PROYECT CASAB

Sergio Andrés Sánchez Dueñas[1], Sebastián Nieto Ramirez[2],

[*]Systems and Computing Engineering, Technological University of Bolívar, Cartagena Bolívar

Abstract: This article details the composition of an access control software architecture (CASAB), which provides an authentication, authorization, and accounting service. The purpose is to provide secure handling of information, regarding the users who can access the system and the permissions that each one of them has; in addition to the use of digital certificates to prove the identity of a user or equipment that executes any of the actions that can be carried out in the system.

1. Introduction

This article details the composition of an access control software architecture (CASAB), which provides an authentication, authorization and accounting service. The purpose is to provide secure handling of information, regarding the users who can access the system and the permissions that each one of them has; in addition to the use of digital certificates to prove the identity of a user or equipment that executes any of the actions that can be carried out in the system.

2. Description of the requirements

a. Functional requirements

- Record the history of accesses to the system made by the user
- Implement an architecture in microservices.
- Guarantee access to the system with double security verification.
- Control user access to system information from an administrator view.
- Connect to the system with the Exia API to obtain the data.
- Analyze the data and create alerts when anomalies are registered in the data.
- Visualize data information through graphs with specific values (day, hour, minute, week, month), in tables with numerical values and time lines.

b. non-functional requirements

- All external communications between the database, the web page and the client must be encrypted using the RSA algorithm.
- Descriptive and prescriptive performance analysis.

c. System Requirements

- Deploy the web page.
- Model a Service-oriented system.
- The FE must have Responsive features.
- Backend development will be done with Django in Python 3.
- Frontend development will be with react.js.
- The system must be scalable.

3. User Roles

All those who through the main endpoint of the application access the main view of said application will be able to access data stored in the database, more, however, there will be data and actions that can only be reviewed and/or modified/deleted/created by certain users, which they have received from a user whose role is “Admin”.

3.1 Admin: This is the one that assigns the role to each new user that registers in the application, this one has permissions to be able to create, modify and/or delete data from the database in case any policy that has been imposed for this has been violated. application, or in any case a user with another user with another role has requested it and said request is accepted. The following sequence diagram details how the “Admin” role user assigns roles to users who do not have the same role:

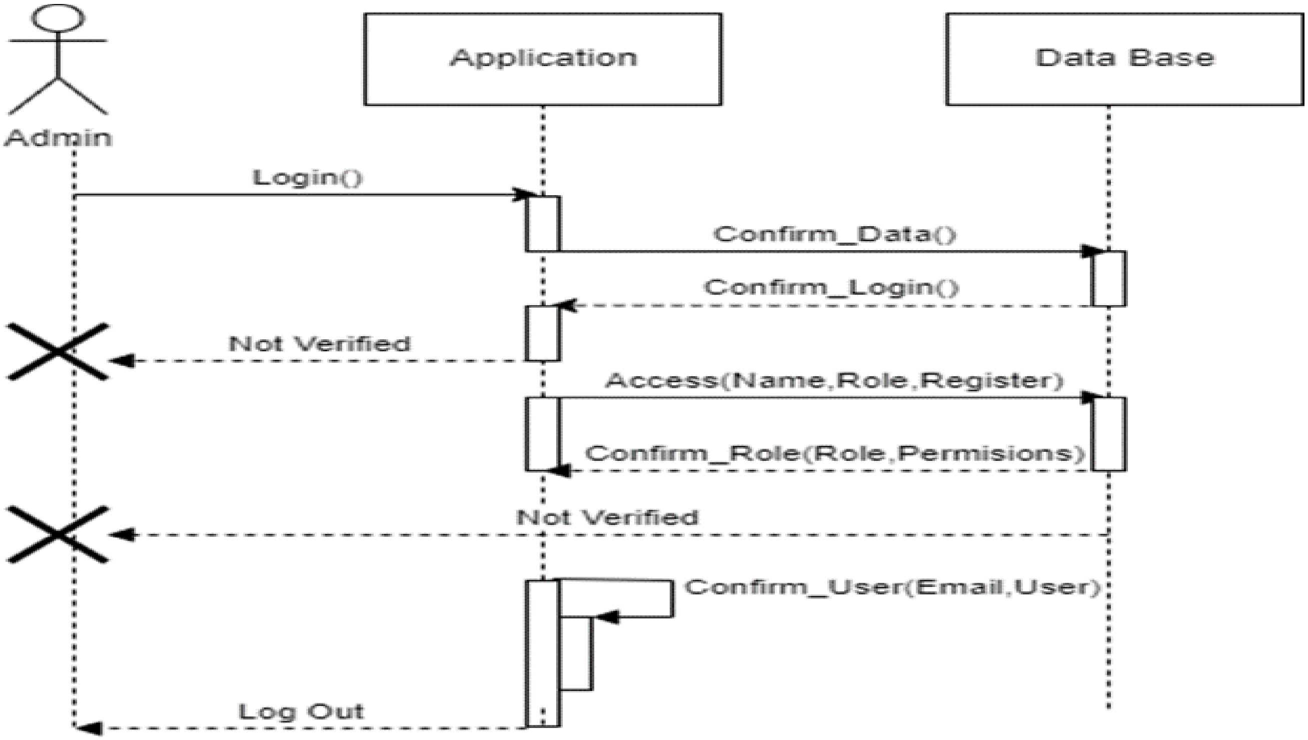


Figure 1. User role admin, which can assign roles to those who do not have the same role.

3.2 User_Generator: This role allows a user to generate new energy resource offers, in these offers you can specify the amount of energy resources offered, as well as the price of said offers.

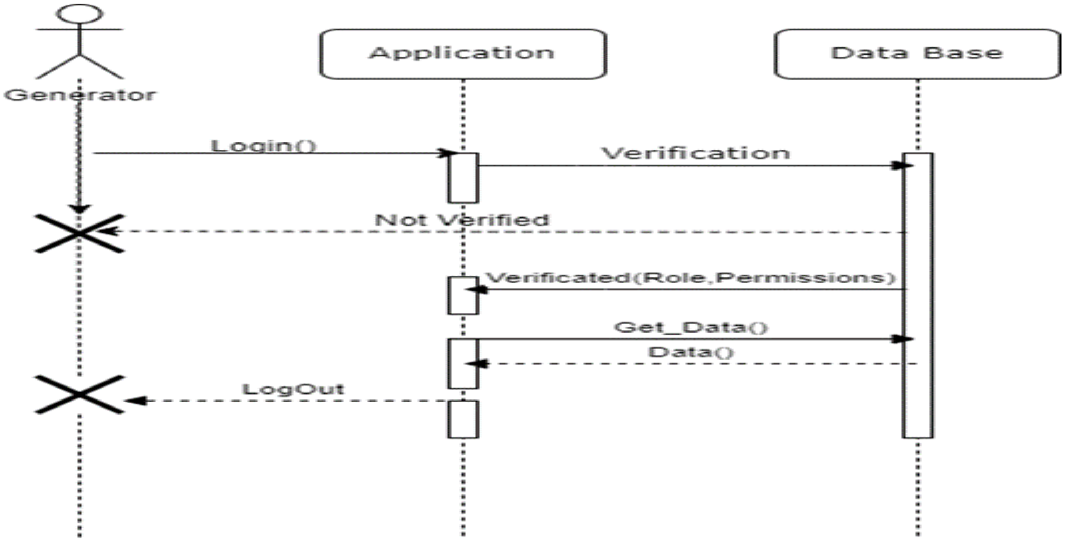


Figure 2. It describes how a user with the generator role accesses the data to which this user has permission to access them.

3.3 User_Consumer: This role allows a user to view the offers that are made by another user whose role can be: User_Generator or Prosumer. This user can read more about the offer made by another user and make the purchase of said offer.

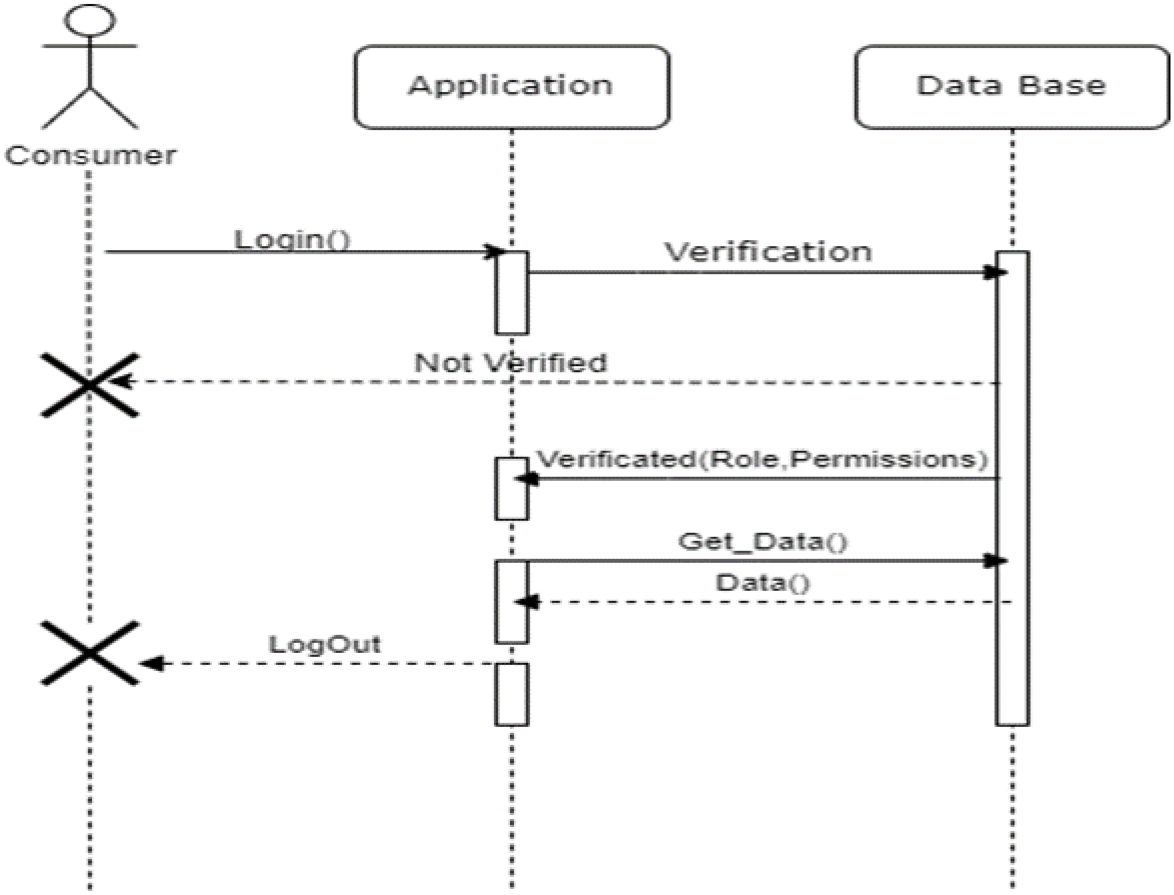


Figure 3. The process with which a user can access the data that this same user has permissions to access is described.

3.4 User_prosumer: This role allows the user to see the amounts of resources that he generates, as well as the amount of resources that he buys, in which case this user is in a situation in which he does not generate enough to be able to cover the demand that his activities require. sales.

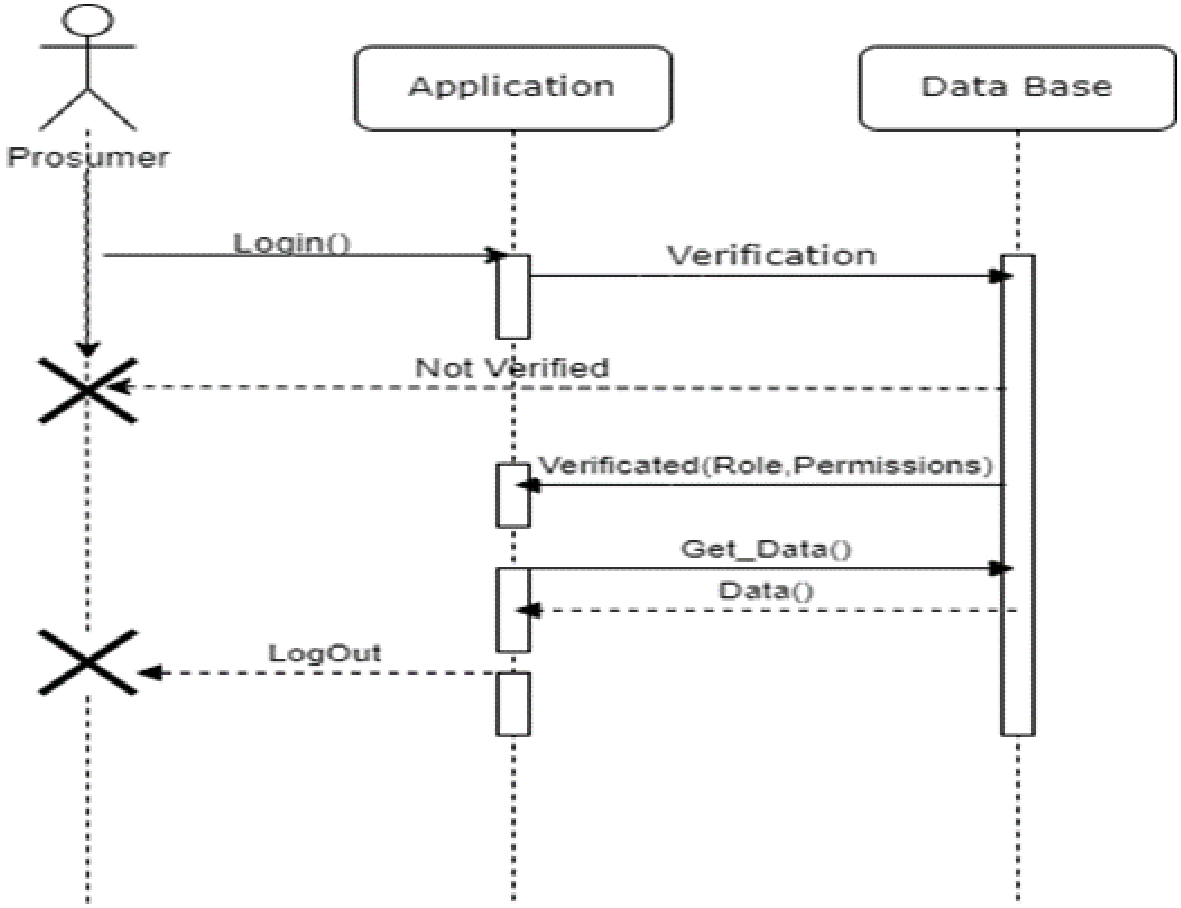


Figure 4. The process with which a user with the prosumer role can access the data that the same user has permissions to access is described.

3.5 Invited: This role allows the user to only see the offers of users with the role "User_Generator" and "User_Prosumer", this is the only action that a user can perform with this role. The following diagram describes how a user with this role can request a different role than the one he already has and different from the admin role.

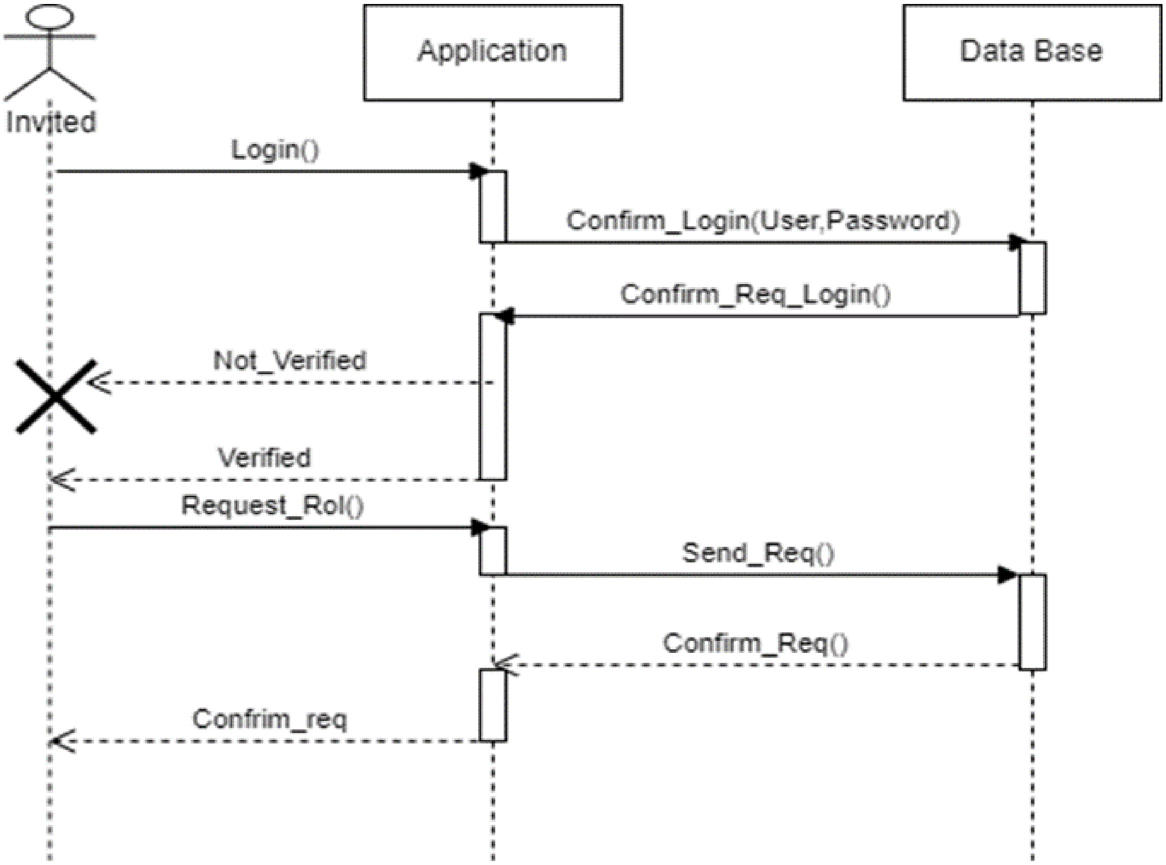


Figure 5. Invited role user, who can only request a different role than the one he already has and the admin role.

3.6 Specific Permissions Table

The following table shows in a detailed and simple way the permissions that each user has in the system.

Summary Privilege	Default Role				
	consumers	Generator	prosumer	invited	admin
Consume API data	●	●	●		●
Request New Role	●	●	●	●	
Create new date		●			●
Validate Role					●

Table 1. Description of the permissions that each type of user can have.

4. Composition of Architecture

4.1 Architecture goals

This architecture seeks to detail the different processes that must be carried out to perform secure access to any application, whether web or mobile, in the particular case of this project, secure access will be made for those users who wish to see detailed information of the solar parks of which they (the users) comply with the filter in order to have access to said data.

4.2 Architecture considerations

With the implementation of this architecture, the aim is to generate a system with a high level of security which offers users the security that their information, as well as that of each and every one of the transactions they carry out, maintain its veracity at any time. In addition to offering them an easy-to-use alternative that meets what is required when keeping a record of production and consumption.

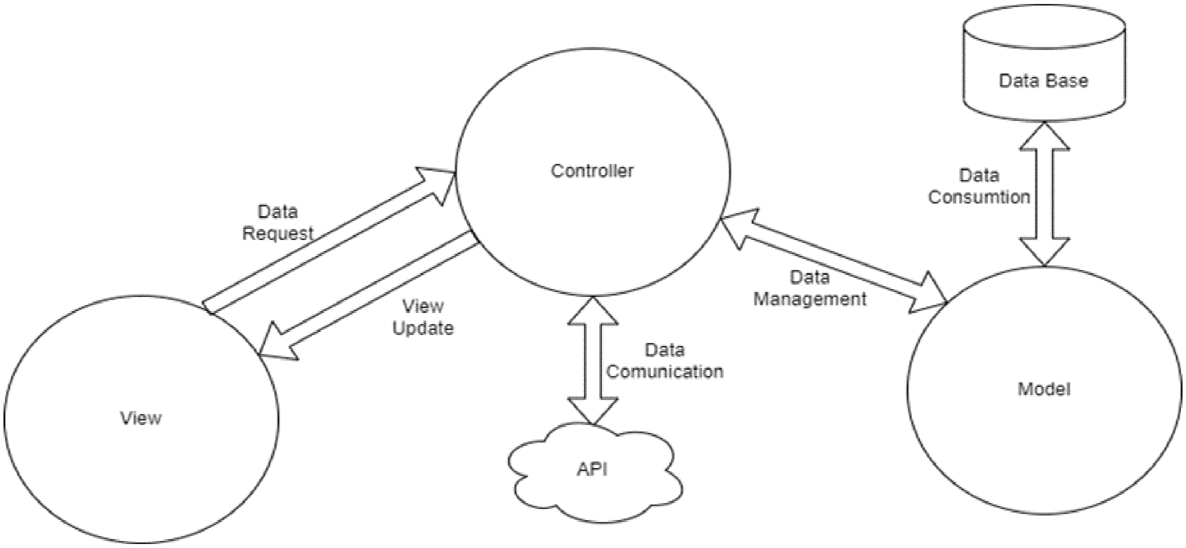


Figure 6. Vista-Controller software architecture model, which presents a slight variation taking into account the characteristic of the CASAB project

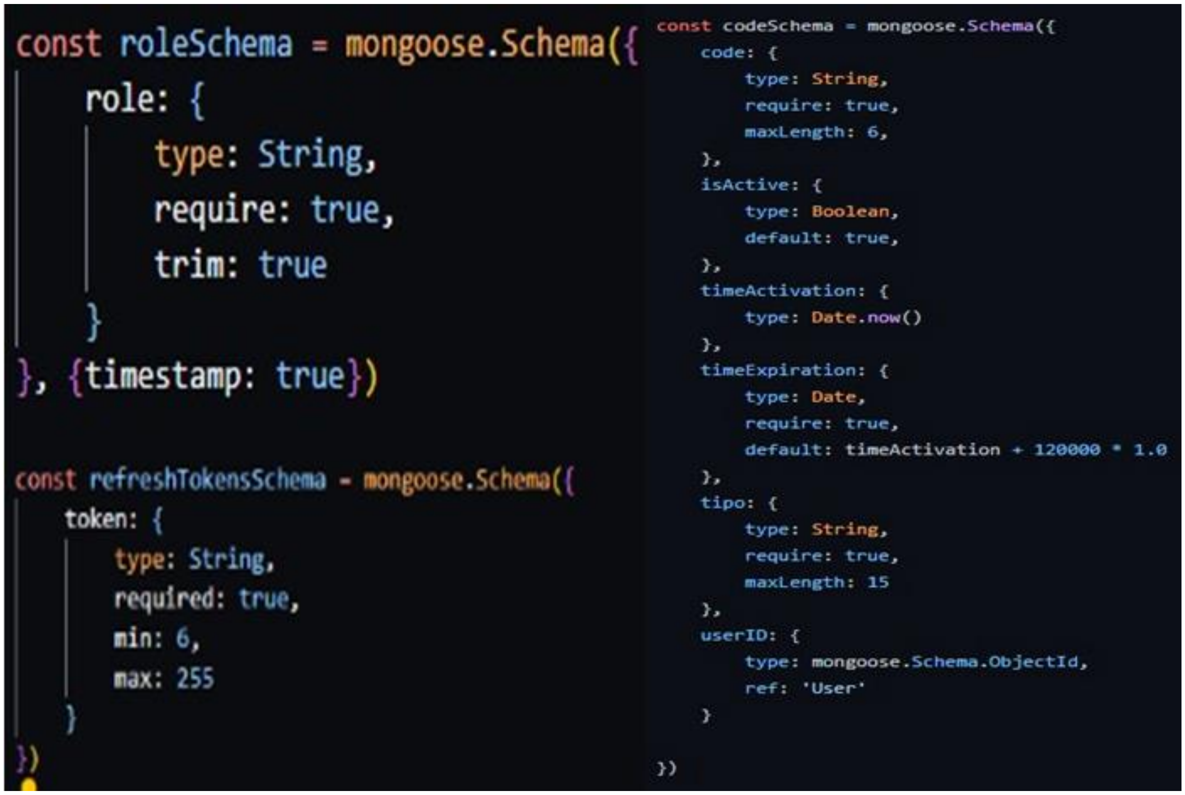
5. System Description

	Language	**note**
Programming Language (Frontend)	**JavaScript**	It is an interpreted programming language, whose use is mainly on the web, both on the server and user side.
Framework (Frontend)	**react**	Library written in javascript, created to facilitate the creation of components that can interact with each other and that are reusable.
Programming Language (Backend)	**Javascript**	It is a one of the most used programming languages of web development due to its characteristics.
Framework (Backend)	**Node js**	This is a framework is a runtime environment for server-side web development.
Database	**Mongodb**	It is a non-relational database management system.

Data Analytics	**python**	The data analysis will be done in Python, with its "Pandas" library, which will allow a clear and excellent analysis of the data that is supplied from the solar parks.

Table 2. Description of the system describes how each block will be developed in the CASAB access control system.

6. Database structure



```
const roleSchema = mongoose.Schema({
    role: {
        type: String,
        require: true,
        trim: true
    }
}, {timestamp: true})

const refreshTokensSchema = mongoose.Schema({
    token: {
        type: String,
        required: true,
        min: 6,
        max: 255
    }
})
```

```
const codeSchema = mongoose.Schema({
    code: {
        type: String,
        require: true,
        maxLength: 6,
    },
    isActive: {
        type: Boolean,
        default: true,
    },
    timeActivation: {
        type: Date.now()
    },
    timeExpiration: {
        type: Date,
        require: true,
        default: timeActivation + 120000 * 1.0
    },
    tipo: {
        type: String,
        require: true,
        maxLength: 15
    },
    userID: {
        type: mongoose.Schema.ObjectId,
        ref: 'User'
    }

})
```

Figure 7. Diagram of a database which represents how it is structured and connected to the different tables that conform it, all this making use of the MySQL workbench environment.

6.1 DataBase Description

The database used is mongodb, which is a non-relational database. For this project this type of database manager was necessary because the type of data that will be stored from the users will not be complex, so a relational database will not be necessary. This database is composed of the following schemas:

- CodeScheme: Contains all the information of the codes that are used for user verification.
- RefreshTokenScheme: Contains the refresh token, which is used to validate the user's identity at login and validate the expiration of the token.
- RoleSchema: Contains the information of the roles that are being managed in this project, which were described throughout this document (See Section 3.).

7. Security

Security is one of the main features in this project. In this section, the strategies that will be used to guarantee secure access will be announced, as well as the ability to detect possible intruders in the system. Next, the security requirements in the CASAB project will be described:

- Guaranteeing Confidentiality in this project is important, since it is important that the user's credentials are only known by the owner and that under no circumstances be known by a third party.
- Given the context for which this project is created and its requirements, authentication takes on great relevance, since it is necessary to guarantee that each user is who they say they are and thus avoid impersonation which could generate a leak of sensitive data. of system users.
- Each of the users in the system receives a role, each of these roles have certain permissions, so the Authorization feature becomes relevant, since it is necessary to guarantee that each user has access to the resources to which they are authorized.
- The data that is entered in the database must be fully secured, as well as these same data must be backed up, the password being the most sensitive data for the login must be encrypted before being saved in the database. Given all this, data storage security is an important aspect in this project.
- One of the important processes at the time of starting up the system is Accounting, which refers to the control of the different accesses of the users, to detect any unusual behavior.

8. Schemes

Sequence diagrams

8.1 Login.

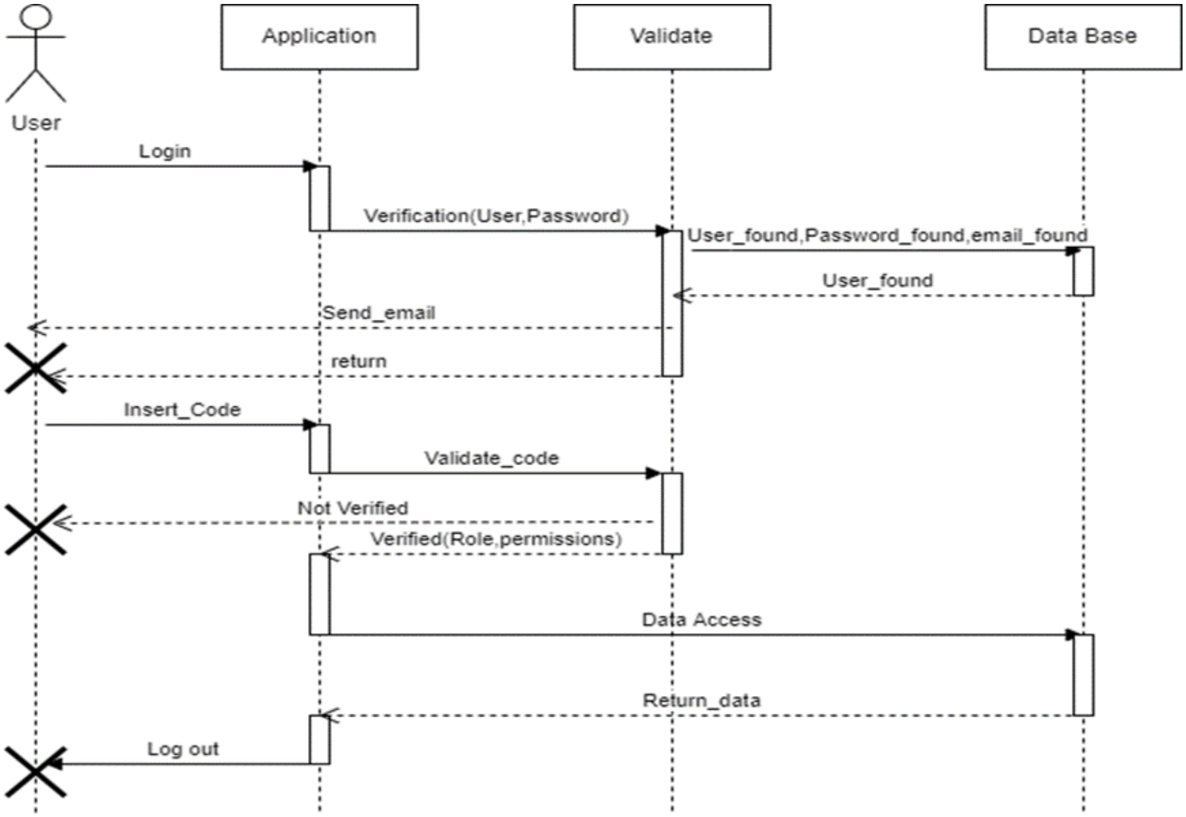


Figure 7: To maintain a high security standard, a 2-step log-in will be generated, through which a user will be able to enter the application as long as their entry has been verified by means of an email sent to them, thanks to the information previously provided by the user.

8.2 Password Recovery

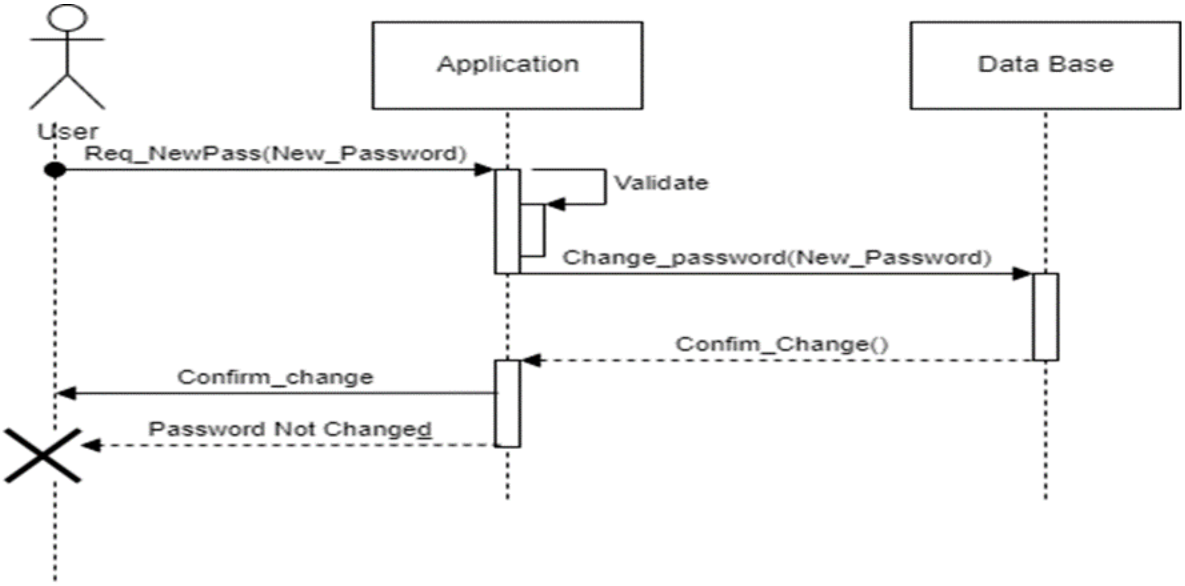


Figure 8. The process that is carried out to be able to recover the password is described.

8.3 Use case diagrams

8.3.1 Admin

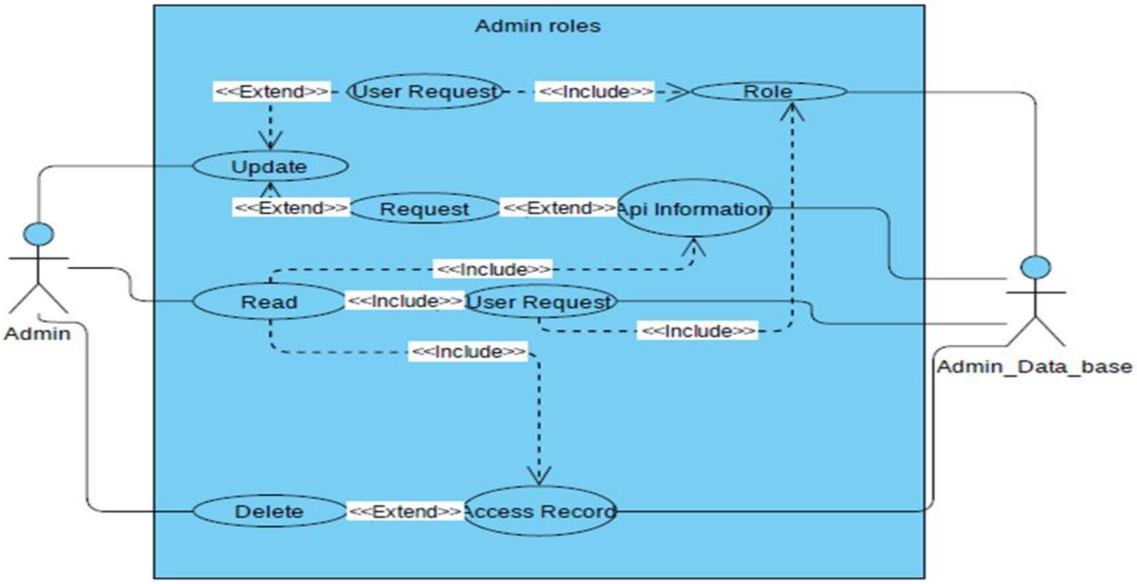


Figure 9. Use case diagram of the Admin role, where the activities that this user can perform are described, such as validating data entered by users, assigning roles and granting certain permissions.

8.3.2 User_consumer

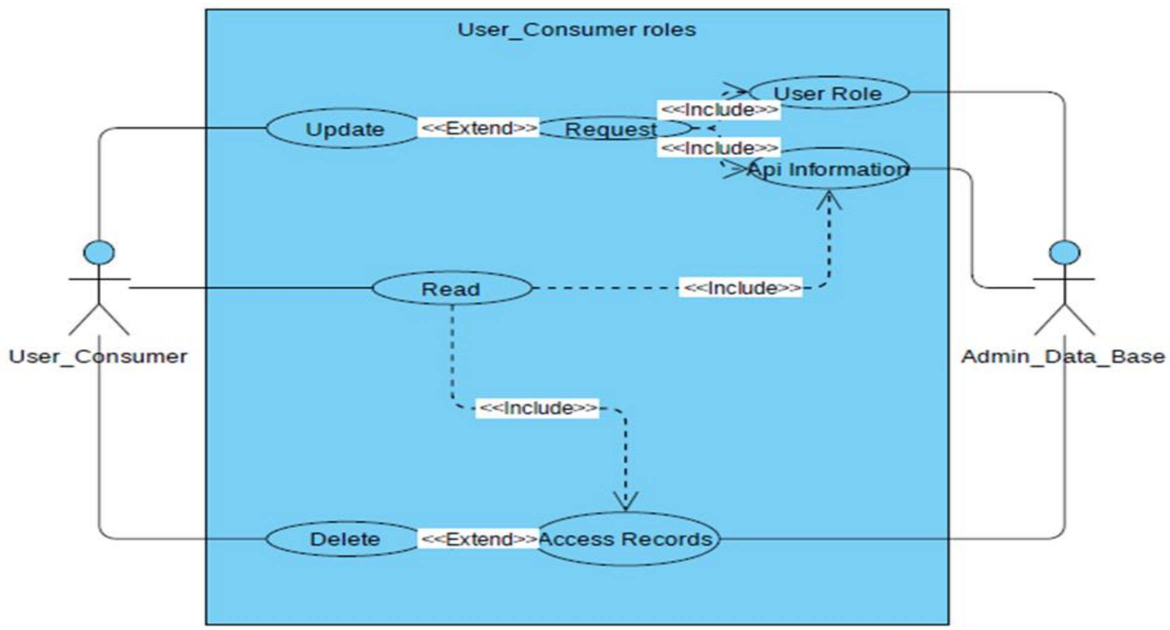


Figure 10. Use case diagram of the User_Consumer role, where the activities that this user can perform are described, such as requesting a new role, viewing API information or deleting access logs.

8.3.3 User_Generator.

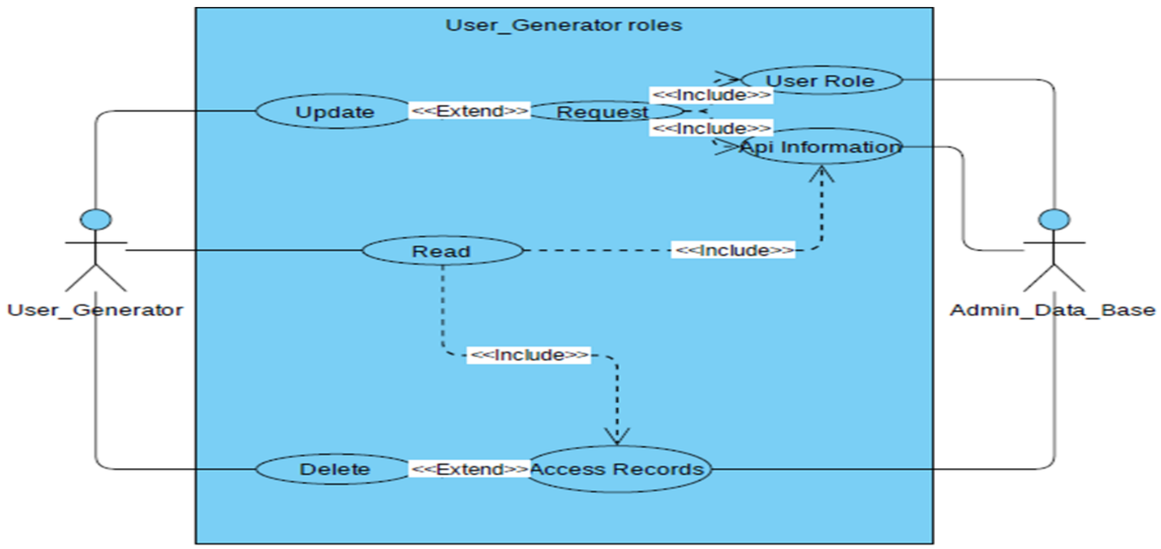


Figure 11. Use case diagram of the User_Generator role, where the activities that this user can perform are described, such as requesting a new role, viewing API information or deleting access logs.

8.3.4 User_Prosumer

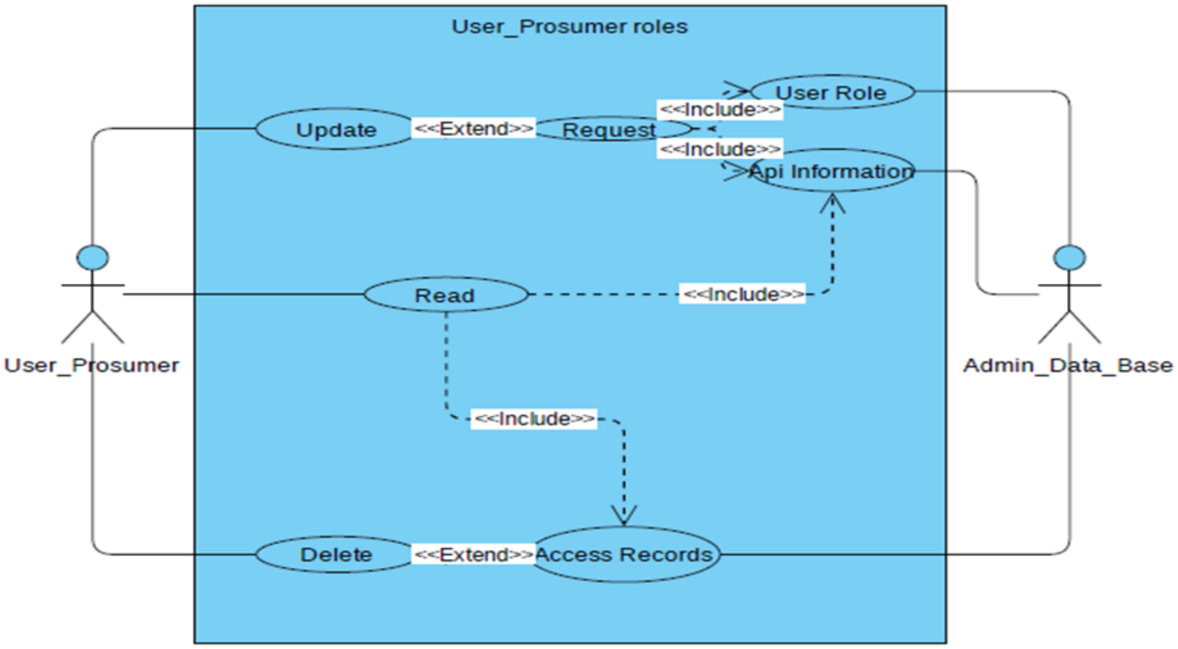


Figure 12. Use case diagram of the User_Prosumer role, where the activities that this user can perform are described, such as requesting a new role, viewing API information or deleting access logs.

8.3.5 Invited

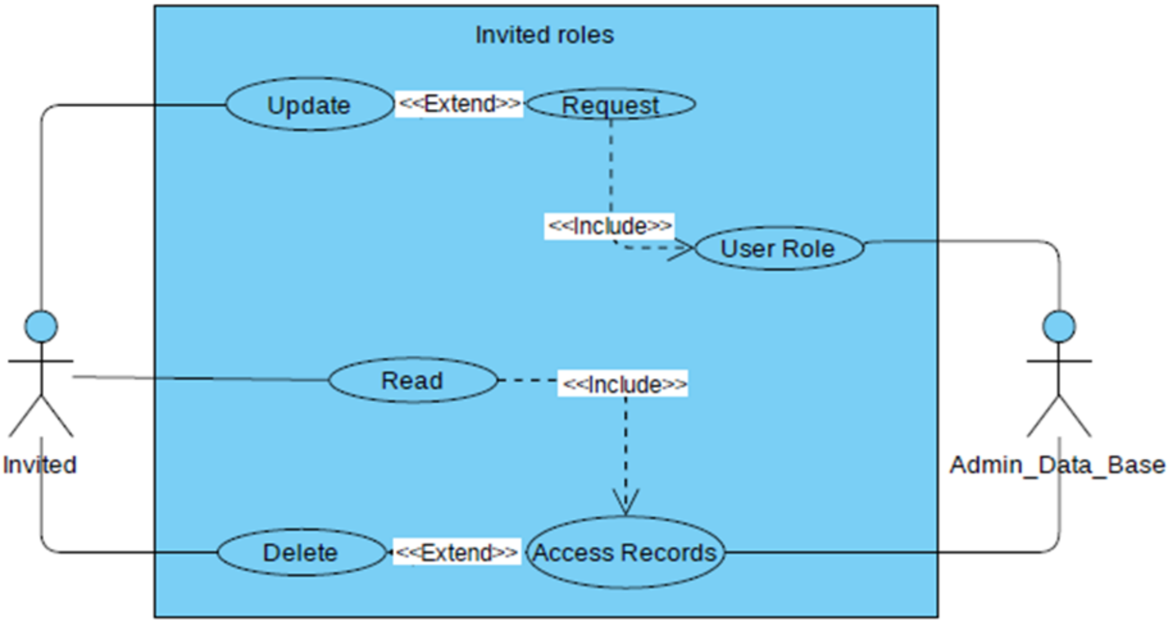


Figure 12. Use case diagram for the Invited role, describing the activities that this user can perform, such as requesting a new role, viewing api information, or deleting access logs.

8.4 Deployment diagram

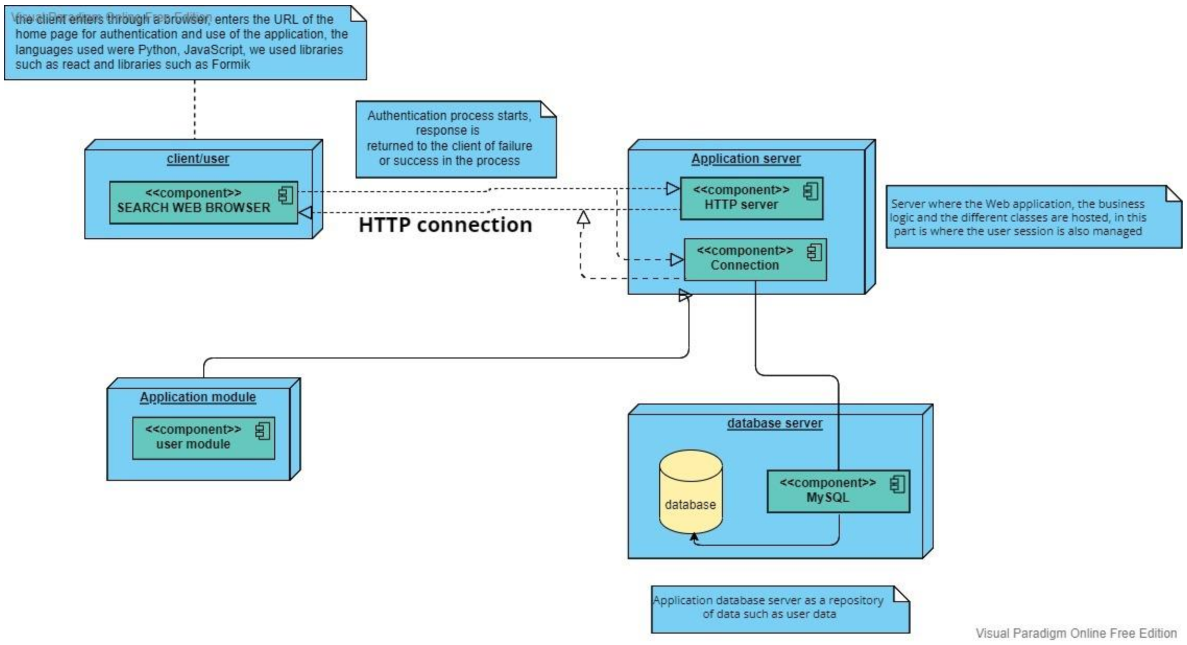

Visual Paradigm Online Free Edition
the client enters through a browser, enters the URL of the home page for authentication and use of the application, the languages used were Python, JavaScript, we used libraries such as react and libraries such as Formik
client/user
<<component>>
SEARCH WEB BROWSER
Authentication process starts, response is returned to the client of failure or success in the process
HTTP connection
Application server
<<component>>
HTTP server
<<component>>
Connection
Server where the Web application, the business logic and the different classes are hosted, in this part is where the user session is also managed
Application module
<<component>>
user module
database server
database
<<component>>
MySQL
Application database server as a repository of data such as user data
Visual Paradigm Online Free Edition

Figure 13. System deployment diagram in each of its deployment and implementation phases.

8.5 Class Diagram

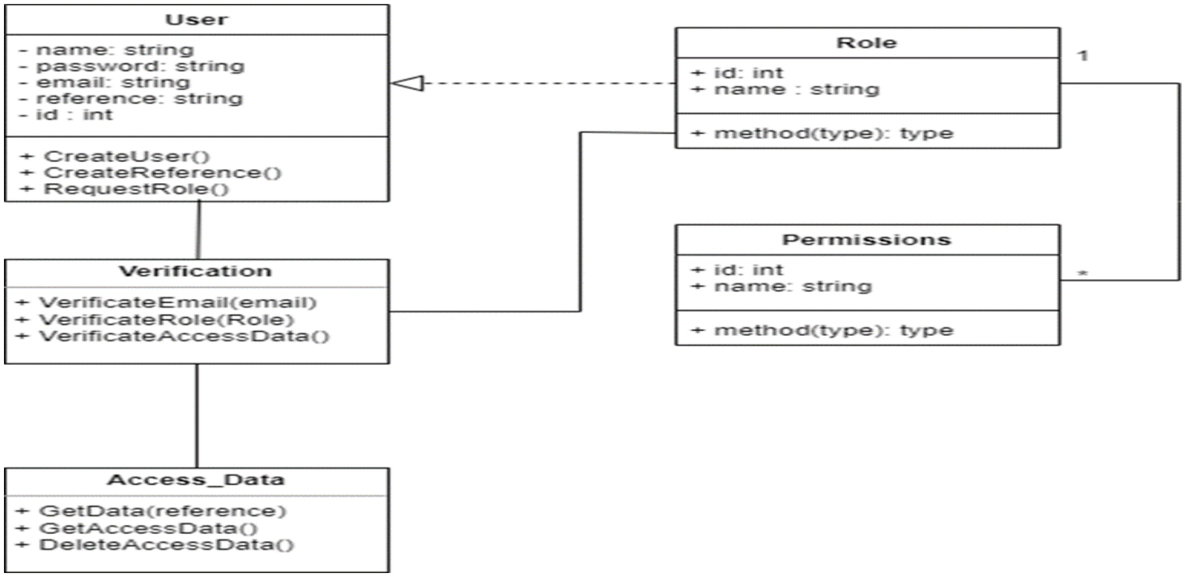
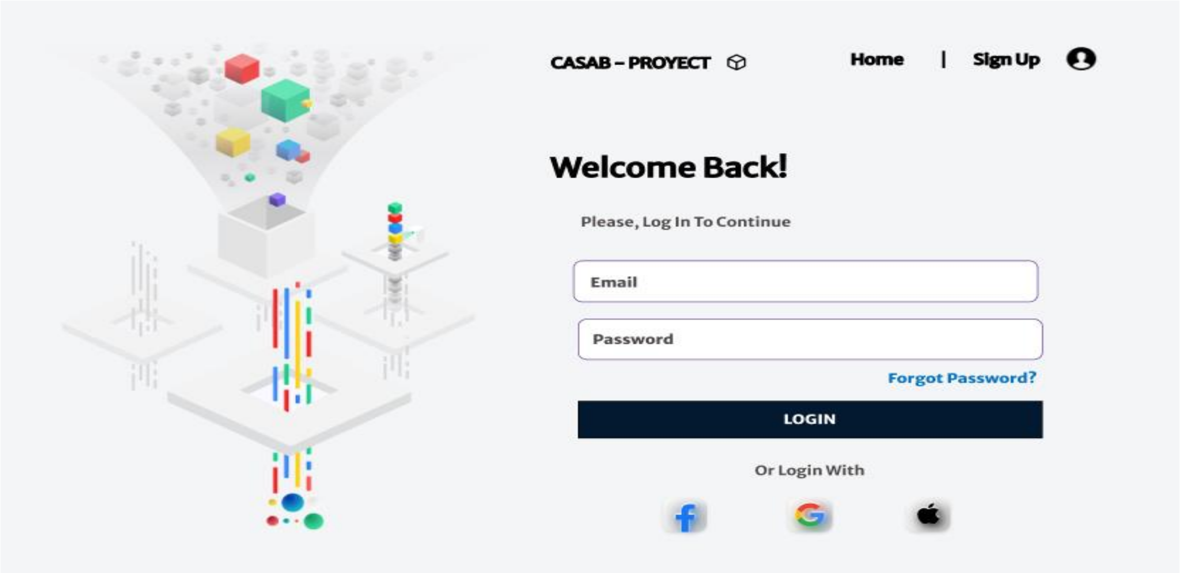


Figure 14. Class diagram which describes how the application is abstractly structured, in the same way the scheme that the application will have is described for developers.

9. User Interface Design

9.1 Log in view

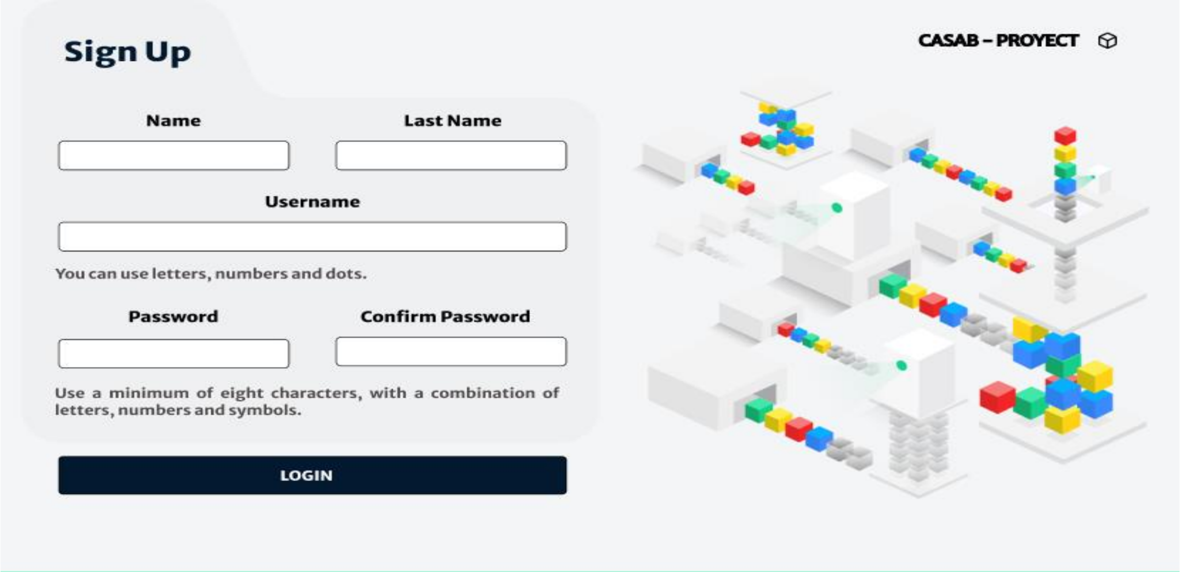


Welcome Back!

Please, Log In To Continue

Figure 15. After the user has registered, he will be able to access the application through this view.

9.2 Sign-up view



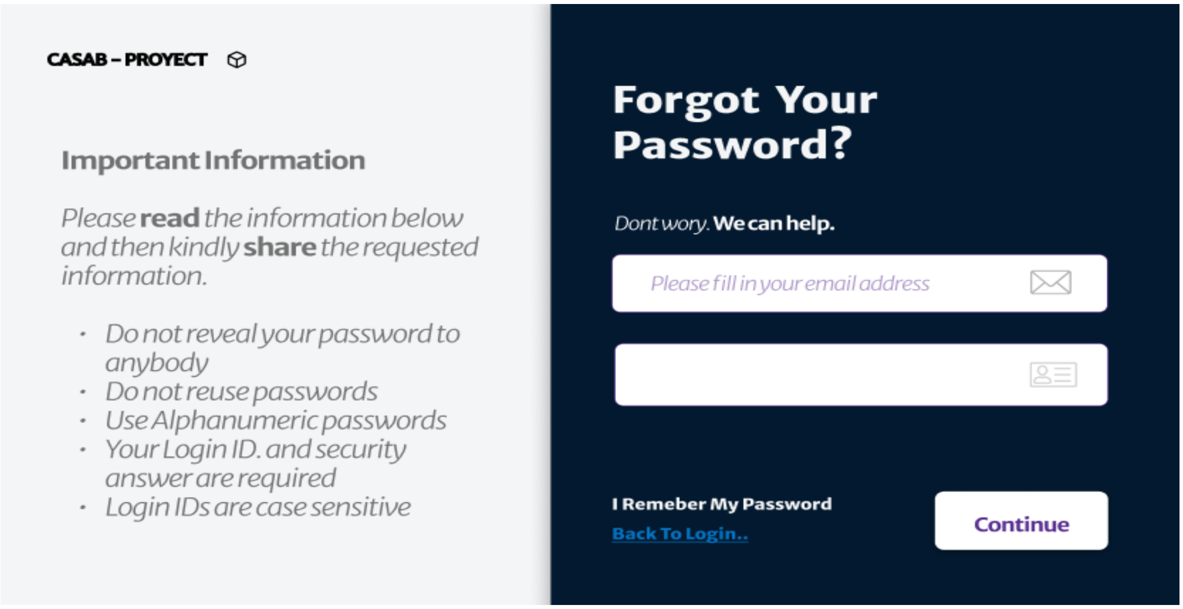
You can use letters, numbers and dots.

Use a minimum of eight characters, with a combination of letters, numbers and symbols.

Figure 16. In order to enter the application, the user must fill out the form that is presented, register, and after this they can go to the login view (Figure 15).

9.3 Password Recovery View

9.3.1 Enter email view



CASAB – PROYECT

Important Information

*Please **read** the information below and then kindly **share** the requested information.*

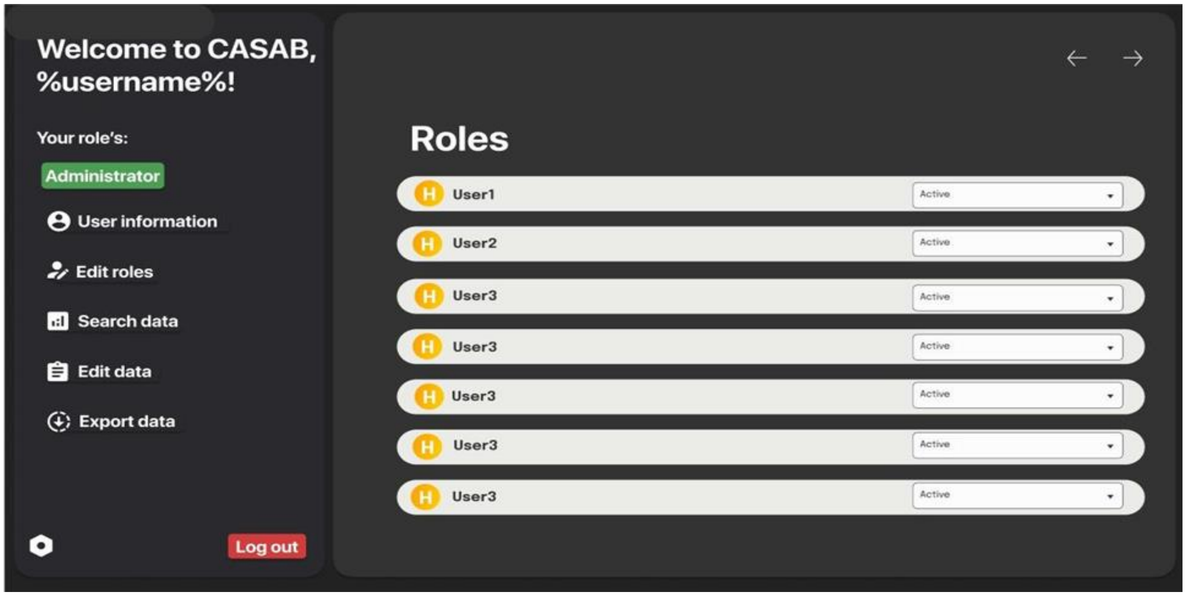
- *Do not reveal your password to anybody*
- *Do not reuse passwords*
- *Use Alphanumeric passwords*
- *Your Login ID. and security answer are required*
- *Login IDs are case sensitive*

Forgot Your Password?

Dont wory. **We can help.**

Figure 17. In the event that a user has forgotten their password, this will be the first view they see, where they will enter the email where they will receive the code that they must enter later (Figure 18), this is done as an authentication measure.

9.4 View "User Role Administrator"



Welcome to CASAB, %username%!

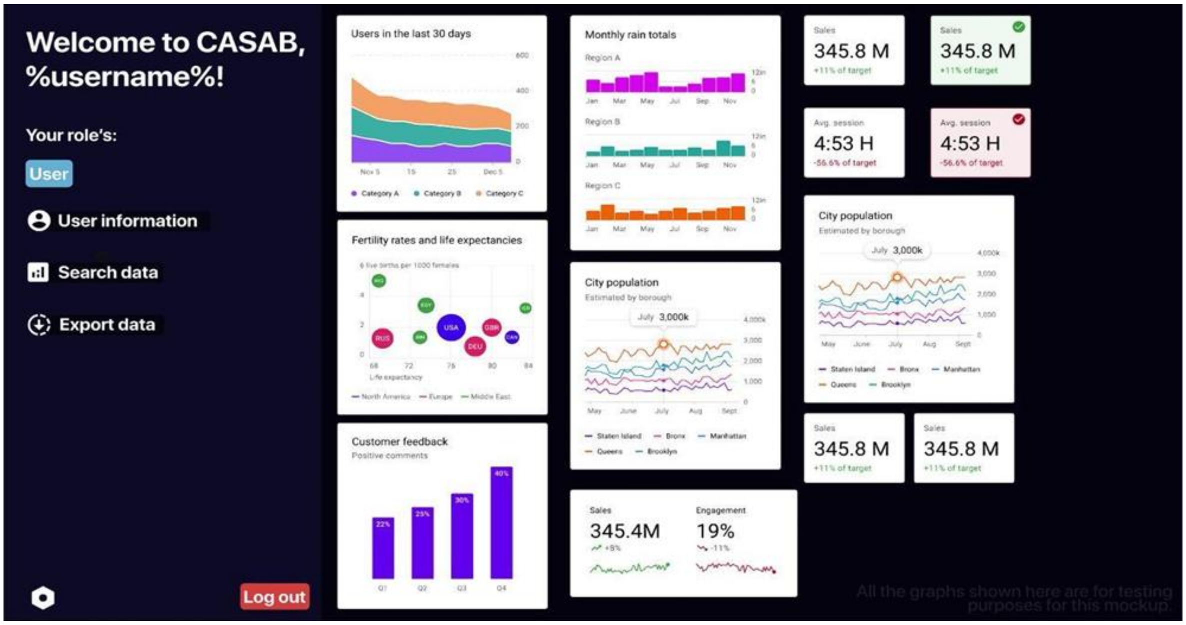
Your role's: Administrator

Roles

User1	Active
User2	Active
User3	Active
User3	Active
User3	Active
User3	Active
User3	Active

Figure 20. The view of the "admin" role user is presented, which can grant roles to other users who do not have the same role.

9.5 “User Role Producer/Consumer/Prosumer” view



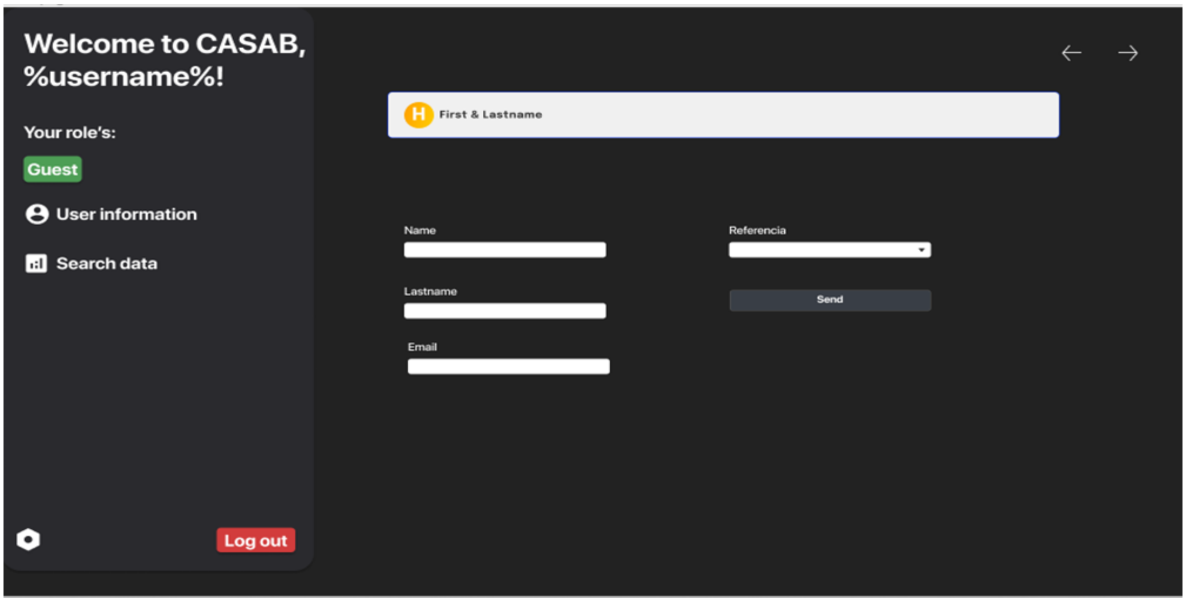
Welcome to CASAB, %username%!

Your role's:

All the graphs shown here are for testing purposes for this mockup.

Figure 21. The view of the users with the `` Generator", `` Producer", “Prosumer” role is presented, they will be able to access the data which they had permission/authorization to observe and/or analyze.

9.6 “Guest Role User (Request New Role)” View

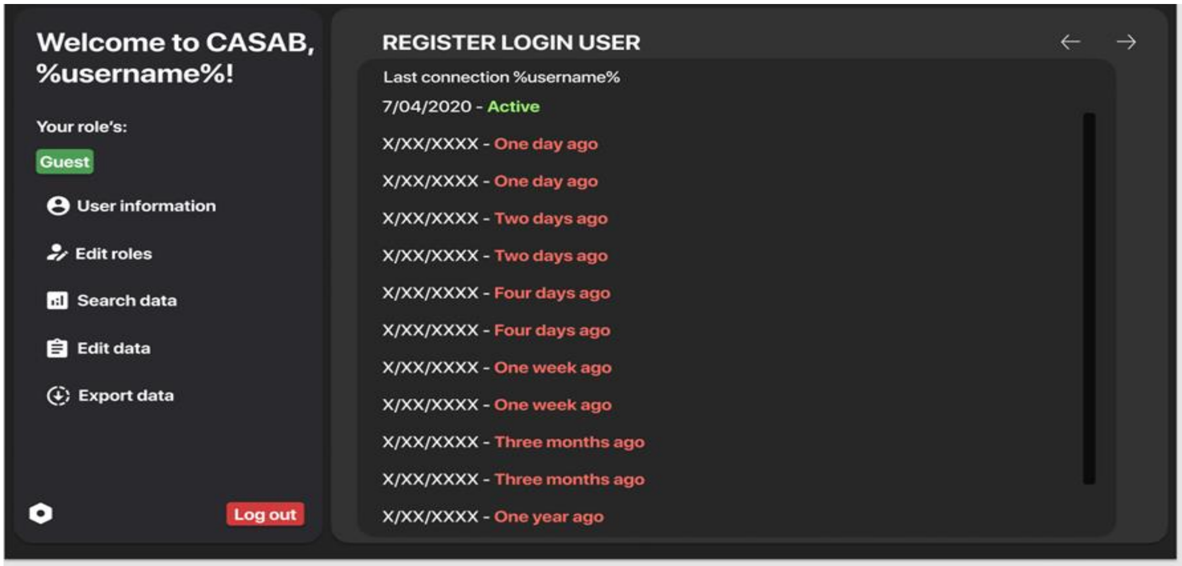


Welcome to CASAB, %username%!

Your role's:

Figure 22. The view of the user with the "Guest" role is presented, who can only request a different role than the one he already has by default.

9.8 “View Access Log” view



Last connection %username%

7/04/2020 - Active

X/XX/XXXX - One day ago

X/XX/XXXX - One day ago

X/XX/XXXX - Two days ago

X/XX/XXXX - Two days ago

X/XX/XXXX - Four days ago

X/XX/XXXX - Four days ago

X/XX/XXXX - One week ago

X/XX/XXXX - One week ago

X/XX/XXXX - Three months ago

X/XX/XXXX - Three months ago

X/XX/XXXX - One year ago

Figure 23. This is the view that any user can access, where an access log is presented, which will also contain the data of the devices from which the accesses were made.

11. References

[1] A. Kumar, K. Abhishek, B. Bhushan, and C. Chakraborty, “Secure access control for manufacturing sector with application of ethereum blockchain,” Peer Peer Netw. Appl., vol. 14, no. 5, p. 3058–3074, 2021, Available:https://link.springer.com/article/10.1007/s12083-021-01108-3

[2] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, “Network intrusion detection system: A systematic study of machine learning and deep learning approaches,” Trans. emerge telecommunity technol., vol. 32, no. 1, 2021, Available:https://onlinelibrary.wiley.com/doi/full/10.1002/ett.4150