Project and Laboratory on Communication Systems

01POJOQ

Prof. Guido Albertengo

Laboratory Report

29 ottobre 2022

Group 3 Chiavello Marco s281692, Crimi Daniele s290164, Noto Gianpietro s281305

Indice

1 Introduction

The project is an implementation of an Inventory Management System. The main purpose is to implement a generic database that can be used for different scenarios (libraries, schools, laboratories, companies etc.).
The requirements are to keep track of items using different fields like description, location, add/-remove, borrow/return, sell/buy.
The system's User interface will be:

- Web;
- Totem dedicated device;
- Smartphone.

Different roles are defined:

- Admin: has a full control of the system, it has all privileges and can play the role of any customer and user;
- Customer: it is the one to whom the service is dedicated, it can add/remove/edit its own users, add/remove/edit items, play the role of any of its users;
- User: the end user it can get/return items.

Since a practical demonstration is required, it was chosen a small shop scenario based on bike sharing service. A Totem is given to the customer to keep track of bike entrances and exits and to rent directly from the device.
An authentication phase is required before rent, since the user must register his personal details for the rent.
The totem recognizes the roles of Admin, Customer and User. The authentication is made through an RFID scanner.

A Web UI is dedicated for the Admin with high privileges, another for the Customer in order to keep track of the items and users registered and also ones for the User to keep track of its personal data.

It was also designed a smarthpone UI dedicated for the Customer to keep track of the items and for the User to check its rent state or its rent/return history. Also in this case an authentication is mandatory using username and password.

2 System implementation

The system (referred to the web site, totem and smartphone) is connected with the database through a web server. In the web server there are different APIs used to manage the HTTP request of the system. In the following figure is shown the flow diagram of the system. The system does not communicate directly with the database but has an interface, for security reasons in order to avoid data leakage. So, every time that the WEB UI, the totem or smartphone ask for data, they make a request through the different APIs.

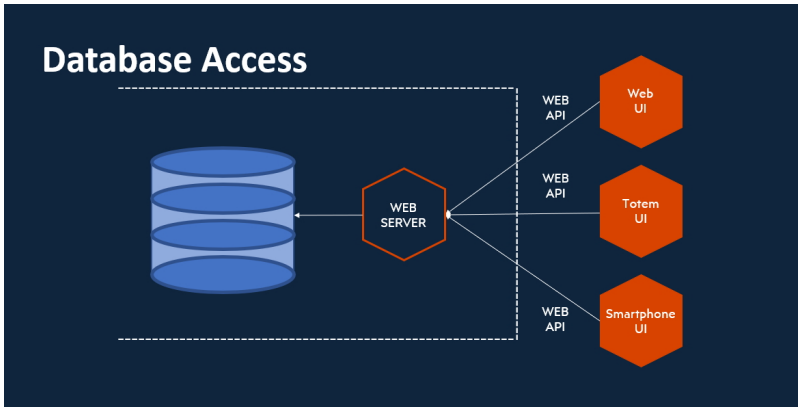

Figura 1: Database access

2.1 Database

The Database organization was implemented in Microsoft SQL server Express environmnet.
The tables are organized as follows:

- customers;
- users;
- items;
- totems_to_customers;
- users_to_customers codes;
- rent_history;

The "User table" identifies the different roles (user, customer, admin) also indicating a unique id, name, surname, user type, username/mail address, password and card id.

MODEM-01\SQLEXPR...ntory - dbo.users | MODEM-01\SQLEXPR...dbo.rent_history | MODEM-01\S

id	name	surname	user_type	usr	pwd	card_id
1	Gianpietro	Noto	Admin	mail1@mail...	1111	1917286187...
2	Daniele	Crimi	Customer	mail2@mail...	1111	4644598958...
3	Marco	Chiavello	User	mail3@mail...	1111	1901132508...
4	Guido	Albertengo	User	mail4@mail...	1111	1917286187...
27	Elon	Musk	User	elonmusk@...	1111	4644598958...
28	Nikola	Tesla	User	nikolatesla...	1111	0000000000...
29	Thomas	Edison	User	thomasedis...	1111	66666666666
30	James	Watt	User	jameswatt...	1111	0000000000...
1046	George	Stephenson	User	georgestep...	1111	0000000000...
1058	prova	prova	User	prova	prova	0000000000...
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figura 2: User table

For the admin and customers, to keep track of the items it was created an "items table" in which are reported id, bike id, card id, category, description, rent date and return date.

MODEM-01\SQLEXPR...tory - dbo.items | MODEM-01\SQLEXPR...dbo.rent_history

id	bike_id	card_id	category	description
1	1	4651085983...	EB	Electric bike
2	2	7399865054...	EB	Electric bike
3	3	1014864412...	EB	Electric bike
4	4	1902474686...	EB	Electric bike
5	5	0000000000...	EB	Electric bike
6	6	00000000	EB	Electric bike
7	7	00000000	EB	Electric bike
8	8	4650918211...	ES	Electric scooter
9	9	00000000	ES	Electric scooter
10	10	00000000	ES	Electric scooter
11	11	00000000	ES	Electric scooter
12	12	00000000	ES	Electric scooter
13	13	00000000	ES	Electric scooter
14	14	00000000	ES	Electric scooter

Figura 3: Items table

For the user's history rent it was created the related table in which are reported id, user id, bike id, rent date, return date.

MODEM-01\SQLEXPR...dbo.rent_history | MODEM-01\SQLEXP...to_customercodes

id	user_id	bike_id	rent_date	return_date
1	3	4	2022-08-05 15:00:00	2022-08-05 15:30:00
2	3	2	2022-08-06 15:00:00	2022-08-06 15:00:00
3	3	4	2022-08-07 15:00:00	2022-08-07 15:00:00
4	1	1	2022-08-08 15:00:00	2022-08-08 15:00:00
5	1	6	2022-09-09 15:00:00	2022-09-09 15:00:00
6	1	2	2022-09-10 15:00:00	2022-09-10 15:00:00
7	3	6	2022-08-11 15:00:00	2022-08-11 15:00:00
8	4	1	2022-09-12 15:00:00	2022-09-12 15:00:00
9	1	2	2022-09-13 15:00:00	2022-09-13 15:30:00
10	3	1	2022-08-14 15:00:00	2022-08-14 15:30:00
11	3	3	2022-08-15 15:00:00	2022-08-15 15:30:00
12	3	4	2022-08-16 15:00:00	2022-08-16 15:00:00
13	3	1	2022-09-13 15:00:00	2022-09-14 16:24:00
14	3	1	2022-09-14 16:35:00	2022-09-14 16:36:00
15	1	2	2022-09-14 16:53:00	2022-09-14 16:54:00

Figura 4: History rent table

Finally for the admin, in order to manage different customers, it was created the "customers table" in which are reported customer name and customer website.

MODEM-01\SQLEXPR...- dbo.customers | SQLQuery4.sql - M

id	website_name	website_address
1	e-motion	www.emotion.com
2	ToShare	www.toshare.com
5	ToRent	www.torent.com
7	RenTo	www.rento.com
8	Noleggio	www.noleggio.com
NULL	NULL	NULL

Figura 5: Customer table

The two last tables "totems to customers" and "Users to customers codes" are linked together with the "Users table" in a way that it is possible to link the totem with the customer.

MODEM-01\SQLEXP...ems_to_customers | M

id	totem_id	customer_c...
1	0193cd	NULL

Figura 6: Totems to customers table

MODEM-01\SQLEXP...to_customercodes MODE

	id	user_id	customer_code
	1	2	123456
	2	29	010101

Figura 7: Users to customers codes table

2.2 Totem features

Once the customer subscribed to the service, provided by the admin, the totem is sent to him, and at first registration, a key6 digit code is sent to its email, so that the customer itself can link the totem to his account. After this, the totem can be clearly used by the users and customer.

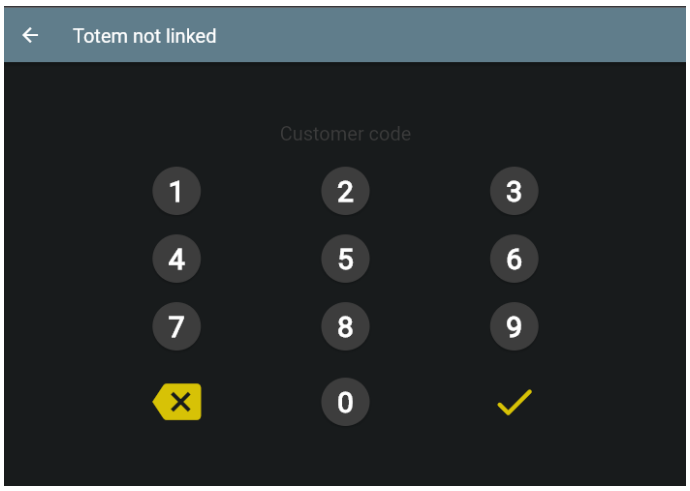

Figura 8: key6 pin code

When an user want to use the service it is required to scan an RFID in order to login, but if is the first access, it needed to register to the service through a QR code, and an RFID card is provided to him by the customer that have linked to its account.

Figura 9: QR code

The only compatible card is a Mifare classic, and when the User press the scan button, it sends a request to the flask server that runs on the totem, so the request will activate the RFID reader, and it will perform the login. The other operation that can be performed on the totem is the return of the item scanning also an RFID code positioned in a safe place on the item. After the login the user is redirected to a page which keep track of active rents and rent history.

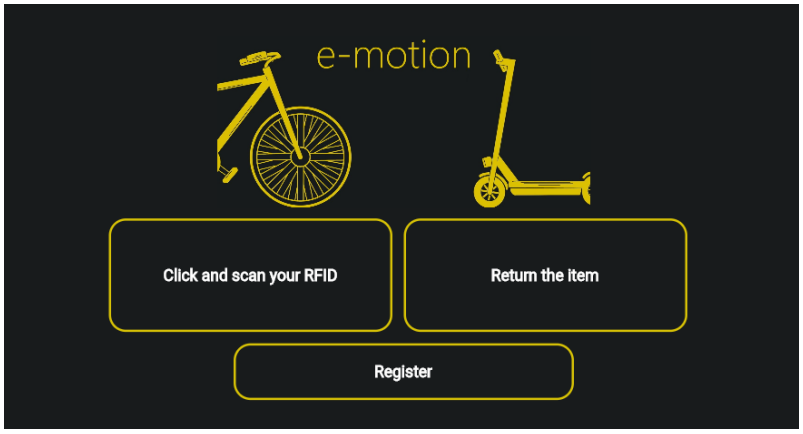

Figura 10: Login User page

3 The Web page

There were developed different Web UI to reach the maximum flexibility, so that, the Admin can play any role of the customers and Users but has the privilege to add/remove Customers; The customers can keep track of the items and Users and can also add/remove/edit them; The Users can only login to visualize the active rents and the recent history rent.

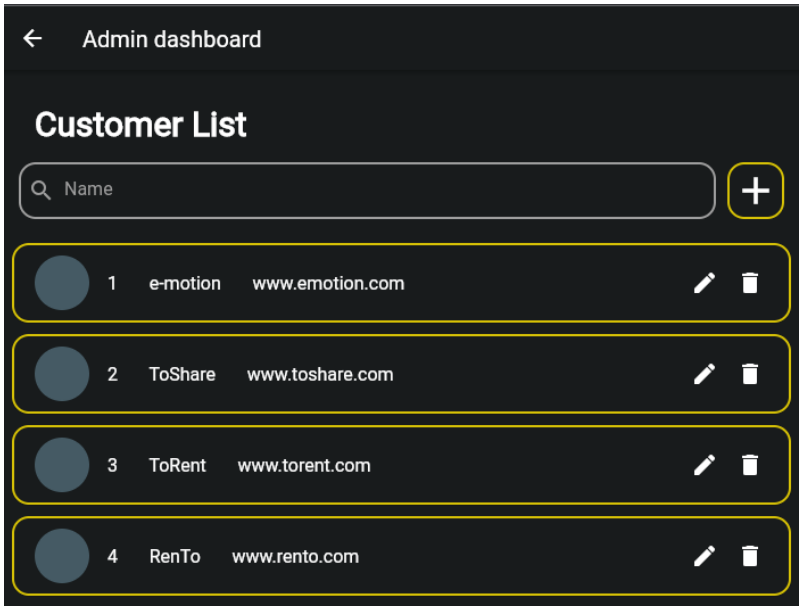

Figura 11: Admin main page

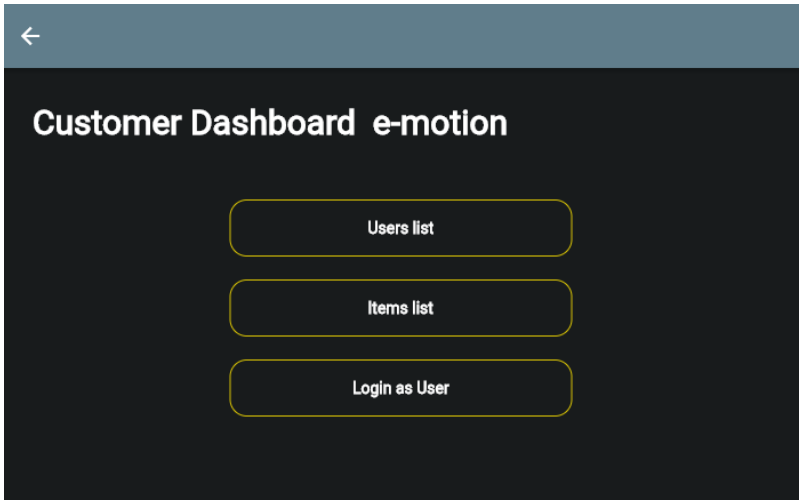

Figura 12: Customer main page

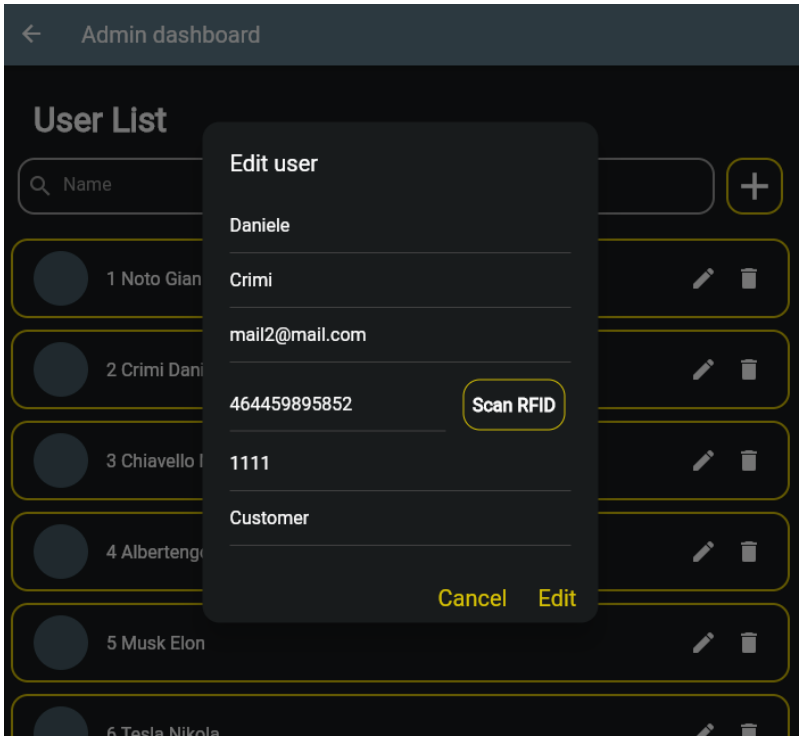

Figura 13: Edit Users feature

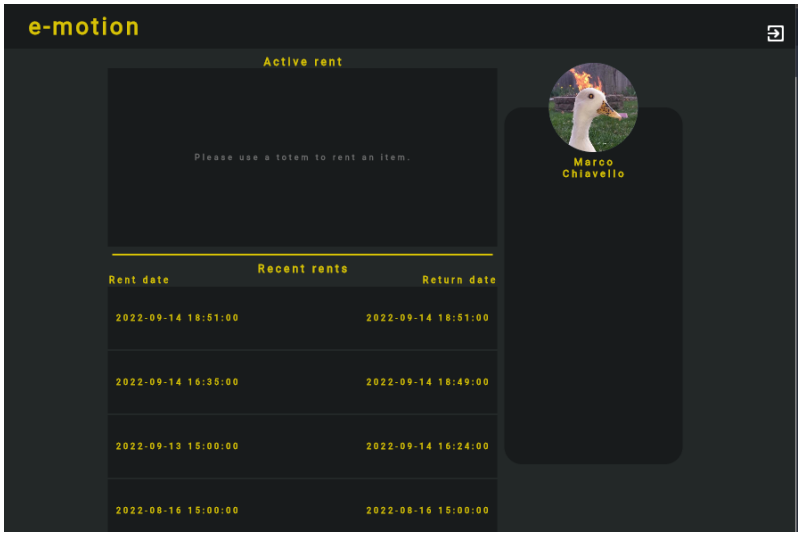

Figura 14: User main page

4 Flutter & Flask communication

The website has been realized with the flutter environment using the dart code language and the entire project is divided in different dart files corresponding to different pages of the website.
The website runs on a flask server, in which there are the APIs that answers to the HTTP requests done by the system to receive data. The HTTP requests are made through the http.dart pckg. Data are passed through JSON files and then decoded in data strings.

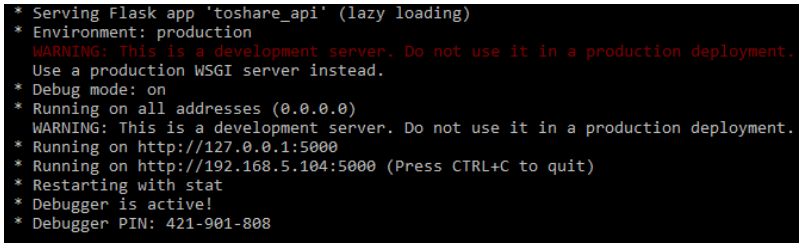

Figura 15: The web server

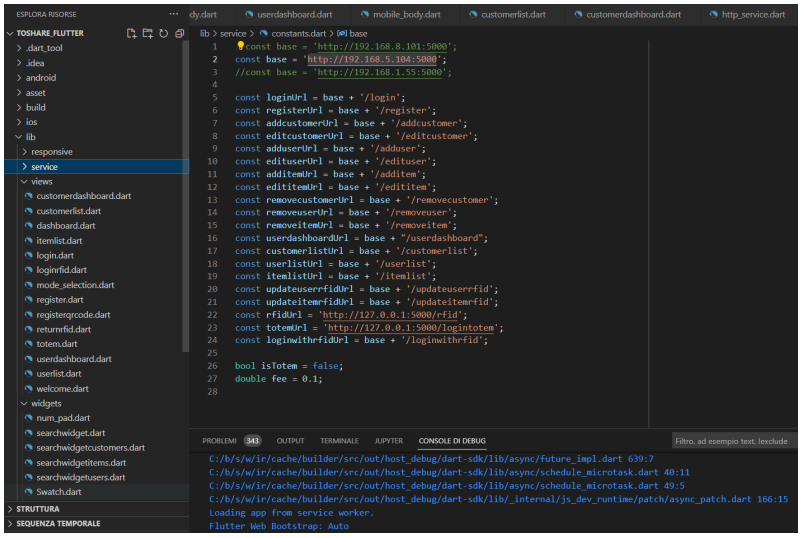

Figura 16: The flutter project

5 Portability on Smartphone

Thanks to the dynamism of the flutter was also developed the Smartphone UI from which the User can login and keep track of the active and all previous rents. It can be also used by the Admin and the customer to add/remove/edit Customers/Users/items. It is only ANDROID compatible.

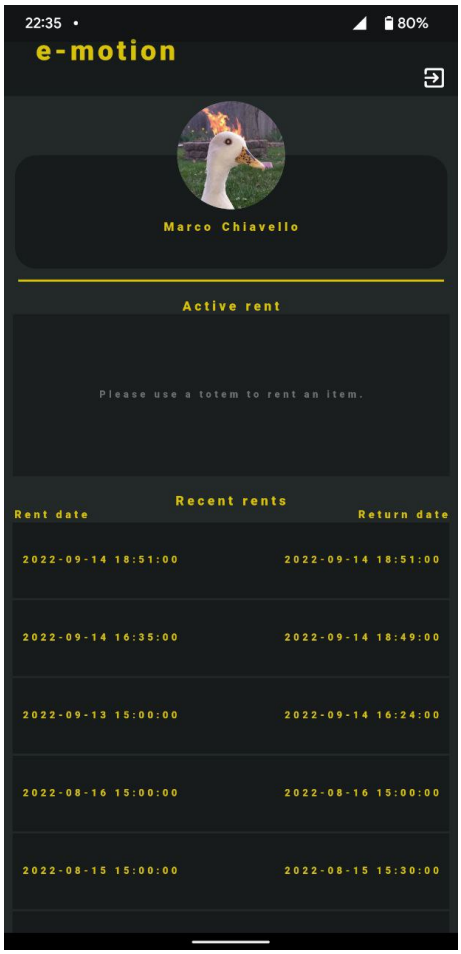

Figura 17: The Smartphone UI