



UNIVERSITÀ DEGLI STUDI
DI GENOVA

Screen In

IOT Project

By : Ziad Janpiah & Amer Al Khoury

2020-2021



UNIVERSITÀ DEGLI STUDI
DI GENOVA

Project proposal

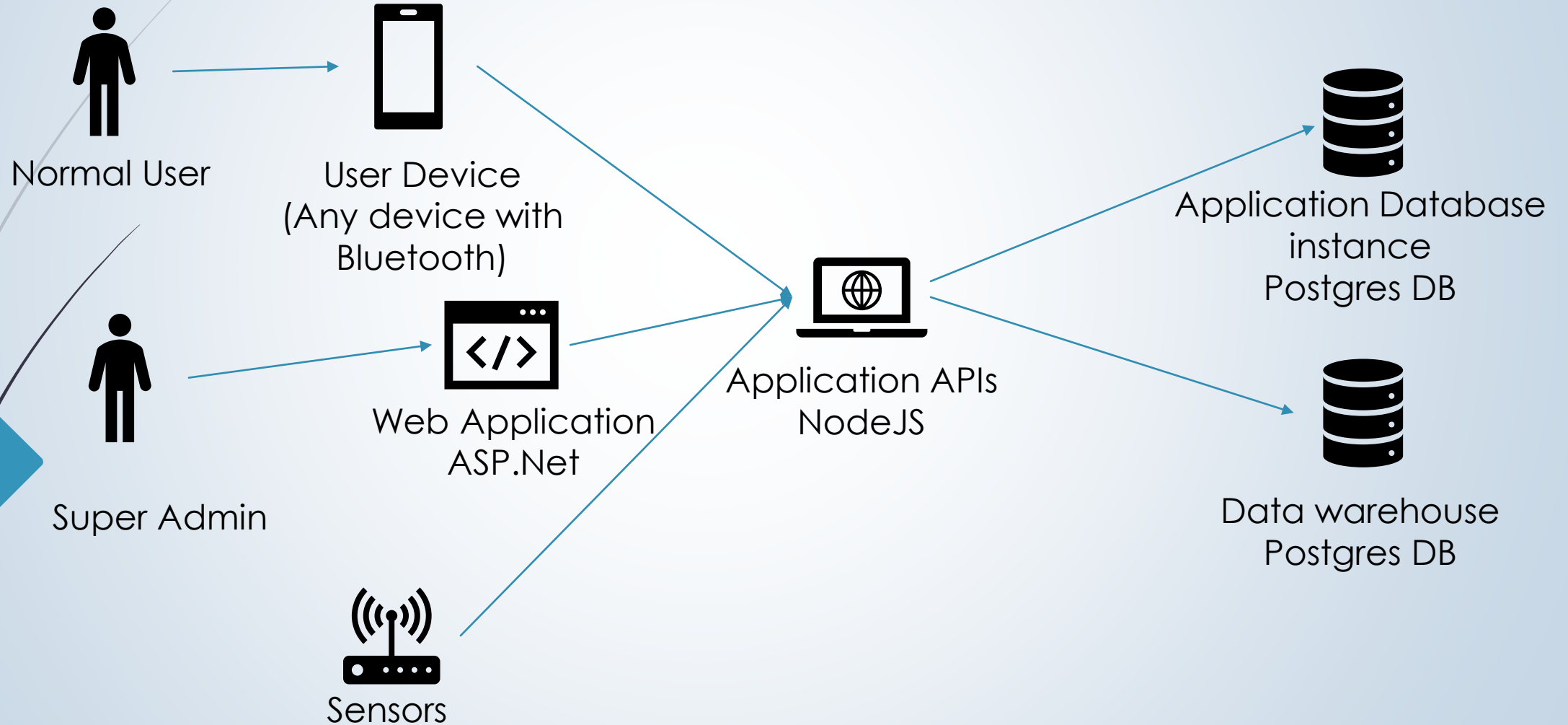


- ❖ The goal of the project is designing a prototype of an indoor contact tracing system to be used to produce an estimation of the number of people inside a building with real-time monitoring and collect data for historical statistics on the usage of the different parts of the building.
- ❖ Our proposed solution consists of :
 1. Server part:
 - ✓ Database layer .
 - ✓ web service layer.
 2. Application part:
 - ✓ Dashboard.
 - ✓ Statistics.
 - ✓ Buildings management.
 - ✓ Movement simulator.



UNIVERSITÀ DEGLI STUDI
DI GENOVA

System Topology



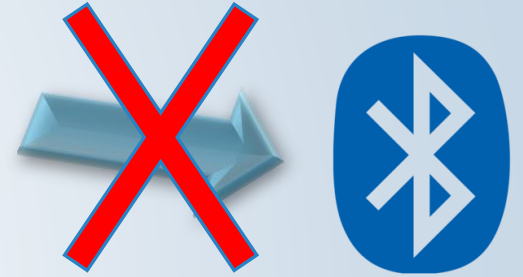
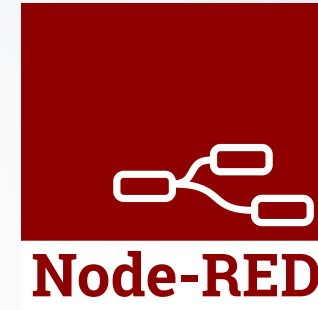


UNIVERSITÀ DEGLI STUDI
DI GENOVA

Issues & Assumptions



We faced a compatibility problem with BLE scanner in Node Red and Node.Js.
The BLE scanner was not able to connect with the Bluetooth driver in our devices



We decided to implement a simulator that mimics the people movements and generate the Bluetooth signals.



How does the system work?



- Users would be able to register the system using email and password
- Users would be able to sign in the application using email and password in order to register their devices

Normal User

Sensors

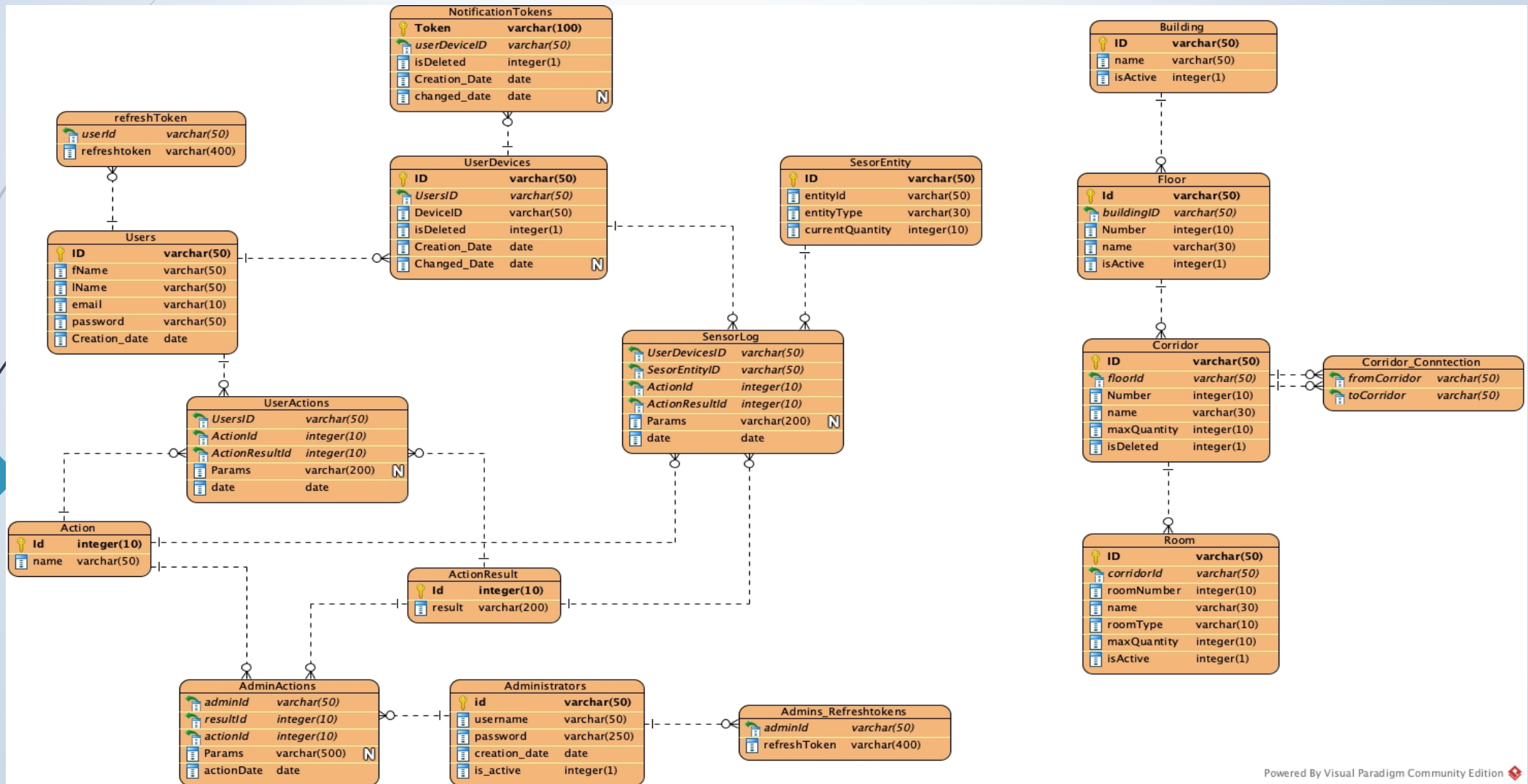
- All building's room, corridor or entrance will have sensor
- Those sensors will scan all the possible devices periodically
- Sending the scanned devices to the server

- Buildings Management
- System reports
- System Statistics
- User movement simulation

Admin Users



Database Design





Application APIs



Application APIs

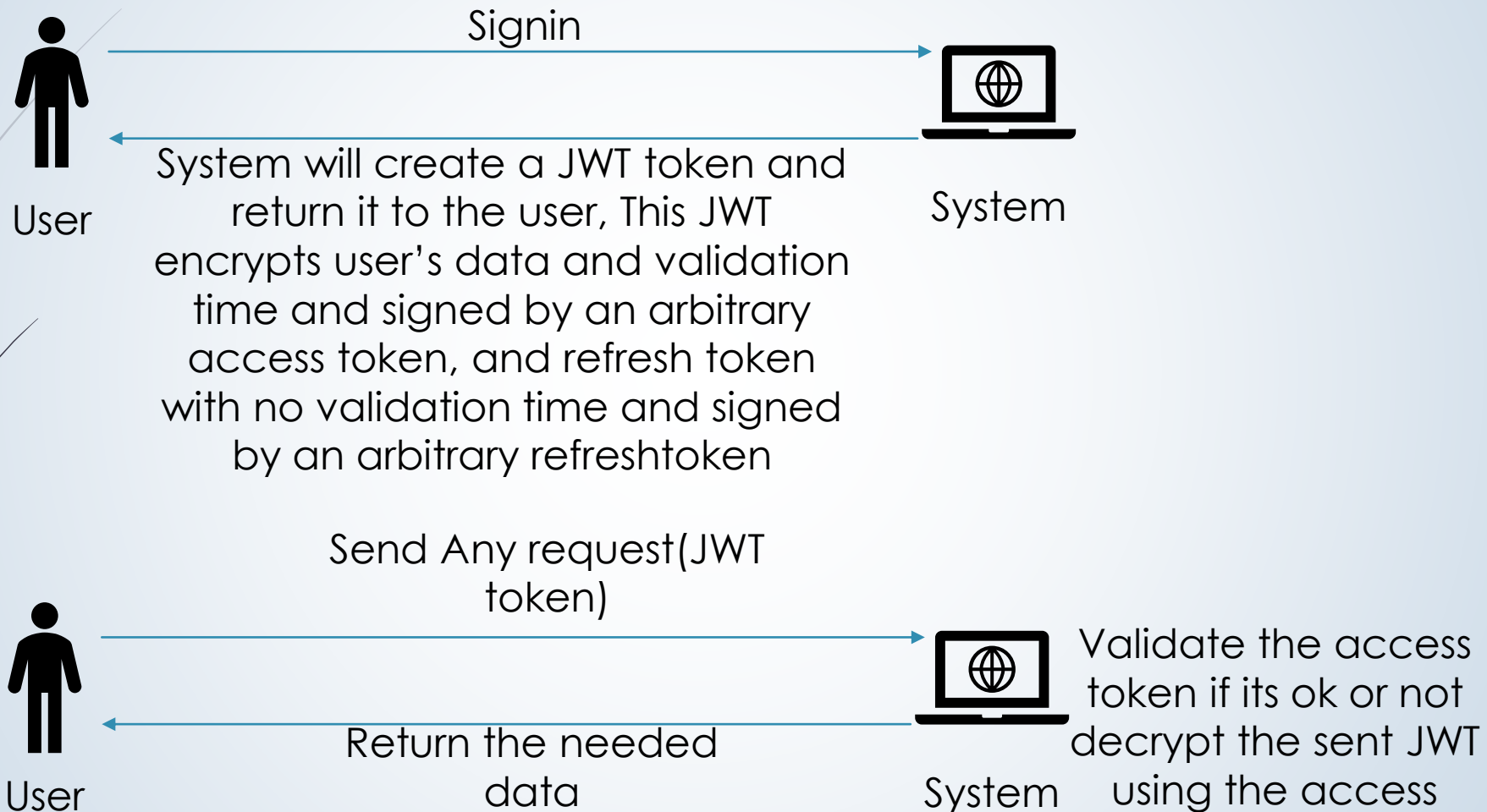
- Restful webservice
- NodeJS

Used libraries

- **express Server:** Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.
<https://www.expressjs.com/>
- **jsonwebtoken:** it make use for Node-JWT through the application to generate webtokens for mobile application users and for the admin web interface.
<https://www.npmjs.com/package/jsonwebtoken>
- **pg:** PostgreSQL client module for node.js. <https://www.npmjs.com/package/pg>
- **UUID:** Node.js module for creating uuids. <https://www.npmjs.com/package/uuid>
- **dotenv:** zero-dependency module that loads environment variables from a file into process.env. <https://www.npmjs.com/package/dotenv>
- **nodemon:** tool that helps the develop nodejs based applications by automatically restarting the node application when file changes in the directory are detected.
<https://www.npmjs.com/package/nodemon>



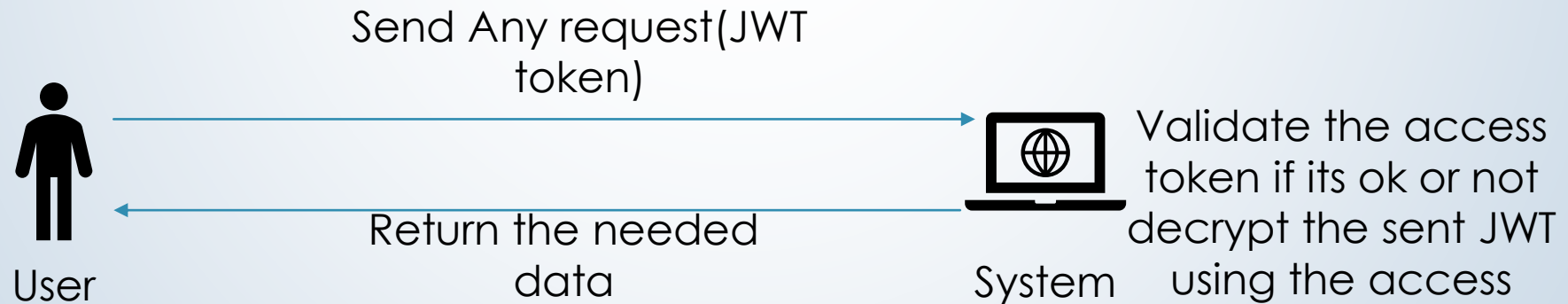
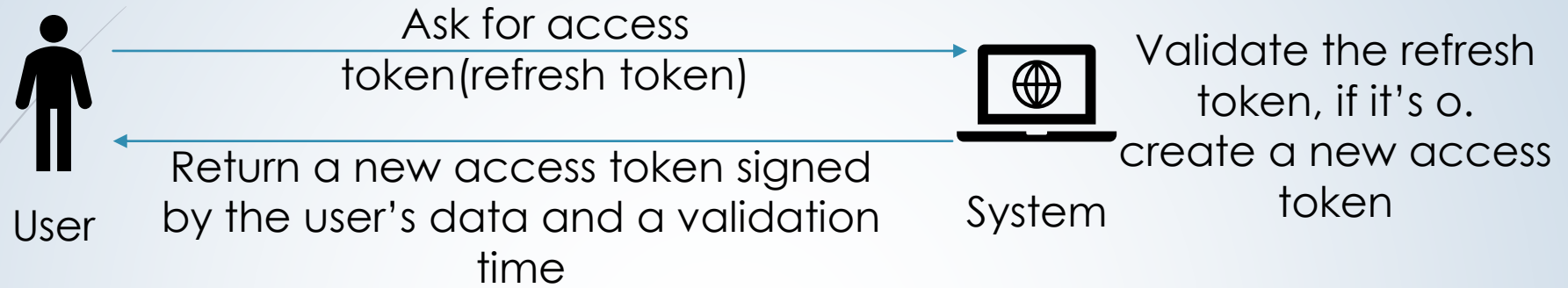
JWT (JSON Web token)





UNIVERSITÀ DEGLI STUDI
DI GENOVA

JWT (JSON Web token)





Application APIs



- **Server.js:** the main file in the web application which considered as the routing file, and have the definition of all the restful webservice
- **Model:** have two folders:
 - Classes: have all the model classes that would be serialized to JSON format on returning the results
 - Helpers: have the authorization module that work on authorize the JWT token that sent by the admin in order to execute some request
- **DB:** have multiple files that works on connecting with the DB using pool of connections, each file is responsible for some APIs

Here we will provide API documentation:



Microsoft Excel
Worksheet



UNIVERSITÀ DEGLI STUDI
DI GENOVA



Building Modeling

- Each building contains one or many floors.
- Each floor contains one or many corridors.
- Each corridor contains one or many rooms.
- Each corridor can be connected with one or many corridors.
- The room can be a normal room or an entrance.
- ❖ There is a BLE scanner in each corridor or room.
- ✓ The user can add or modify buildings by following the previous structure.
- ✓ the user can activate or deactivate the scanners.
- ✓ The user can setup the maximum capacity of each corridor or room.



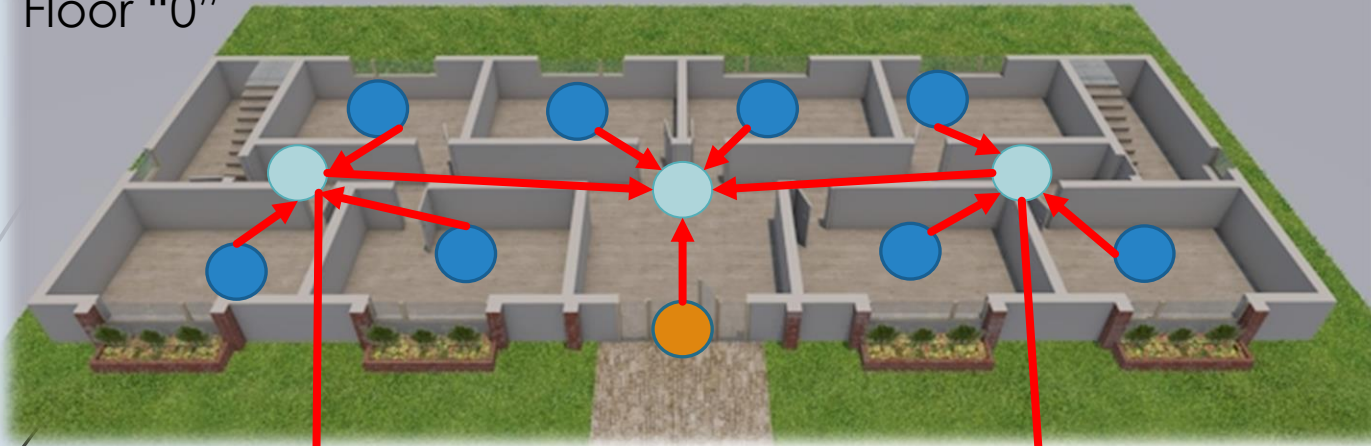


UNIVERSITÀ DEGLI STUDI
DI GENOVA

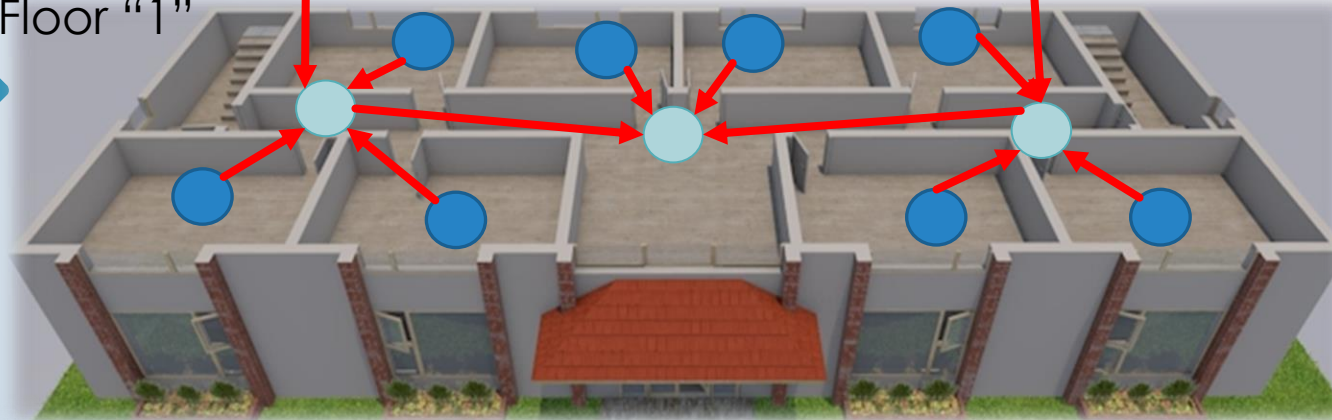
Building Modeling



Floor "0"



Floor "1"



- Entrance
- Room
- Corridor

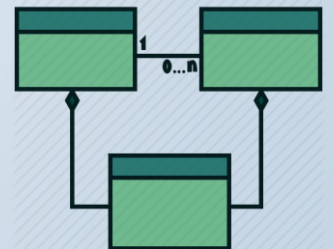
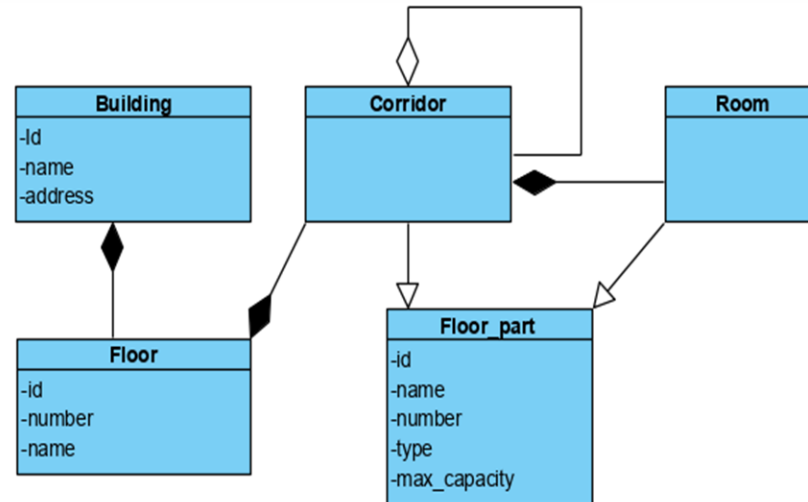


UNIVERSITÀ DEGLI STUDI
DI GENOVA

Movement Simulator

To simulate the people movements through the building, we need to know the building structure, formulate it in a structure that can enabling navigation or traversing, and the best structure in such situation is the Graph data structure.

Get the building data form Database.
Convert it into an OOP structure





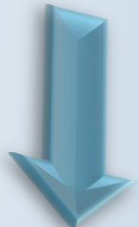
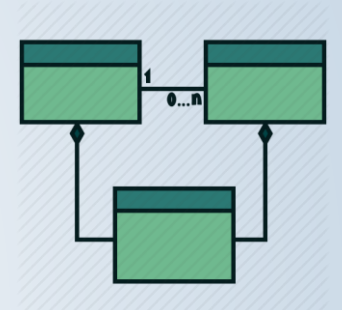
UNIVERSITÀ DEGLI STUDI
DI GENOVA

Movement Simulator

Generate_building_graph(Building b)

```
vertices=[]  
links=[]  
Foreach (Floor f in b.floors)  
    foreach(Corridor c in f.corridors)  
        vertices.add(c)  
        foreach(Room r in c.rooms)  
            vertices.add(r)  
            links.add(c,r)  
        foreach(Corridor cl in c.corridor.corridor_links)  
            if (cl !exist in (links))  
                links.add(c,cl)  
graph.vertices=vertices  
graph.links=links  
return graph
```

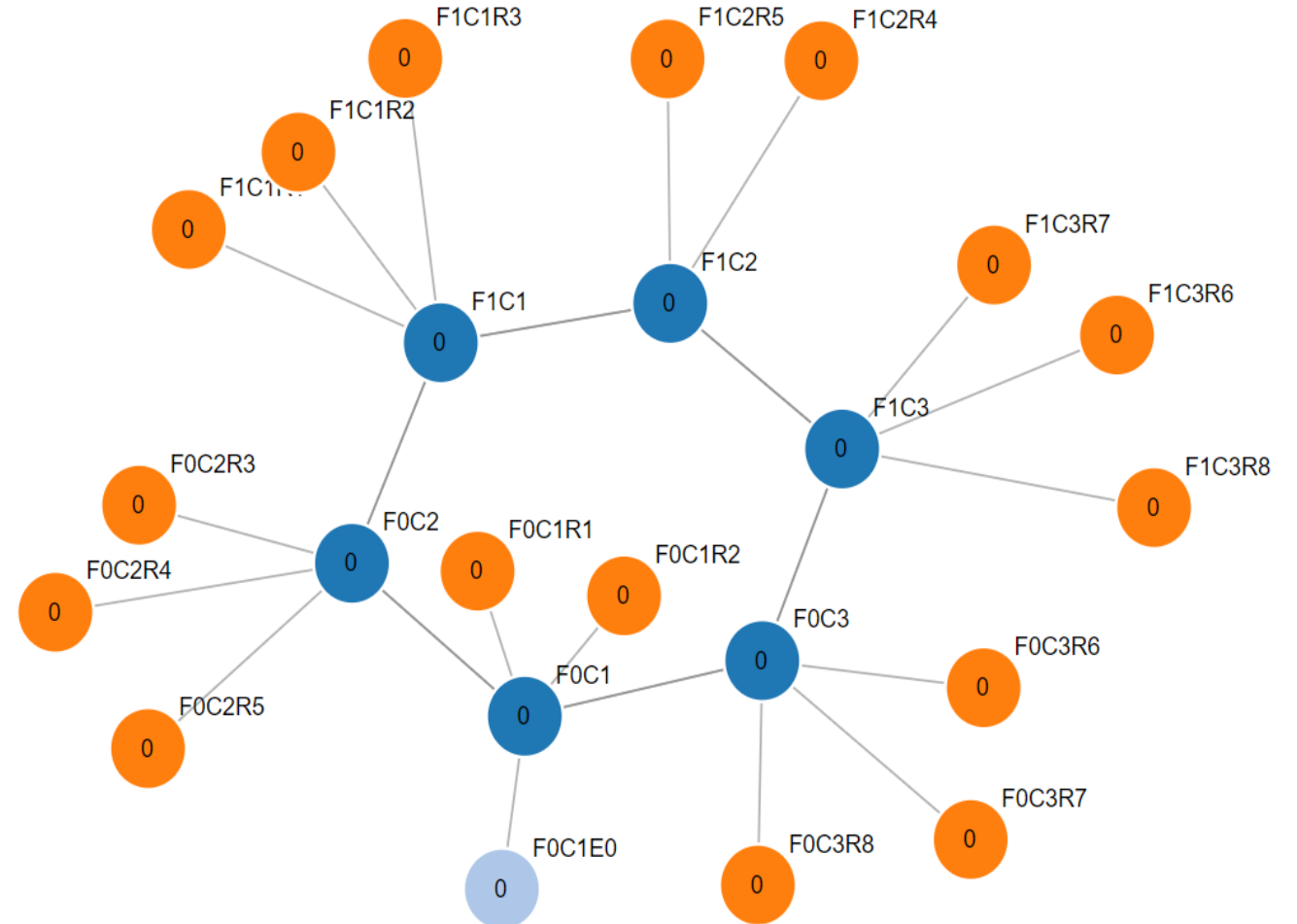
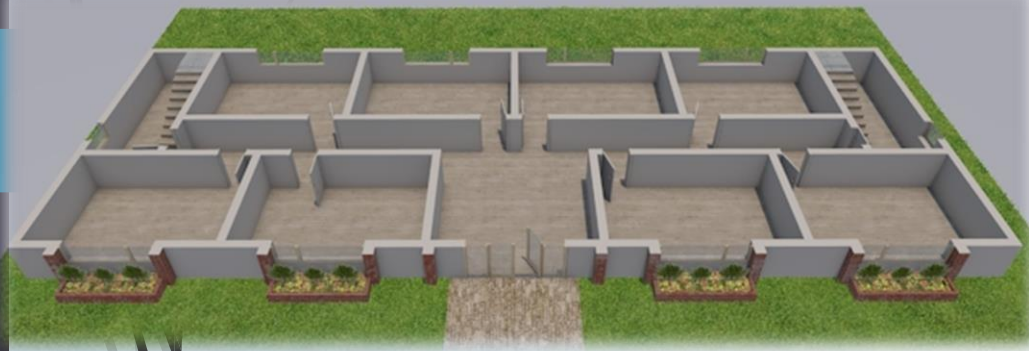
Also, we generate the Adjacency Matrix from the returned graph.





UNIVERSITÀ DEGLI STUDI
DI GENOVA

Movement Simulator





UNIVERSITÀ DEGLI STUDI
DI GENOVA

Movement Simulator



Assumptions

People : students

Maximum Lectures : 4

Configuration

Lecture duration.

Corridor duration.

Scanning interval.

Number of People

Select Building

building 1 ▼

Scanning interval / s : 4



Start Simulation

Building Graph

Lecture duration / m : 45



Stop Simulation

Corridor duration / m : 10



Number of People: 50



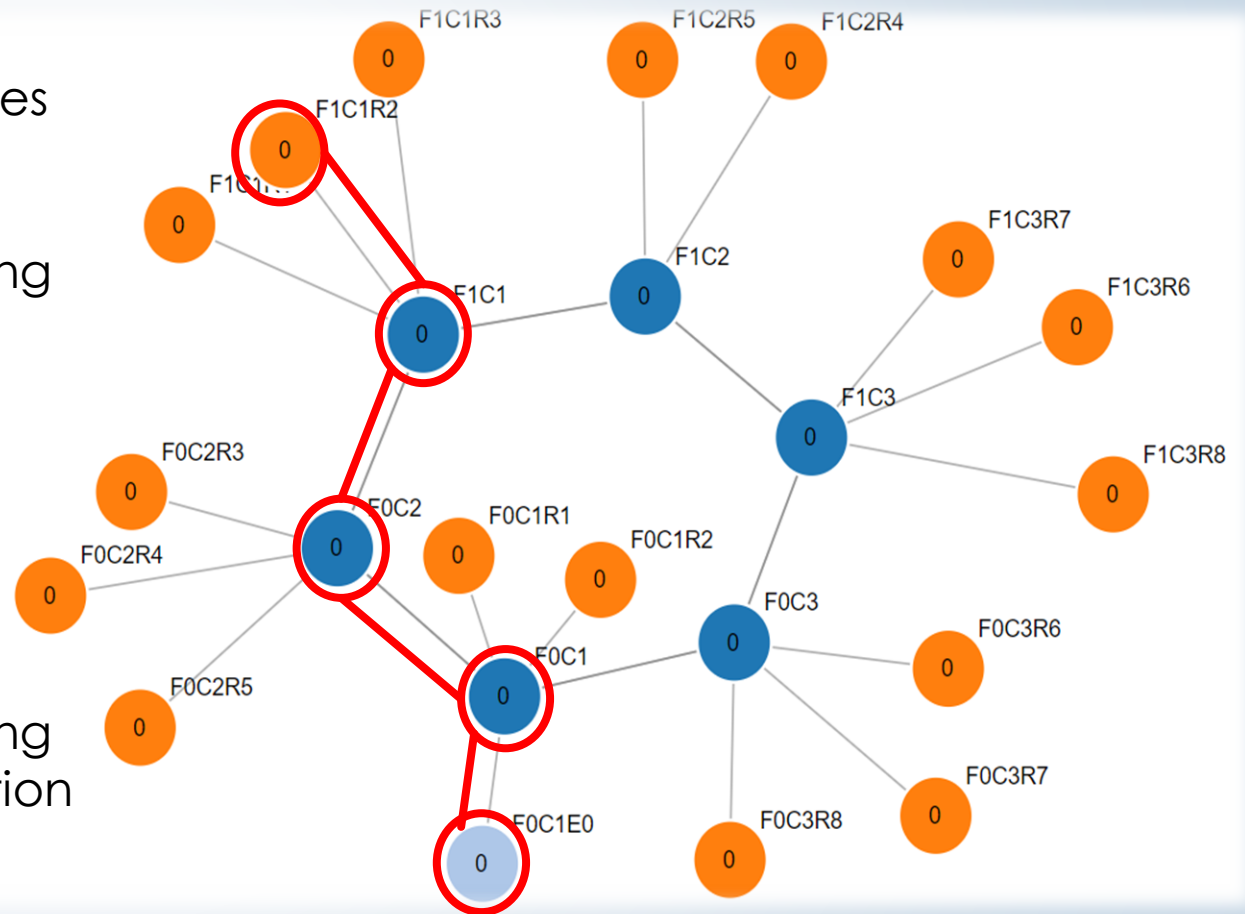


UNIVERSITÀ DEGLI STUDI
DI GENOVA

Movement Simulator



- ❖ To Enter:
 - ✓ Get Random number of Lectures
 - ✓ Get a random Entrance
 - ✓ Get a random Room
 - ✓ Calculate the shortest path using "Dijkstra"
 - ✓ Start moving according to the path and the position time.
- ❖ To exit :
 - ✓ Get a random Entrance
 - ✓ Calculate the shortest path using "Dijkstra" from the current position
 - ✓ Start moving according to the path and the position time.

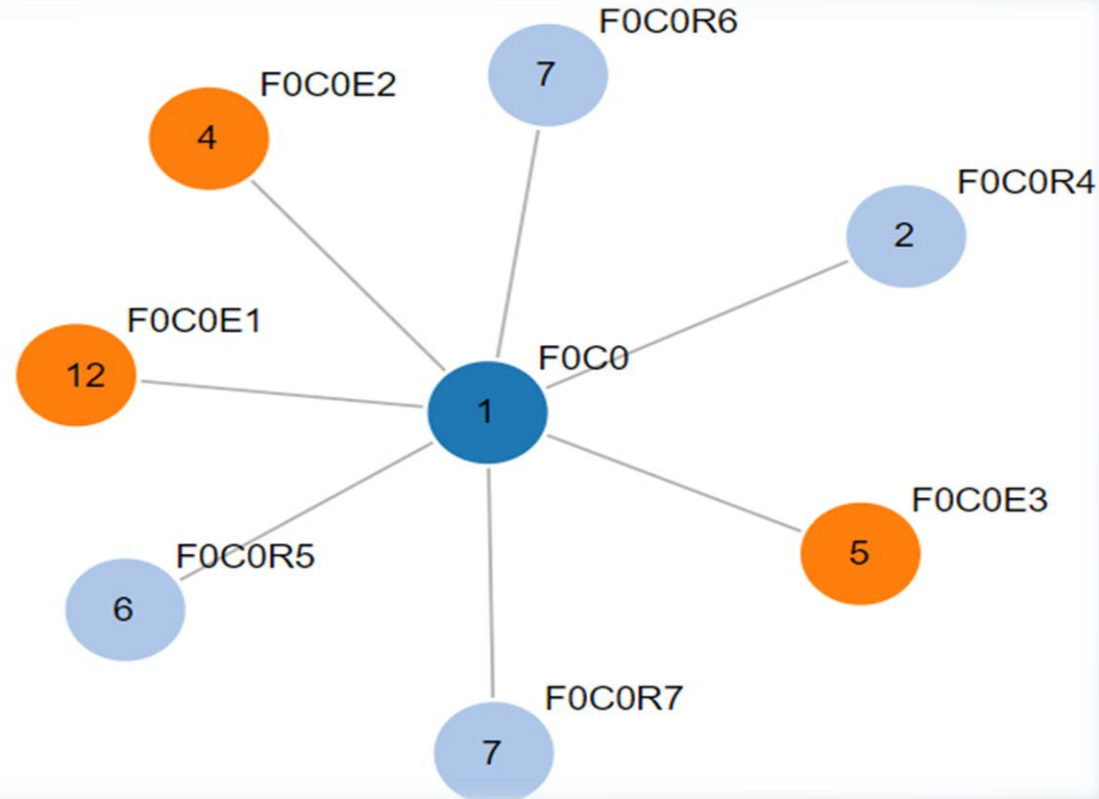




Movement Simulator

❖ BLE Simulator

While the simulator moving the people through the building , the BLE scanners will check the position of each person and count them for each BLE scanner in a specific time interval and send the data to the server.



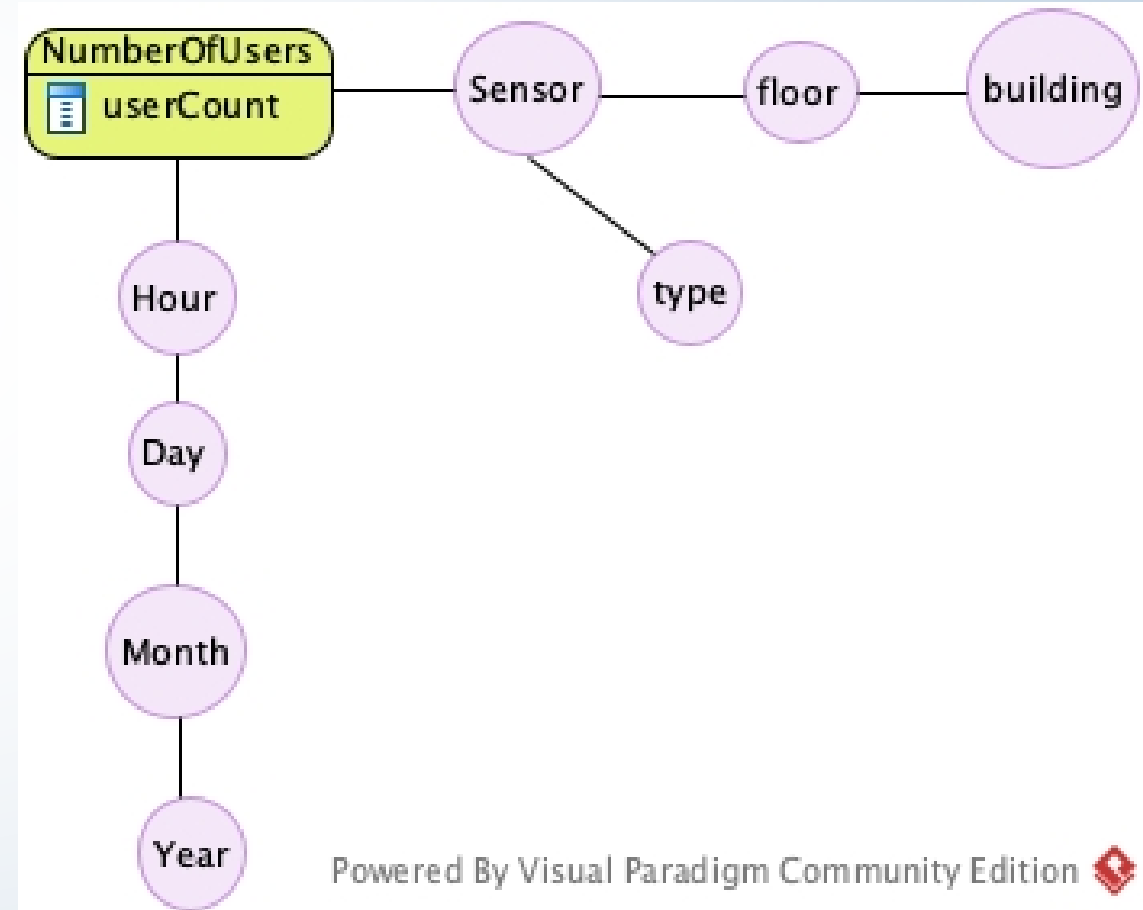


DW Conceptual Design



Workload

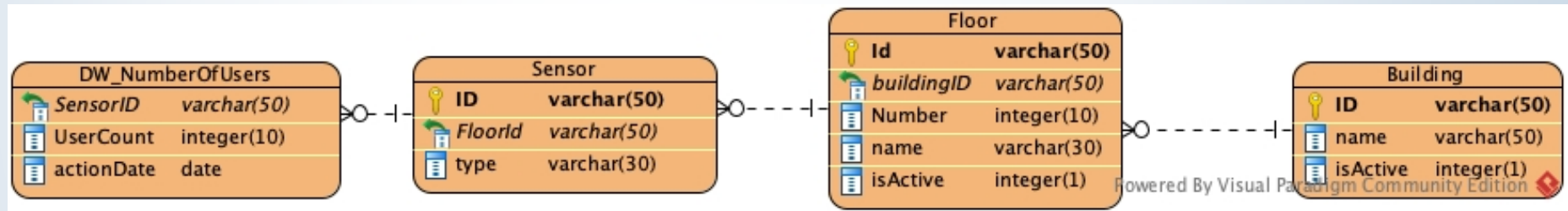
- Number of users who visited a specific building on specific date on hourly basis
- For each Room, entrance, corridor, floor and building the average number of users who was inside this entity on day hour, day week, day month, monthly, month year and yearly basis
- For each Room, entrance, corridor, floor and building the minimum and maximum number of users who was inside this entity on day hour, day week, day month, monthly, month year and yearly basis





UNIVERSITÀ DEGLI STUDI
DI GENOVA

DW Logical Design

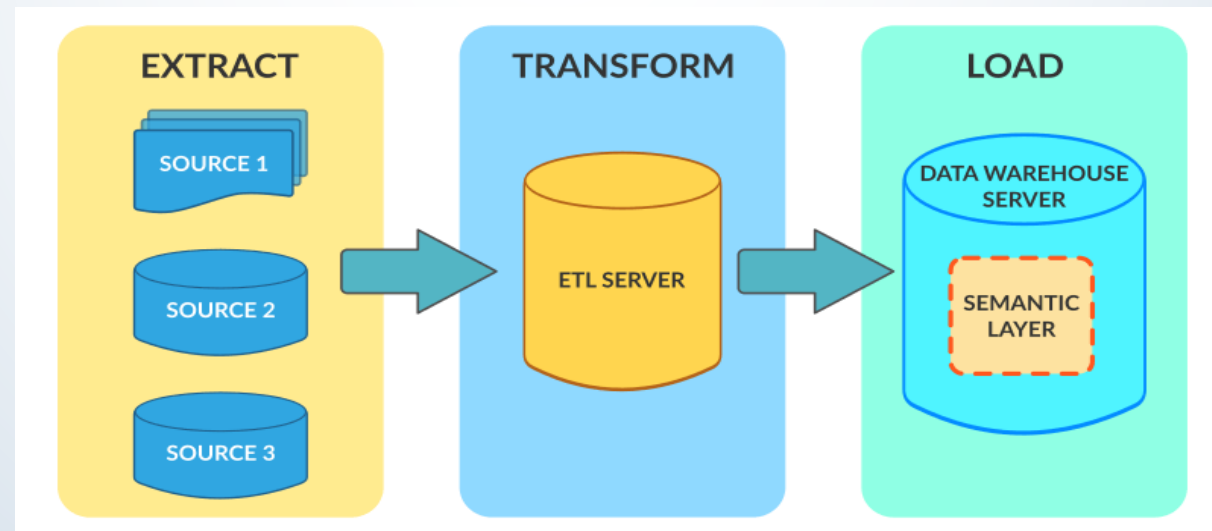




ETL Process

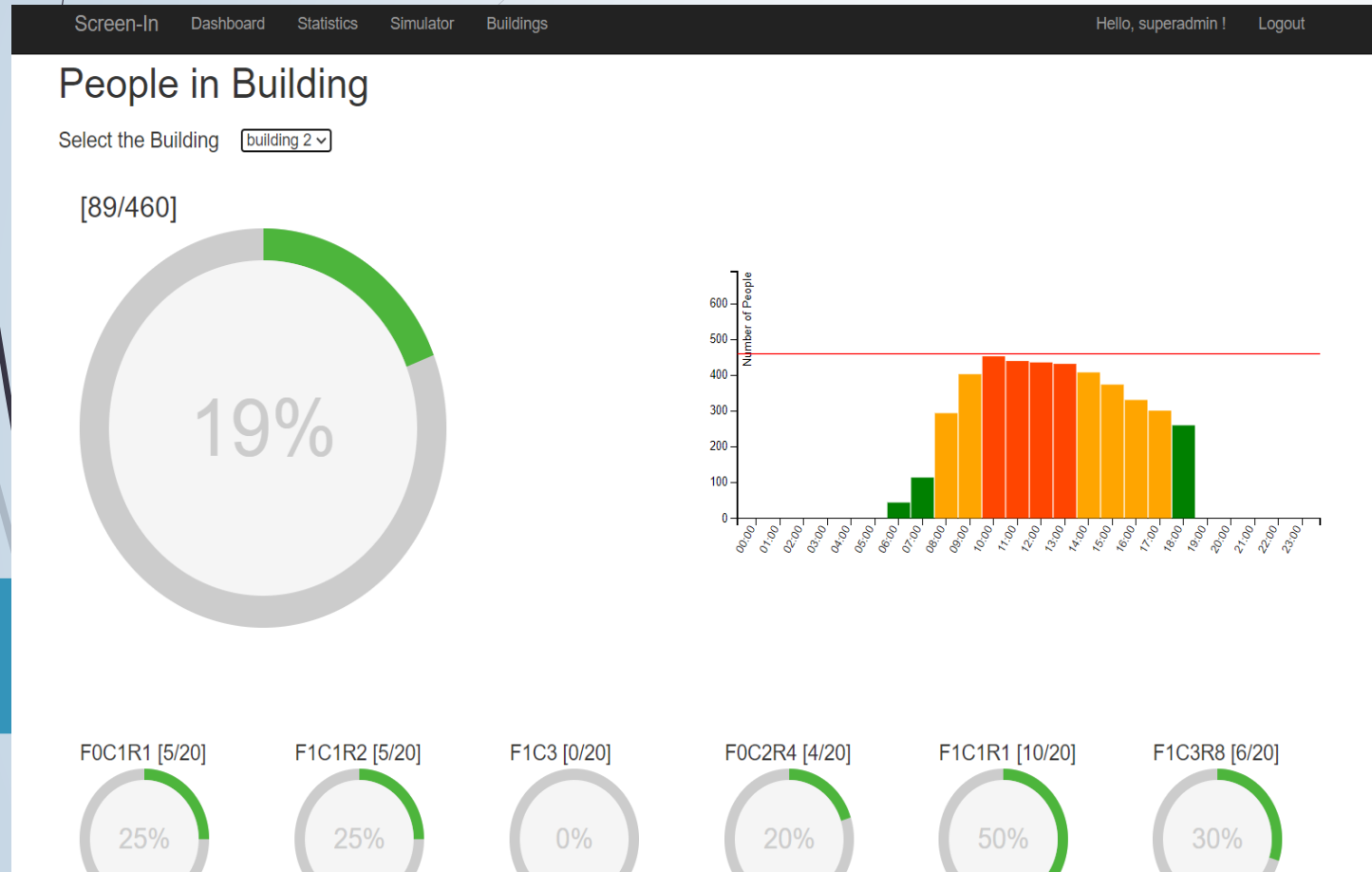


- ETL process work on loading the data from sensor to the created DW
- The process consists of two steps:
 - **DB-Procedure:** works on read the data from the suitable tables and insert the data into dimension tables and into data mart table
 - **Job:** a scheduled job that called on hourly basis to call the database procedure





Dashboard



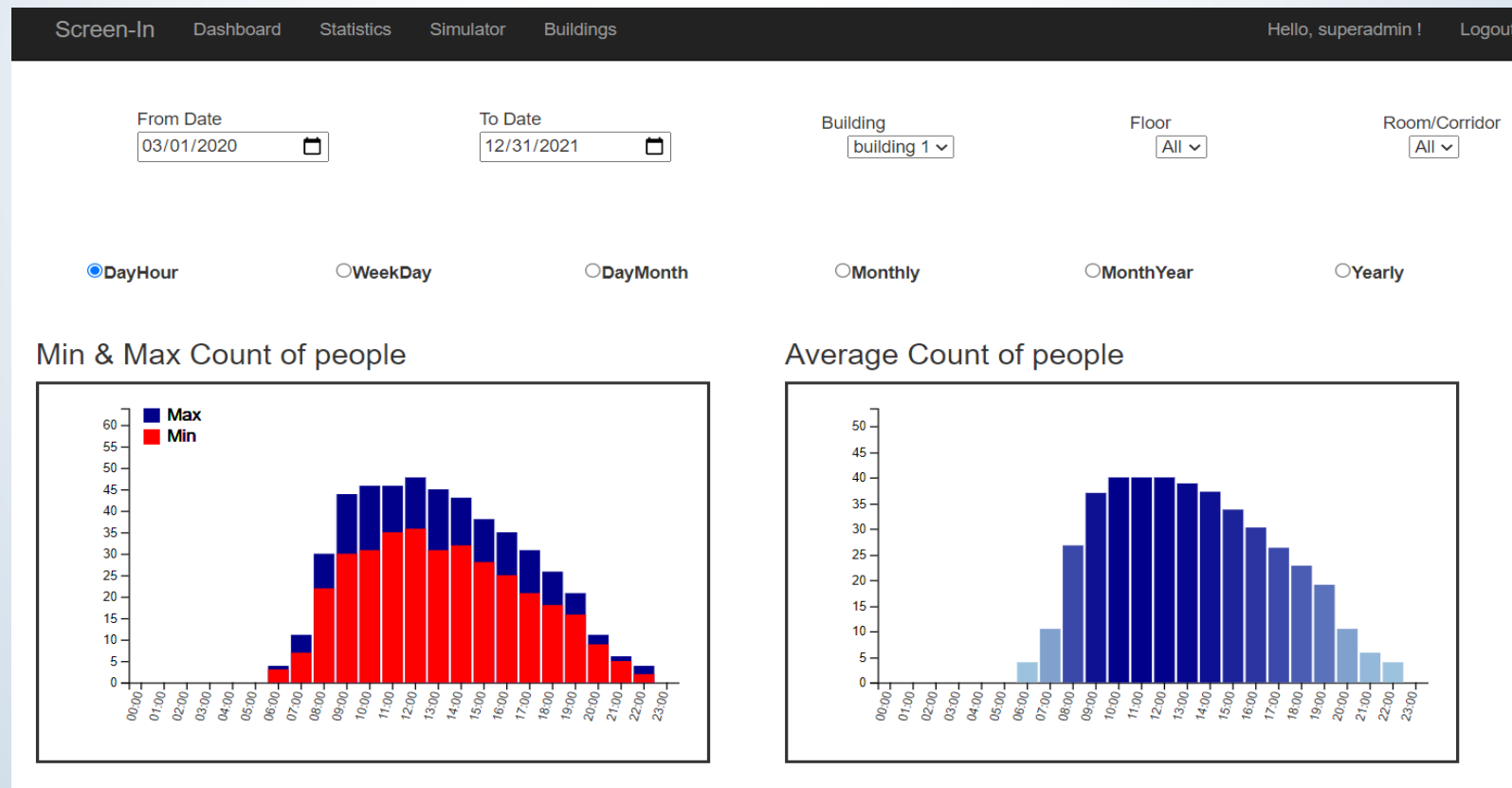
- ❖ To get a real time monitoring we designed and implemented an interface that show the following information:
1. The total count and percentage of people in a specific building
 2. The count and percentage of people in a specific building per sensor (Room, Corridor, Entrance)
 3. Bar chart of the people count in a specific building per hour in the current day



Statistics



- ❖ The application provide two types of statistical reports
 - The average number of people
 - The min max number of people
- And these reports can be generated over the building hierarchy
- Also, over many date formats (Hourly, Week_day, . . . , Yearly)





UNIVERSITÀ DEGLI STUDI
DI GENOVA

future work



Document Based NoSQL-DB (MONGODB)

- For More efficient performance we can change the log saving to be on NoSQL-DB
- Where we can save UserActions, AdminActions and SensorLogs
- Mongo DB have strong consistency
- Master-Slave replication protocol
- Hash based & Range Based Partitioning protocol

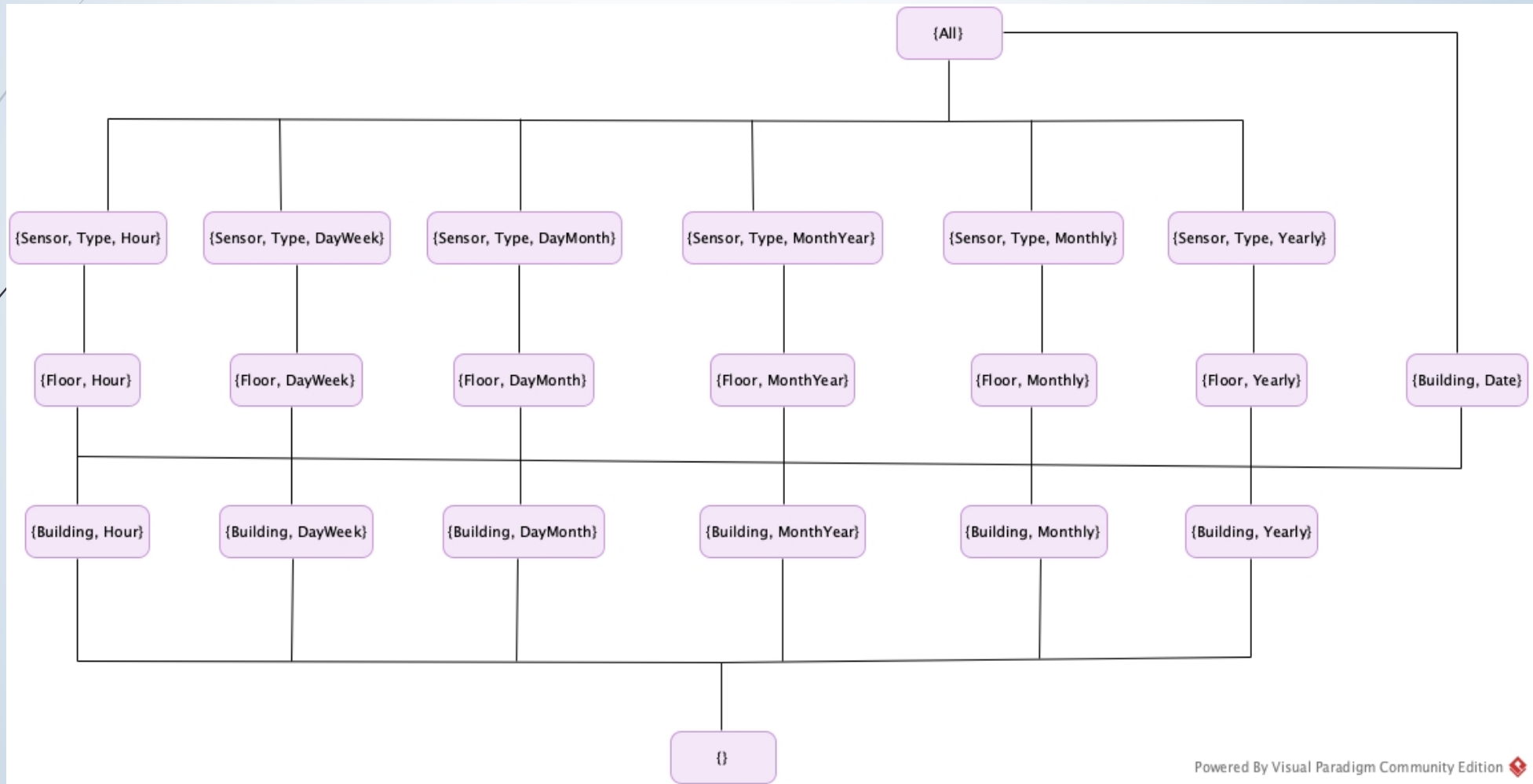




future work



Materialized View





UNIVERSITÀ DEGLI STUDI
DI GENOVA

future work



HIVE

- We can use a distributed file system like Hadoop which ensures that the data is partitioned and replicated
- An infrastructure built on top of Hadoop For providing data summarization, query and analysis
- Kind of data warehousing on top of Hadoop MapReduce
- Working on enabling the content of a folder in HDFS to be queried as a table through HIVEQL
- HIVEQL is SQL-like query statement





UNIVERSITÀ DEGLI STUDI
DI GENOVA



Demo