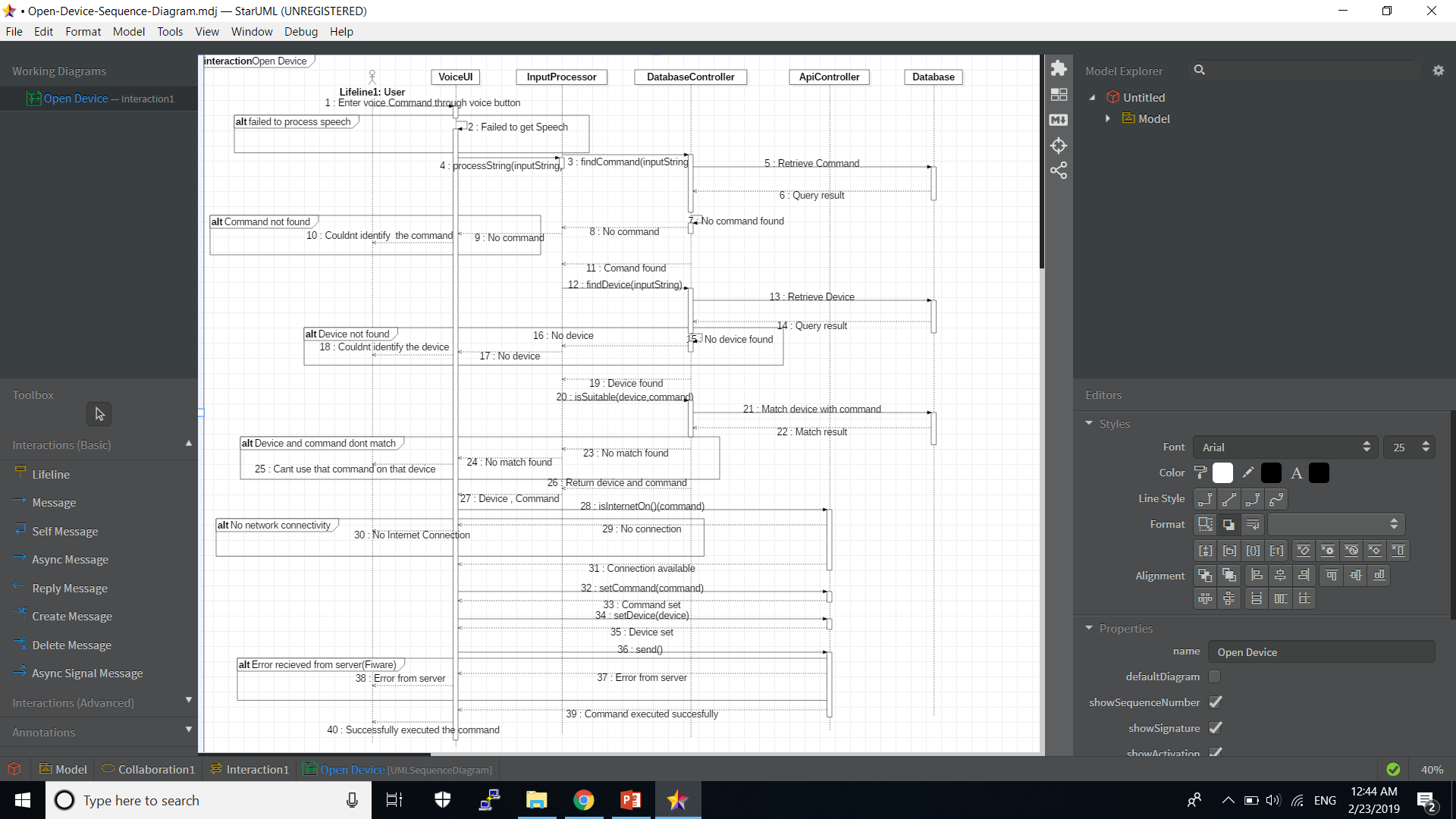


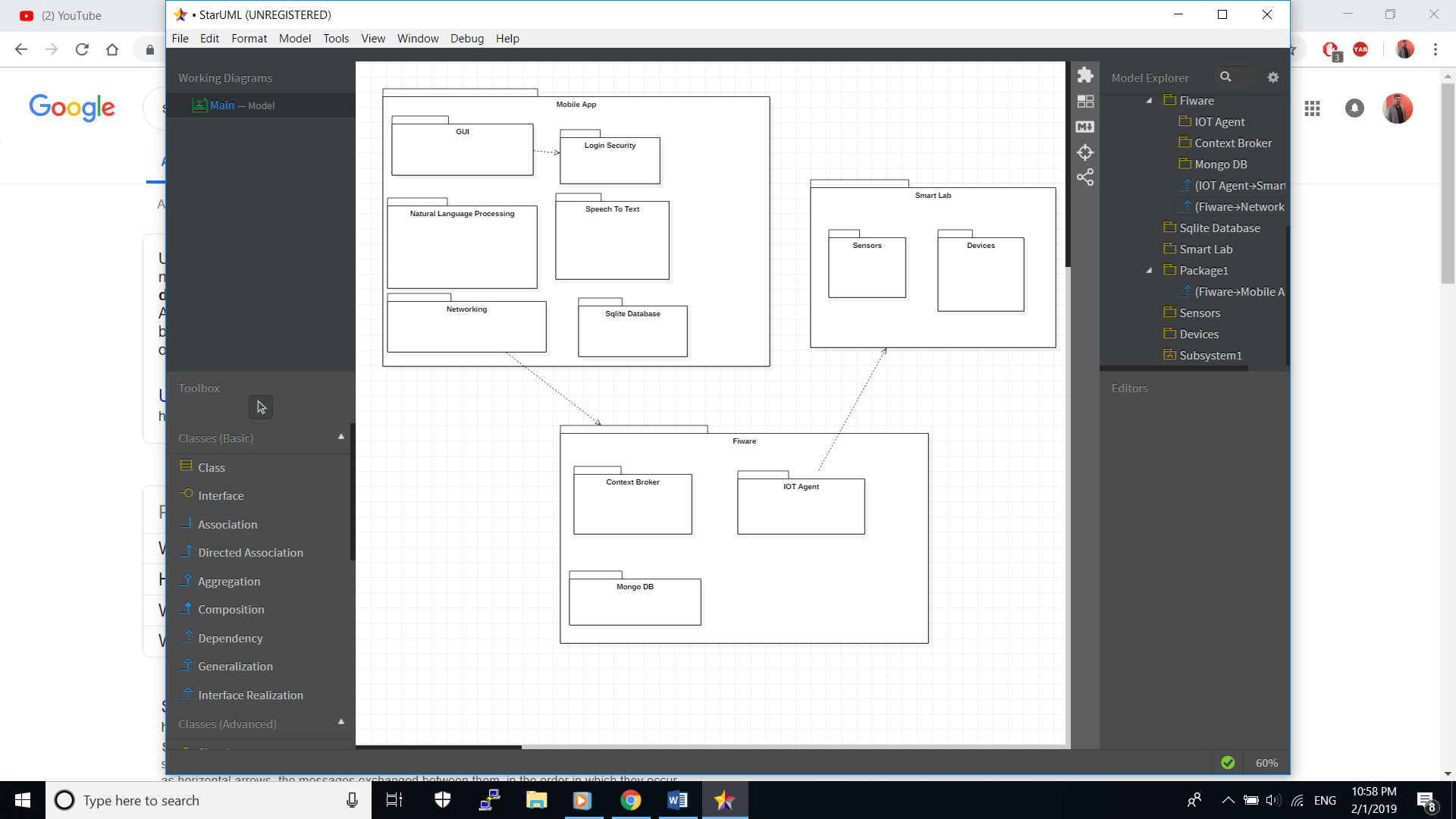
6.2- Class Diagram:



6.3- Sequence Diagrams:





**7-General Model for the System:**

**8-Functional Requirements:**

-User should be able to send commands either by voice or using a GUI in the mobile application to choose a certain action on a certain device.

-User should be able to open and close devices in the smart lab through one of the input methods received by the mobile application.

-User should receive notification in case of emergencies ex: fire.

-User should be able to ask for Readings (light and brightness) through one of the input methods.

**9-Non-Functional Requirements:**

**-Security:** private key is required for the user to login to the application and use its features, which is encrypted using sha256.

**-Usability:** mobile application UI should be well defined and easy to use.

**-Performance:** response from the platform should not exceed 10 seconds.

**10-Exception Handling Specification:**

|  |  |  |
| --- | --- | --- |
| ***Exception*** | ***action*** | ***Message*** |
| Wrong login key | ask user to re-enter it | “you entered the wrong login key!” |
| No internet connectivity | Any input made by the user won’t be sent over to FIWARE platform. |  |

**11-Glossary of terms:**

**GEs:** Generic enablers, components that provide certain functionality.

**NGSI:** JSON + Header, a type of format to send and receive data over the internet.

**Private Key:** a key shared by us and given to authorized users.

**12-References:**

<https://fiware-iot-stack.readthedocs.io/en/latest/>

<https://developer.android.com/training/volley/>