**Description**

The API feeds a client for a Live Chat communication. First initialize communication and it can send messages and request them. Also devolve contacts. Finally, the client can close the communication for security reasons.

It was thanked for a principal user and a secondary one. For example, some user sends a message to another user (this is the secondary). If the secondary calls, will be the principal.

The messages have a datetime, a string and the type of the message. The string message could be a normal one, a video link or an image (this could be a link or a code for another API call that was not implemented).

The “principal” user in contact model is obviously the contact owner and the secondary, is his contact. Also, the contact model saves the first message.

**Steps**

1. The Client ask for the token with the userid and password to initialize connection. The system generates a token for the user and return the same.
2. The client sends the message, user id, recipient id and security code (token). Return a boolean.
3. The client asks for the messages with the user id, a message and the security code. Return a list of the messages in json format.
4. The asks for the contacts of the user sending the user id and the security code. Return a list of the contact in json format.
5. The client sends the user id to finish the connection. The system cleans the users token and responds with a Boolean.

**Tecnologies**

* SQL DB
* .NET CORE 3.1

**Tools**

* Postman

**Solution Layers**

* Model
* Services
* Controller

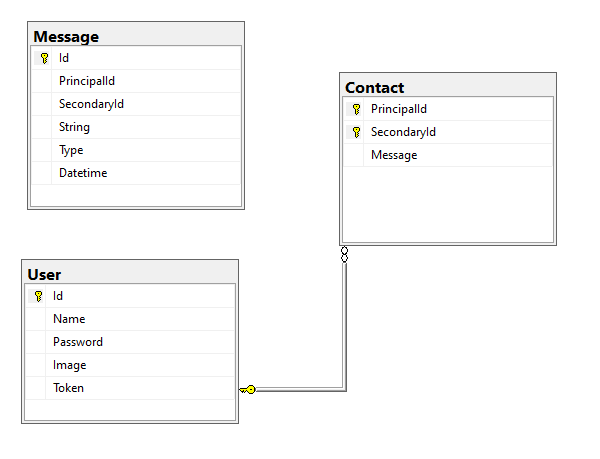
**Design Pattern**

* Singleton
* MVC

**Steps**

1. Create API solution on visual studio.
2. Add nuget packages (Entity, EntityFreamwork.tools, EntityFreamwork.SqlServer, SignalR, Newtosoft.Json)
3. Create database on SQL
4. Create model and context for entity.
5. Initialize migration for entity and database update.
6. Add Json controller in the startup class.
7. Create controller, models controller and service layer.
8. Test with postman.
9. Add signalR folder and class.
10. Add signalR and Cors configuration on startup class.
11. Implement the signalR class on the services layer.

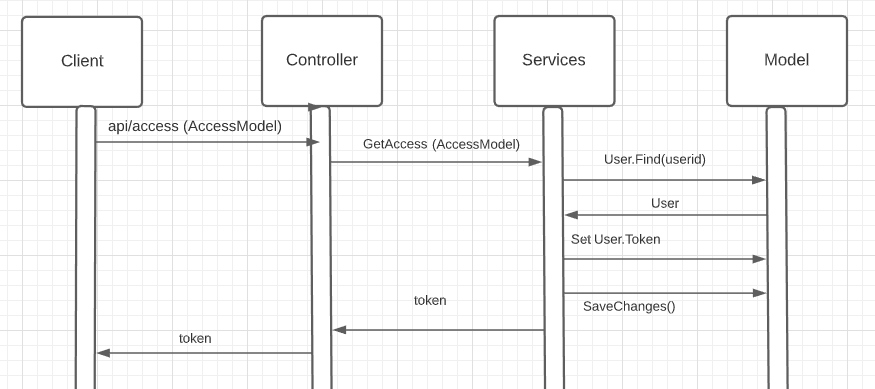
**Data Model**



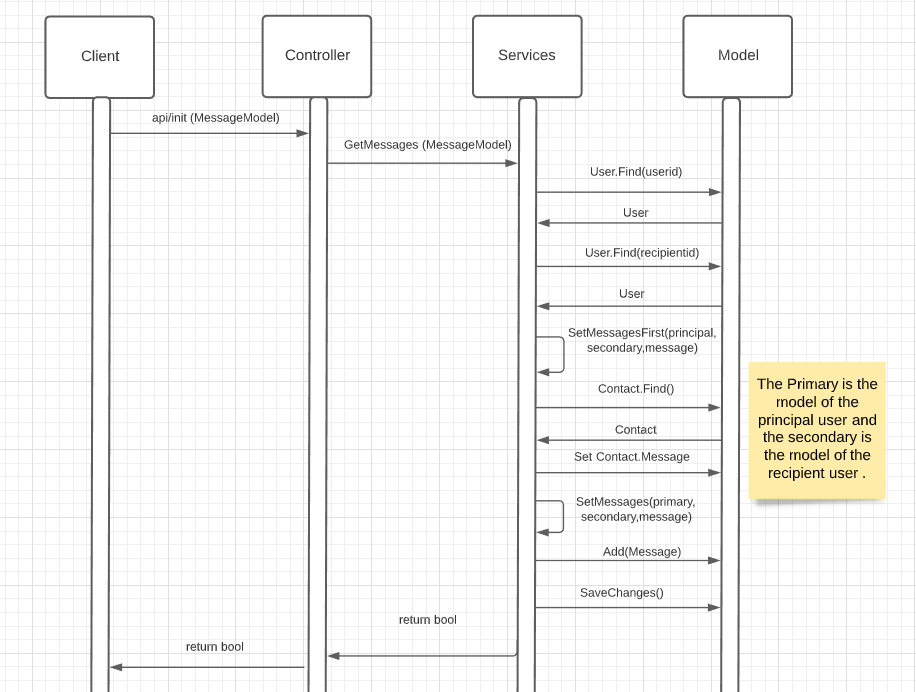
**Note**: The model was made using code first with Entity Freamwork.

**Sequency Diagram**

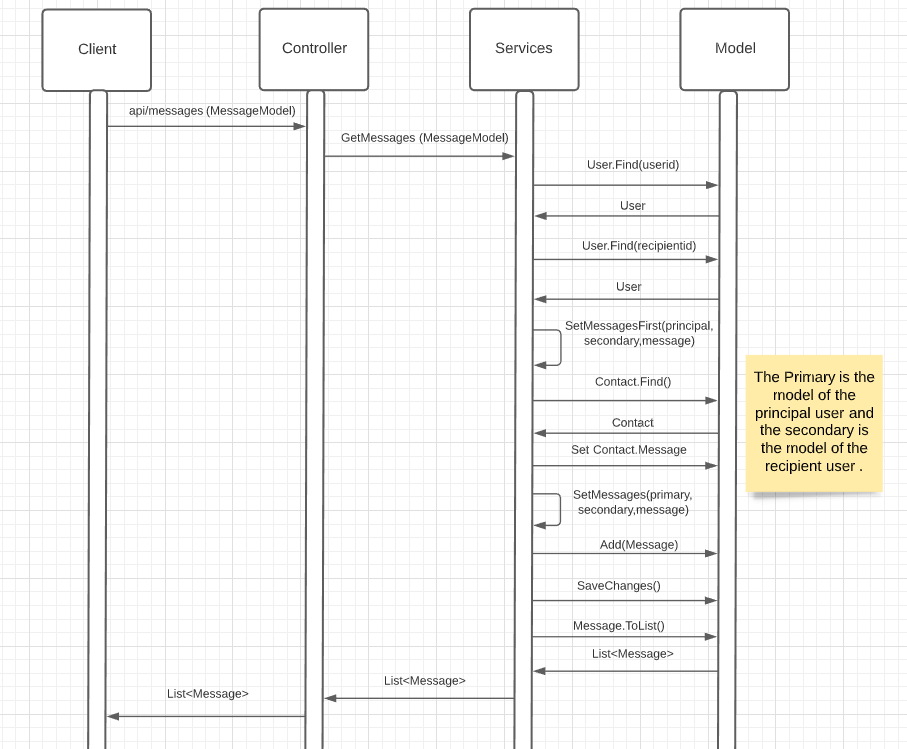
**Access**



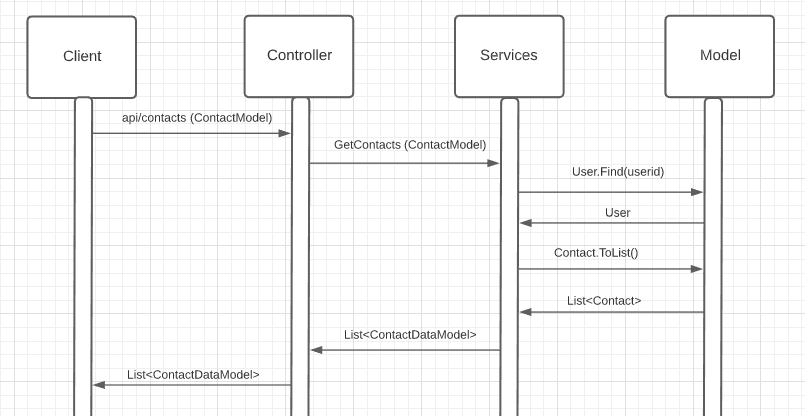
**Initialize**



**Messages**



**Contacts**



**Close**

