UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

# Web Applications A.Y. 2022-2023

## Homework 1 – Server-side pharmacy management system

## Master Degree in Computer Engineering
## Master Degree in ICT for Internet and Multimedia

Deadline: 28 April, 2023

| **Group Acronym** | **ACME** | |
|---|---|---|
| **Last Name** | **First Name** | **Badge Number** |
| Baghishani | Reihaneh | 2072534 |
| Erfanian Omidvar | Ali | 2080482 |
| Celik | Irem Goksu | 2086040 |
| Haeri | Hamideh | 2080632 |
| Hajizadeh Chavari | Ferdos | 2071542 |
| Hosseinpour | Hossein | 2080601 |
| Sirvanci | Gulce | 2087153 |
| Tahvildari | Ali | 2071563 |
| Torabi | Mohammad | 2080501 |

# 1 Objectives

The objective of the pharmacy website is to provide a secure and user-friendly platform for pharmacy staff to manage drugs and materials, prescriptions, and orders efficiently, and maintain accurate records of inventory, suppliers, and sales. The system is designed to generate reports that provide better insights and facilitate decision-making for pharmacy staff. By achieving these objectives, the system aims to enhance the productivity and effectiveness of the pharmacy staff while ensuring the secure and efficient management of drugs and materials.

# 2 Main Functionalities

The pharmacy website offers several key functionalities for its staff members to manage business operations. The admin creates accounts for staff members, who can then access the system using their login credentials to perform various tasks such as browsing products, placing orders with suppliers, and managing prescriptions. Staff members can also upload prescriptions on behalf of customers, and pharmacists can view the prescription and dispense the medication accordingly.

Another important feature is storage management. The website provides a system for managing the storage of drugs and materials. It tracks the quantity available and creates alerts for low stock levels. This helps ensure that there is always sufficient stock on hand.

Order management is another critical functionality offered by the website. Staff members can place orders with suppliers for drugs and materials through the website, and once the orders are received from the suppliers, pharmacists can review and fulfill the orders by dispensing the requested items from the storage. This process is streamlined and efficient, ensuring that customers receive their orders in a timely manner.

Supplier management is also a key component of the website. Suppliers who provide drugs and materials to the pharmacy are managed through the system. The website stores information about the supplier and the products they supply. Reports can be generated to track supplier performance and assess the quality of the products received.

Reporting is another vital functionality provided by the pharmacy website. The system generates reports on various aspects of the business such as sales and storage levels. These reports help the pharmacy to make informed decisions about stocking and pricing products, as well as identifying trends in customer behavior.

Payment processing is an essential part of the website. Staff members can record how the customer intends to pay for their order, such as cash or credit card, but the actual payment is not processed through the website.

Finally, receipt management is an important feature. The system generates receipts for each order and stores them for future reference. This helps ensure that the pharmacy maintains accurate records and can easily track customer transactions.

Role-based access control is also in place to ensure that only authorized personnel can perform certain functions within the system. This enhances security and ensures that sensitive data is protected.

# 3 Data Logic Layer

## 3.1 Entity-Relationship Schema

The Entity-Relationship schema of this project (Figure 1) contains 11 main entities:

- *pharmacy*: An entity named Pharmacy has several attributes, including an ID, a name, an address, a telephone number, a list of opening times, a logo image, a list of stored items, and a list of staff members.
- *TimeTable*: this entity is a system that represents the opening hours of a pharmacy. It has attributes such as an id, $from\_hour$, and $to\_hour$ strings which represent the start and end times for a given opening period.
- *User*: The *User* is an entity in a system that represents a user. It has various attributes such as name, last name, gender, birth date, phone number, address, role, email, account status, profile picture, and pharmacy.
- *role*: The *Role* is an entity that represents a database table with an ID and a unique role name.
- *login*: The entity contains numerous attributes, such as an automatically generated ID, a username, a password, and a session flag. Additionally, it holds information regarding the user's last login time, activation date, termination date, and a reference to another User object.
- *Supplier*: it is an entity in a system that represents a supplier. It has various attributes such as an id, name, address, email, and telephone number. It also has two lists of drugs and materials that the supplier provides.
- *order*: The ordering entity has several attributes, including an ID, an order date, lists of drugs and materials, an order status, a price, and an active status.
- *Storage*: This is an entity in a system that represents the number of drugs and materials available at a pharmacy. It has attributes such as an id, amount (the quantity of the drug or material stored), and threshold (the minimum amount of a drug or material that must be kept in stock).
- *drug*: The Drug is an entity that represents a drug in a database. It contains 16 attributes, including id, name, supplier, expirationDate, image, shape, gender (the gender specification of the drug), ageGroup (the age group that the drug is intended for), isSensitive (a boolean value indicating if the drug is sensitive), needPrescription (a boolean value indicating if the drug requires a prescription), description, limitation (the number of drugs that a patient can purchase), price, countryOFProduction (the country of production for the drug), lastModifiedDate(the date and time the drug was last modified)
- *material*: Material entity is similar to drug entity. It has an ID, a name, a supplier, a country of production, an expiration date, an image, a gender, a price, an age group, a last modified date, and a description.
- *reciept*: The entity has several attributes, including an ID, lists of drugs and materials, an image of the receipt, a date, and a payment method.

## 3.2 Other Information

There are some attributes such as gender, country, $age\_group$, $order\_status$, and $payment\_method$ that can have a finite set of possible values. So they will be defined as enumerable values.

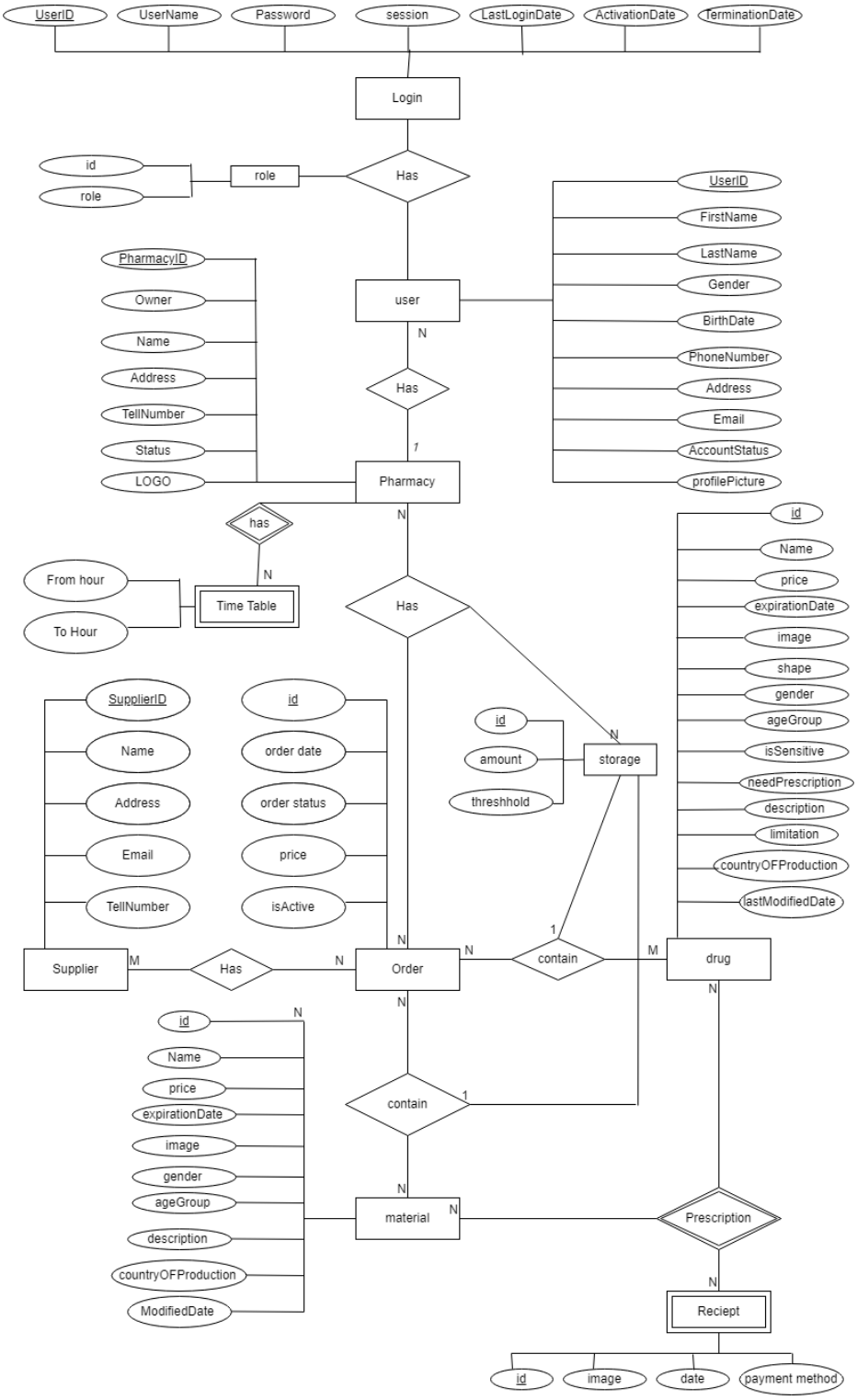


Figure 1: Entity-Relationship Schema

# 4 Presentation Logic Layer

Wireframing is an essential part of designing a website or application. It refers to the process of creating a visual layout of a webpage or screen, outlining the position and size of each element on the page. By using wireframes, designers can determine the user experience and the complexity of the UI, as well as the content that the user sees. It helps to create a tangible representation of the user interface and provides a clear understanding of how the user interacts with the system.

In our project, we utilized wireframing to design and implement the user interface of our web application. We used wireframes to determine the layout and design of our web pages and to ensure a consistent user experience across the entire application. By using wireframes, we were able to visualize the content and layout of the web pages and make necessary changes before implementing the final design. Our wireframes helped us to create a user-friendly and efficient web application, enhancing the user experience and ensuring the success of our project.

## 4.1 Login page

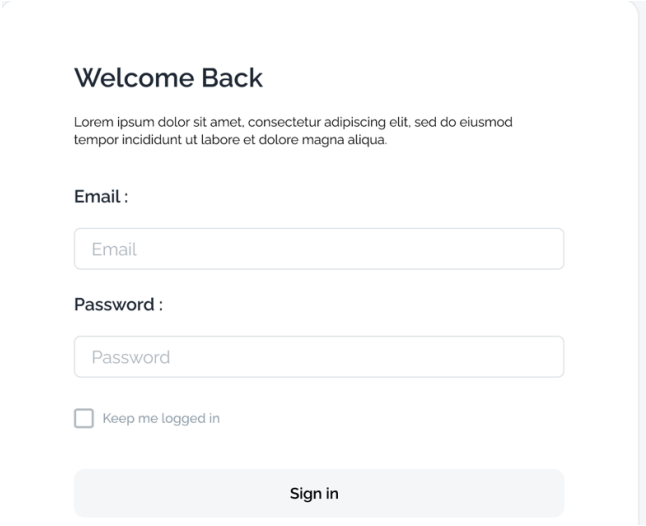


Figure 2: Login Page

The login page of the pharmacy system (Figure 2) would likely have two input fields for the user to enter their email address and password. There would also be a "Login" button or similar call-to-action element that the user can click once they have entered their login credentials. This button would submit the form and initiate the authentication process.

## 4.2 Reciept page

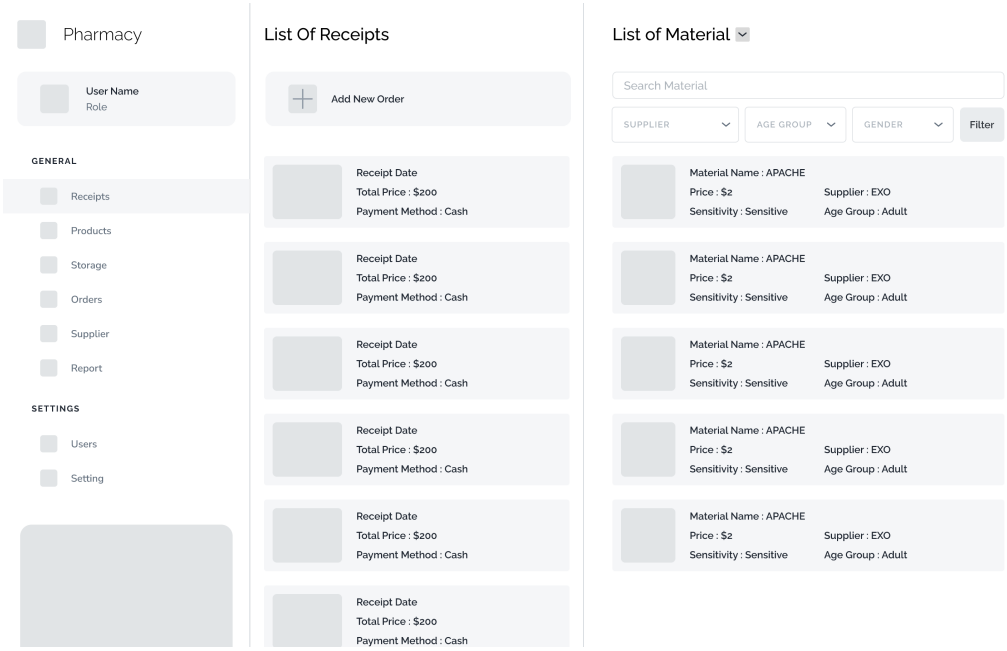


Figure 3: Login Page

The receipt form (Figure 3) shows the receipt of a prescription. it may contain drugs or materials or both.

# 5 Business Logic Layer

## 5.1 Class Diagram

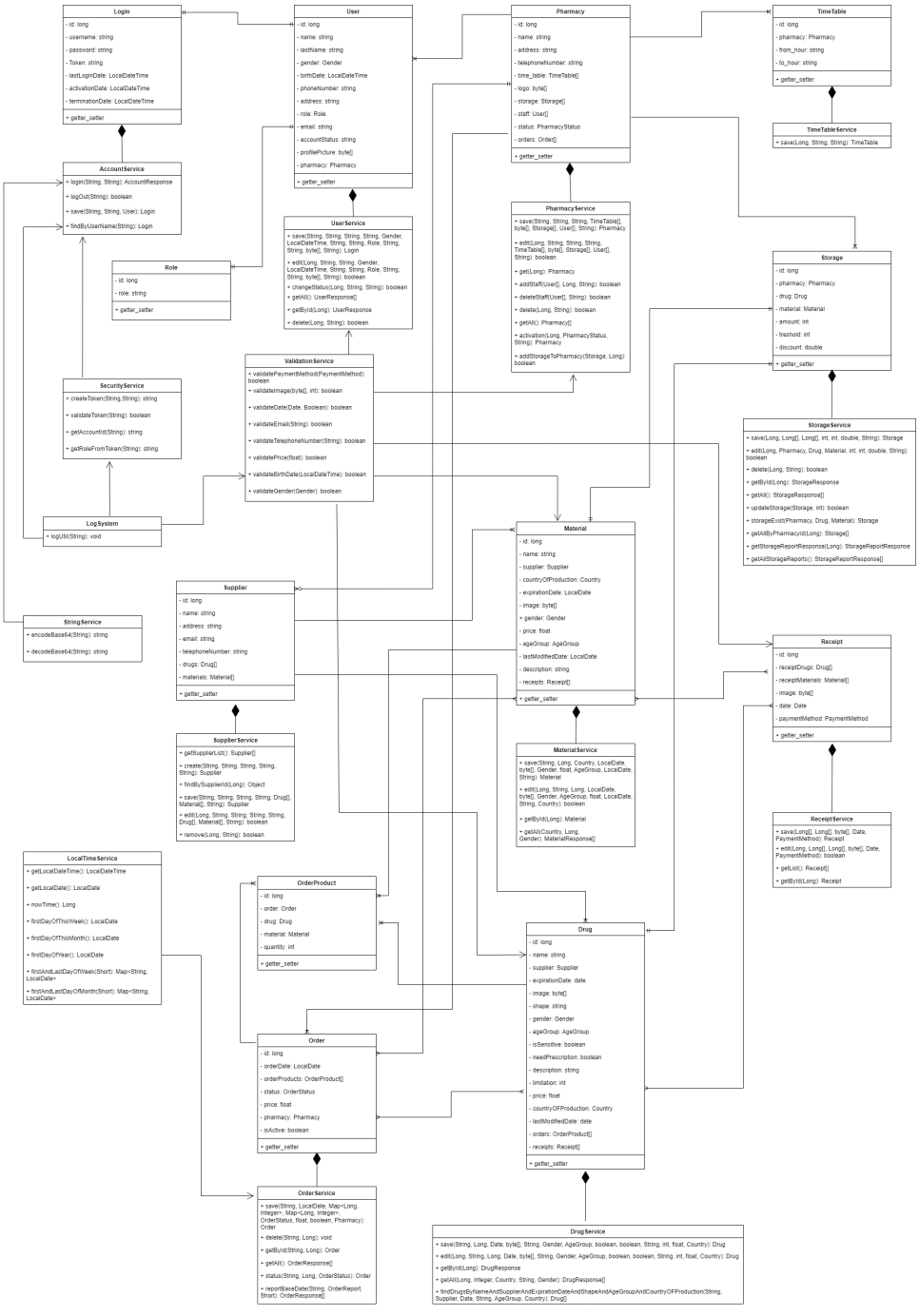


Figure 4: Class Diagram

The class diagram (Figure 4) consists of several classes that represent the entities and services in the system. Here's a brief overview of each class and its purpose:

- Login: This class represents a user account in the system. It contains the username and password fields, which are used for authentication. The Account class is used by the service to handle login, logout, and account creation.
- User: This class represents a user in the system. It contains fields such as name, username, password, and email, as well as additional fields such as gender, birth date, and address. The User class is used by the UserService to handle user management, such as adding, editing, and deleting user accounts.
- Pharmacy: This class represents a pharmacy in the system. It contains fields such as name, address, and phone number, as well as a list of staff members, storage items, and timetables. The Pharmacy class is used by the PharmacyService to handle pharmacy management, such as adding, editing, and deleting pharmacy accounts.
- TimeTable: This class represents a weekly timetable that defines the working hours of a pharmacy. It contains the fromHour and toHour fields, which represent the start and end times of a workday. The TimeTable class is used by the Pharmacy class to define the working hours of a pharmacy.
- Drug: This class represents a drug that can be stored in a pharmacy. It contains fields such as name, supplier, expiration date, price, and additional fields such as age group and country of production. The Drug class is used by the DrugService to handle drug management, such as adding, editing, and deleting materials/drugs.
- Material: The Material class is a class that shares similarities with a drug class. It has various attributes such as name, supplier, expiration date, and price, as well as extra attributes like the age group and country where it was produced. The MaterialService utilizes this class to perform tasks related to material management including adding, modifying, and removing materials.
- Supplier: This class represents a supplier of materials/drugs. It contains fields such as name, address, email, and telephone number. The Supplier class is used by the Material/Drug class to specify the supplier of a material/drug.
- Storage: This class represents a storage item in a pharmacy, which associates a material/drug with a quantity. It contains fields such as pharmacy, drug, and quantity. The Storage class is used by the Pharmacy class to manage the inventory of a pharmacy.
- Order: This class represents an order placed by a user. It contains fields such as date, total cost, user, and a list of order items. The Order class represents a customer's order, which can be fulfilled by a pharmacy.
- Role: The Role class represents a role in the system. It contains fields such as id and role, which are used to uniquely identify a role and its name, respectively.
- LogSystem: This interface represents a system for logging messages. It contains a single method, logUtil, which takes a message as input and logs it into the system.
- OrderProduct: The OrderProduct class represents an individual product within an order. It contains fields such as the product's unique identifier (id), the order it belongs to (order), the drug or material being ordered (drug or material), and the quantity being ordered (quantity). This class is used by the Order class to specify the products included in an order.
- StringService: StringService is an interface that provides two methods for encoding and decoding strings using Base64. The encodeBase64() method takes an input string and returns its Base64-encoded equivalent. The decodeBase64() method takes a Base64-encoded string and returns its decoded value.

- SecurityService: This interface represents a security service in the system. It provides methods for creating and validating tokens, as well as retrieving information from tokens. The create token method takes in an accountId and role and generates a unique token that can be used for authentication purposes. The validateToken method checks if a token is valid or not. The getAccountId method extracts the accountId from a given token, while the getRoleFromToken method extracts the role from a given token. These methods are used by other services to ensure secure access to sensitive information and operations in the system.
- LocalTimeService: LocalTimeService is an interface that provides access to methods for working with local dates and times. It has three methods: getLocalDateTime(), which returns the current date and time in the system's default time zone; getLocalDate(), which returns the current date in the system's default time zone; and nowTime(), which returns the current time in milliseconds since the Unix epoch (January 1, 1970, 00:00:00 GMT).
- AccountService: AccountService is an interface that represents the functionality of a service responsible for managing user accounts. It provides methods for login, logout, and creating a new user account. The login method takes in the username and password of a user and returns an AccountResponse object, which contains information about the user account, including a token for future authentication. The logout method takes in a token and logs out the user associated with that token. The save method is used to create a new user account and takes in the username, password, and user information. The findByUserName method is used to find a user account based on the username.
- ValidationService: ValidationService is an interface that defines methods to validate certain types of data used in the system. It contains methods to validate payment methods, images, dates, email addresses, telephone numbers, prices, birth dates, and gender. The methods take input parameters of the data to be validated and return a Boolean value indicating whether the data is valid or not. By providing a centralized place for data validation, the validation service helps to ensure consistency and accuracy throughout the system.

## 5.2 Sequence Diagram

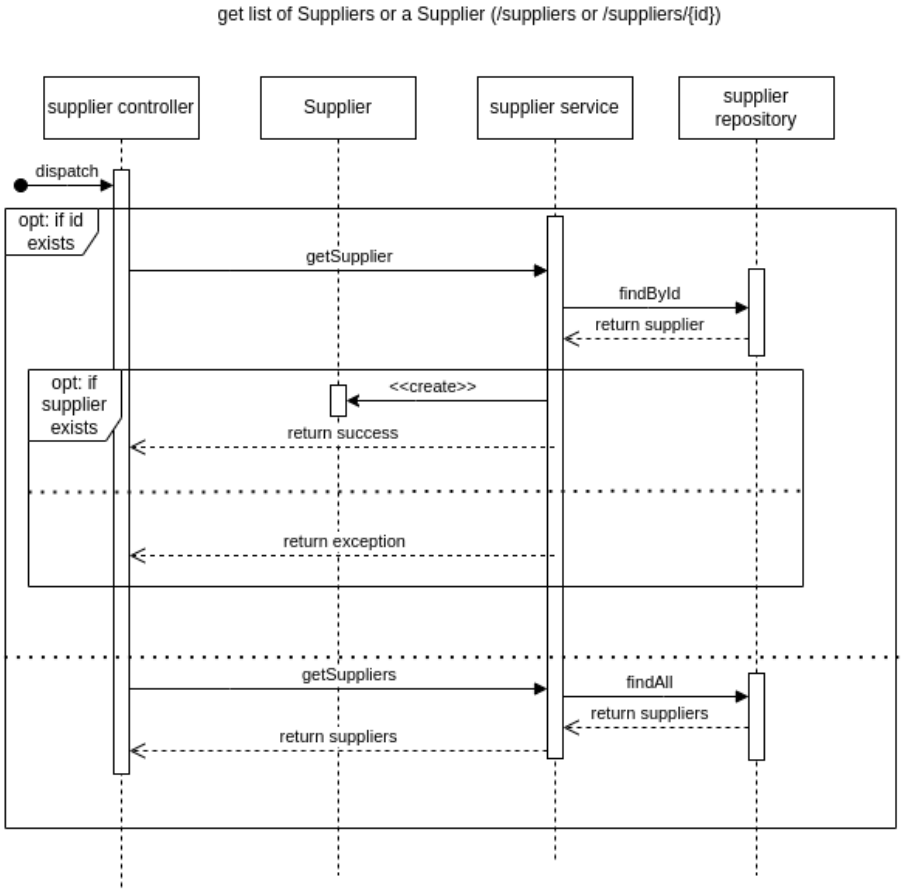


Figure 5: Fetching Suppliers Sequence Diagram

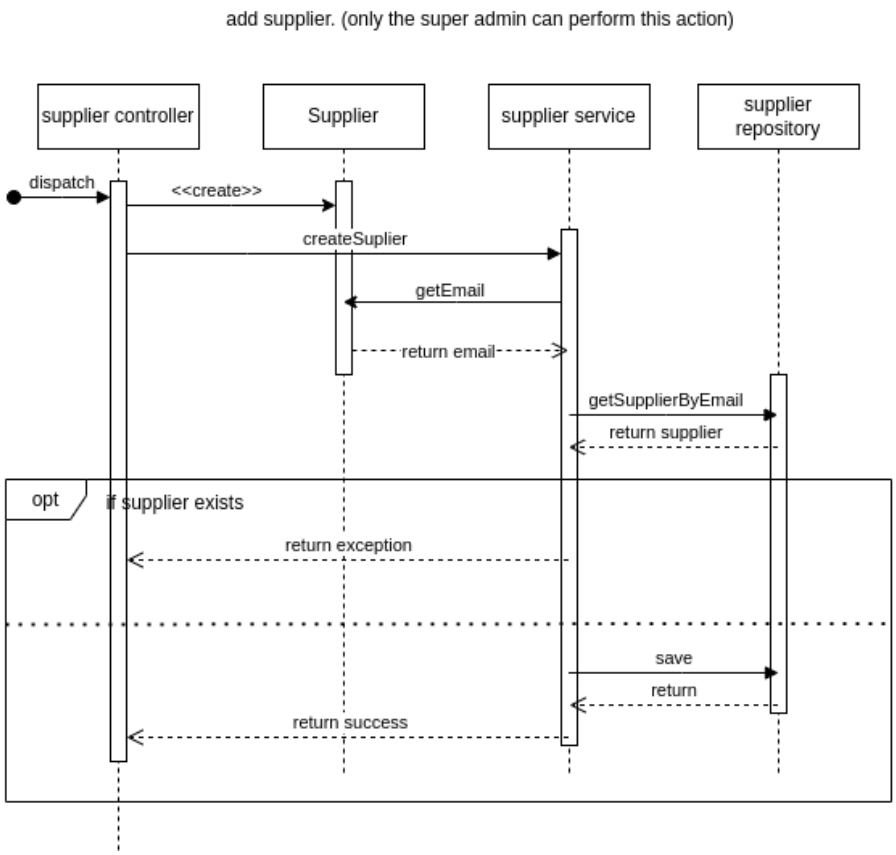


Figure 6: Adding Suppliers Sequence Diagram

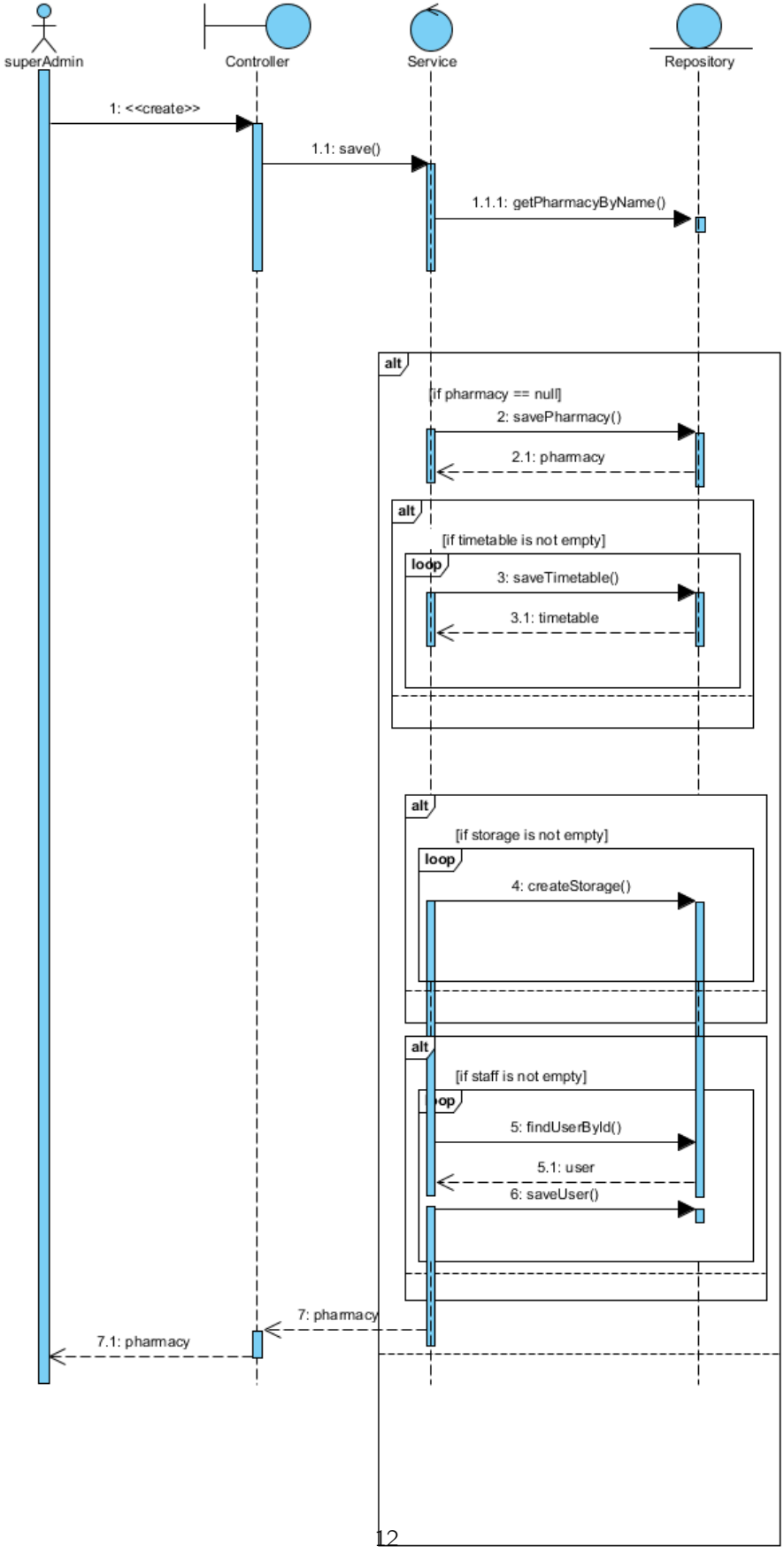


Figure 7: Creating Pharmacy Sequence Diagram

## 5.3 REST API Summary

We used Swagger UI to document our REST API. Swagger is an open-source tool that can help developers design, build, and document RESTful APIs. With Swagger UI, APIs can be easily tested and explored using a user-friendly interface. Additionally, it can generate interactive documentation that can be shared with other developers. The following table shows the URIs, HTTP methods, and descriptions for each endpoint in our API:

| URI | Method | Description |
|---|---|---|
| user/id | GET | This API endpoint retrieves the details of a specific user based on their unique ID. |
| user/id | PUT | This API endpoint allows for updating the information of a specific user based on their unique ID. |
| user | GET | This API endpoint retrieves a list of all users stored in the system. |
| user | POST | This API endpoint allows for the creation of a new user in the system. |
| user/change-status/id/newStatus | PATCH | This API endpoint allows for changing the status of a specific user based on their unique ID. |
| user/delete/id | DELETE | This API endpoint allows for deleting a specific user based on their unique ID. |
| supplier/id | GET | This API endpoint retrieves the details of a specific supplier based on their unique ID. |
| supplier/id | PUT | This API endpoint allows for updating the information of a specific supplier based on their unique ID. |
| supplier/id | DELETE | This API endpoint allows for deleting a specific supplier based on their unique ID. |
| supplier | POST | This API endpoint allows for the creation of a new supplier in the system. |
| storage/id | GET | This API endpoint retrieves the details of a specific storage based on its unique ID. |
| storage/id | PUT | This API endpoint allows for updating the information of a specific storage based on its unique ID. |
| storage/id | DELETE | This API endpoint allows for deleting specific storage based on its unique ID. |
| storage | GET | This API endpoint retrieves the list of all storage available in the system. |
| storage | POST | This API endpoint allows for creating new storage in the system. |
| receipt/id | GET | This API endpoint retrieves the details of a specific receipt identified by the given ID. |

| | | |
|---|---|---|
| receipt/id | PUT | This API endpoint updates the details of a specific receipt identified by the given ID. |
| receipt | POST | This API endpoint allows for creating a new receipt in the system. |
| pharmacy/id | GET | This API endpoint retrieves the details of a specific pharmacy based on its unique ID. |
| pharmacy/id | PUT | This API endpoint allows for updating the information of a specific pharmacy based on its unique ID. |
| pharmacy/id | DELETE | This API endpoint allows for deleting a specific pharmacy based on its unique ID. |
| pharmacy/id/add-staff | PUT | This API endpoint allows for adding a new staff member to a specific pharmacy based on its unique ID. |
| pharmacy | GET | This API endpoint retrieves a list of all pharmacies stored in the system. |
| pharmacy | POST | This API endpoint allows for the creation of a new pharmacy in the system. |
| pharmacy/pharmacyId | PATCH | This API endpoint allows for updating the information of a specific pharmacy activation status based on its unique ID. It also changes the staff status. |
| pharmacy/delete-staff | DELETE | This API endpoint allows for deleting a specific staff member from a pharmacy based on their unique ID. |
| material/id | GET | This API endpoint retrieves the details of a specific material based on its unique ID. |
| material/id | PUT | This API endpoint allows for updating the information of a specific material based on its unique ID. |
| material | GET | This API endpoint retrieves a list of all materials stored in the system. |
| material | POST | This API endpoint allows for the creation of new material in the system. |
| drug/id | GET | This API endpoint retrieves the details of a specific drug based on its unique ID. |
| drug/id | PUT | This API endpoint allows for updating the information of a specific drug based on its unique ID. |
| drug | GET | This API endpoint retrieves a list of all drugs stored in the system. |
| drug | POST | This API endpoint allows for the creation of a new drug in the system. |

| | | |
|---|---|---|
| order/save | POST | This API endpoint allows for saving a new order in the system. |
| order/status/id/status | PATCH | This API endpoint allows for updating the status of a specific order based on its unique ID. |
| order/getById/id | GET | This API endpoint retrieves the details of a specific order based on its unique ID. |
| order/get-all | GET | This API endpoint retrieves a list of all orders stored in the system. |
| order/delete/id | DELETE | This API endpoint allows for deleting a specific order based on its unique ID. |
| account/token | POST | This API endpoint allows for creating a new account in the system or logging in to an existing account. |
| account | GET | This API endpoint retrieves the details of the authenticated user's account. |

Table 2: All the REST API Summaries describe in this table your REST API

## 5.4 REST Error Codes

| Error Code | HTTP Status Code | Description |
|---|---|---|
| 400 | BAD_REQUEST | Account Access Not Found |
| 404 | NOT_FOUND | User Name Or Password Not Exists Exception |
| 500 | INTERNAL_SERVER_ERROR | Stack Overflow Error |
| 400 | BAD_REQUEST | Invalid Parameter Exception |
| 403 | FORBIDDEN | Invalid Token Exception |
| 404 | NOT_FOUND | Not Found Exception |
| 403 | FORBIDDEN | Permission Denied |
| 405 | METHOD_NOT_ALLOWED | Pharmacy Exists Before |
| 404 | NOT_FOUND | User Do Not Exist |
| 400 | BAD_REQUEST | Create Pharmacy Data Not Found |
| 404 | NOT_FOUND | Entity Not Found Exception |
| 400 | BAD_REQUEST | Illegal Argument Exception |
| 424 | FAILED_DEPENDENCY | Illegal State Exception |
| 404 | NOT_FOUND | Entity Not Found Exception |

Table 3: All the REST Error code Described in this table

## 5.5 REST API Details

### CREATE DRUG

The following endpoint allows the creation of a new drug in the pharmacy system.

- **URL:**
  `/drug`
- **Method:**
  `POST`
- **Data Parameters:**

```
{
    "name": "proteine",
    "supplier": 2,
    "expirationDate": "2024-06-30",
    "shape": "tablet",
    "gender": "FEMALE",
    "ageGroup": "ADULTS",
    "isSensitive": true,
    "needPrescription": false,
    "description": "protein is a needed.",
    "limitation": 20,
    "price": 7.99,
    "countryOFProduction": "AF"
}
```

- **Success Response:**

```
{
    "msg": "OK",
    "data": {
        "id": 5,
        "name": "proteine",
        "expirationDate": "2024-06-30",
        "image": null,
        "shape": "tablet",
        "gender": "FEMALE",
        "ageGroup": "ADULTS",
        "needPrescription": false,
        "description": "protein is a needed.",
        "limitation": 20,
        "price": 7.99,
        "countryOFProduction": "AF",
        "lastModifiedDate": null,
        "sensitive": true
    }
}
```

- **Error Response:**

```
{
"msg": "drug_exists_before",
"data": "com.unipd.semicolon.business.exception.DrugExistsException"
}
```

## EDIT DRUG

The following endpoint allows the modification of an existing drug in the pharmacy system.

- **URL:**
  `/drug/id`
- **Method:**
  `PUT`
- **Data Parameters:**

```
{
    "name": "Ibuprofen",
    "expirationDate": "2024-06-30",
    "supplier": 1,
    "shape": "tablet",
    "gender": "FEMALE",
    "ageGroup": "ADULTS",
    "needPrescription": false,
    "description": "Ibuprofen is a nonsteroidal anti-inflammatory drug (
       NSAID) used to relieve pain and reduce fever.",
    "limitation": 20,
    "price": 7.99,
    "countryOFProduction": "PR",
    "lastModifiedDate": null,
    "sensitive": false
}
```

- **Success Response:**

```
{
    "msg": "OK",
    "data": {
        "id": 5,
        "name": "Ibuprofen",
        "expirationDate": "2024-06-30",
        "image": null,
        "shape": "tablet",
        "gender": "FEMALE",
        "ageGroup": "ADULTS",
        "needPrescription": false,
        "description": "Ibuprofen is a nonsteroidal anti-inflammatory
           drug (NSAID) used to relieve pain and reduce fever.",
        "limitation": 20,
        "price": 7.99,
        "countryOFProduction": "PR",
```

```
            "lastModifiedDate": null,
            "sensitive": true
        }
    }
```


### GET DRUG

The following endpoint retrieves the details of a specific drug based on its unique ID.

- **URL:**
  `/drug/id`
- **Method:**
  `GET`
- **Success Response:**


```
{
    "msg": "OK",
    "data": {
        "id": null,
        "name": "proteine",
        "supplier": {
            "id": 2,
            "name": "sup1",
            "address": "here in padova",
            "email": "su1@pd.it",
            "telephoneNumber": "099999",
            "drugs": [
                {
                    "id": 4,
                    "name": "proteine",
                    "expirationDate": "2024-06-30",
                    "image": null,
                    "shape": "tablet",
                    "gender": "FEMALE",
                    "ageGroup": "ADULTS",
                    "needPrescription": false,
                    "description": "protein is a needed.",
                    "limitation": 20,
                    "price": 7.99,
                    "countryOFProduction": "AF",
                    "lastModifiedDate": null,
                    "sensitive": true
                }
            ],
            "materials": [

            ]
        },
        "expirationDate": "2024-06-30",
```

```
            "image": null,
            "shape": "tablet",
            "gender": "FEMALE",
            "ageGroup": "ADULTS",
            "needPrescription": false,
            "description": "protein is a needed.",
            "limitation": 20,
            "price": 7.99,
            "countryOFProduction": "AF",
            "lastModifiedDate": null,
            "orders": null,
            "receipts": null,
            "sensitive": true
        }
    }
```


## GET ALL DRUG

The following endpoint retrieves a list of all drugs stored in the pharmacy system.

- **URL:**
  `/drug`
- **Method:**
  `GET`
- **Success Response:**


```
{
    "msg": "OK",
    "data": [
        {
            "id": null,
            "name": "Ibuprofen",
            "supplier": {
                "id": 1,
                "name": "sup0",
                "address": "pd",
                "email": "su0@pd.it",
                "telephoneNumber": "00000",
                "drugs": [
                    {
                        "id": 3,
                        "name": "Ibuprofen",
                        "expirationDate": "2024-06-30",
                        "image": null,
                        "shape": "tablet",
                        "gender": "MALE",
                        "ageGroup": "ADULTS",
                        "needPrescription": false,
```

```
                        "description": "Ibuprofen is a nonsteroidal anti
                            -inflammatory drug (NSAID) used to relieve
                            pain and reduce fever.",
                        "limitation": 20,
                        "price": 5.99,
                        "countryOFProduction": "PR",
                        "lastModifiedDate": null,
                        "sensitive": false
                    }
                ],
                "materials": []
            },
            "expirationDate": "2024-06-30",
            "image": null,
            "shape": "tablet",
            "gender": "MALE",
            "ageGroup": "ADULTS",
            "needPrescription": false,
            "description": "Ibuprofen is a nonsteroidal anti-
                inflammatory drug (NSAID) used to relieve pain and reduce
                 fever.",
            "limitation": 20,
            "price": 5.99,
            "countryOFProduction": "PR",
            "lastModifiedDate": null,
            "orders": null,
            "receipts": null,
            "sensitive": false
        }
    ]
}
```


## CREATE MATERIAL

The following endpoint allows the creation of a new material in the pharmacy system.

- **URL:**
  `/material`
- **Method:**
  `POST`
- **Data Parameters:**


```
{
  "name": "Example Product",
  "supplier": 2,
  "countryOfProduction": "PR",
  "expirationDate": "2022-12-31",
  "gender": "FEMALE",
  "price": 50.0,
```

```
  "ageGroup": "ADULTS",
  "description": "This is an example product description. It can be as
      long or as short as you like."
}
```

- **Success Response:**

```
{
    "msg": "OK",
    "data": {
        "id": 6,
        "name": "Example Product",
        "countryOfProduction": "PR",
        "expirationDate": "2022-12-31",
        "image": null,
        "gender": "FEMALE",
        "price": 50.0,
        "ageGroup": "ADULTS",
        "lastModifiedDate": null,
        "description": "This is an example product description. It can
            be as long or as short as you like."
    }
}
```

## EDIT MATERIAL

The following endpoint allows the modification of an existing material in the pharmacy system.

- **URL:**

  `/material/id`

- **Method:**

  `PUT`

- **Data Parameters:**

```
{
    "name": "Example Product",
    "supplier": 2,
    "countryOfProduction": "PR",
    "expirationDate": "2022-12-31",
    "gender": "FEMALE",
    "price": 50.0,
    "ageGroup": "ADULTS",
    "description": "This is an example product description. It can be as
        long or as short as you like."
}
```

- **Success Response:**

```
{
    "msg": "OK",
    "data": true
}
```

- **Error Response:**


Figure 8: Creating Material Error Response

## GET MATERIAL

The following endpoint retrieves the details of a specific material based on its unique ID.

- **URL:**
  `/material/id`
- **Method:**
  `GET`
- **Success Response:**

```
{
    "msg": "OK",
    "data": {
        "id": 6,
        "name": "Example Product",
        "countryOfProduction": "PR",
        "expirationDate": "2022-12-31",
        "image": null,
        "gender": "FEMALE",
        "price": 50.0,
        "ageGroup": "ADULTS",
        "lastModifiedDate": null,
        "description": "This is an example product description. It can
            be as long or as short as you like."
    }
}
```

## GET ALL MATERIAL

The following endpoint retrieves a list of all materials stored in the pharmacy system.

- **URL:**
  `/material`
- **Method:**
  `GET`
- **Success Response:**

```
{
    "msg": "OK",
    "data": [
        {
            "id": null,
            "name": "Example Product",
            "supplier": {
                "id": 2,
                "name": "sup1",
                "address": "here in padova",
                "email": "su1@pd.it",
                "telephoneNumber": "099999",
                "drugs": [],
                "materials": [
                    {
                        "id": 6,
                        "name": "Example Product",
                        "countryOfProduction": "PR",
                        "expirationDate": "2022-12-31",
                        "image": null,
                        "gender": "FEMALE",
                        "price": 50.0,
                        "ageGroup": "ADULTS",
                        "lastModifiedDate": null,
                        "description": "This is an example product
                            description. It can be as long or as short as
                             you like."
                    }
                ]
            },
            "expirationDate": "2022-12-31",
            "image": null,
            "gender": "MALE",
            "ageGroup": "ADULTS",
            "price": 50.0,
            "lastModifiedDate": null,
            "description": "This is an example product description. It
               can be as long or as short as you like.",
            "countryOfProduction": "PR",
            "orders": null,
            "receipts": null
        }
    ]
}
```


## CREATE RECIEPT

The following endpoint allows the creation of a new receipt in the pharmacy system.

- **URL:**

/receipt

- **Method:**
  POST

- **Data Parameters:**

```
{
    "id":0,
    "list_drug_id":[1,2],
    "list_material_id":null,
    "image":[1,0,0,0,1,1],
    "date":"2023-04-27T07:08:30.267Z",
    "paymentMethod":"CASH"
}
```

- **Success Response:**

```
{
    "msg":"OK",
    "data":{
        "id":1,
        "receiptDrugs":[
            {
                "id":1,
                "name":"new drug",
                "expirationDate":"2023-04-23T10:44:38.843+00:00",
                "image":null,
                "shape":"shape",
                "gender":"MALE",
                "ageGroup":"ADULTS",
                "needPrescription":true,
                "description":"string",
                "limitation":10,
                "price":60,
                "countryOFProduction":"AF",
                "lastModifiedDate":null,
                "sensitive":true
            },
            {
                "id":2,
                "name":"new drug",
                "expirationDate":"2023-04-23T10:44:38.843+00:00",
                "image":null,
                "shape":"shape",
                "gender":"MALE",
                "ageGroup":"ADULTS",
                "needPrescription":true,
                "description":"string",
                "limitation":10,
                "price":60,
                "countryOFProduction":"AF",
```

```
            "lastModifiedDate": null,
            "sensitive": true
        }
    ],
    "receiptMaterials": [],
    "image": "AQAAAAEB",
    "date": "2023-04-27T07:08:30.267+00:00",
    "paymentMethod": "CASH"
  }
}
```


- **Error Response**:

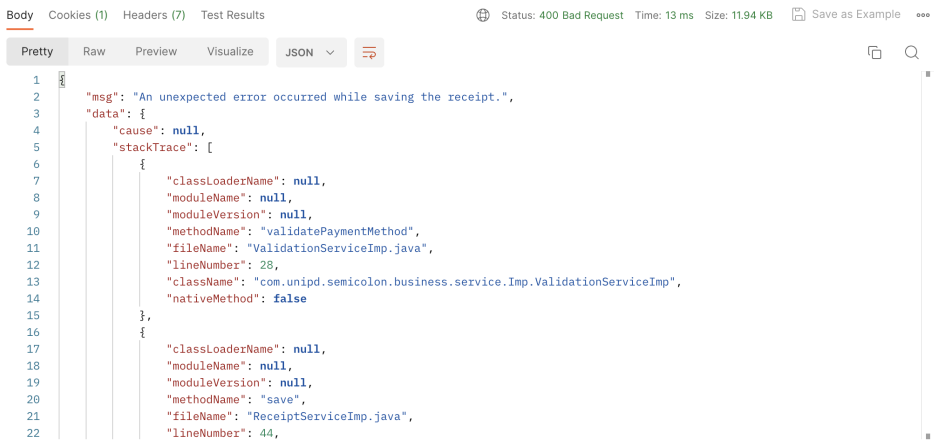


Figure 9: Error Response of Reciept

## EDIT RECIEPT

The following endpoint allows the modification of an existing receipt in the pharmacy system.

- **URL:**
  `/receipt/id`
- **Method:**
  `PUT`
- **Data Parameters:**


```
{
    "id":1,
    "list_drug_id":[1,2,3],
    "list material id":null,
    "material id":0,
    "image":null,
    "date":"2023-04-27T07:42:56.226Z",
    "paymentMethod":"PAYPAL"
}
```


- **Success Response:**

```
{
    "msg":"OK",
    "data":true
}
```


### GET RECIEPT

The following endpoint retrieves the details of a specific receipt based on its unique ID.

- **URL:**
  `/receipt/id`
- **Method:**
  `GET`
- **Success Response:**


```
{
    "msg":"OK",
    "data":{
        "id":1,
        "receiptDrugs":[
            {
                "id":1,
                "name":"new drug",
                "expirationDate":"2023-04-23T10:44:38.843+00:00",
                "image":null,
                "shape":"shape",
                "gender":"MALE",
                "ageGroup":"ADULTS",
                "needPrescription":true,
                "description":"string",
                "limitation":10,
                "price":60,
                "countryOfProduction":"AF",
                "lastModifiedDate":null,
                "sensitive":true
            },
            {
                "id":2,
                "name":"new drug",
                "expirationDate":"2023-04-23T10:44:38.843+00:00",
                "image":null,
                "shape":"shape",
                "gender":"MALE",
                "ageGroup":"ADULTS",
                "needPrescription":true,
                "description":"string",
                "limitation":10,
                "price":60,
                "countryOFProduction":"AF",
```

```
            "lastModifiedDate": null,
            "sensitive": true
        },
        {
            "id": 3,
            "name": "new drug",
            "expirationDate": "2023-04-23T10:44:38.843+00:00",
            "image": null,
            "shape": "shape",
            "gender": "MALE",
            "ageGroup": "ADULTS",
            "needPrescription": true,
            "description": "string",
            "limitation": 10,
            "price": 60,
            "countryOfProduction": "AF",
            "lastModifiedDate": null,
            "sensitive": true
        }
    ],
    "receiptMaterials": [

    ],
    "image": "AQAAAAEB",
    "date": "2023-04-27T07:42:56.226+00:00",
    "paymentMethod": "PAYPAL"
  }
}
```


## CREATE SUPPLIER (JSP)

The following endpoint (Figure 10) allows the creation of a new supplier in the pharmacy system.

| Name | Email | Address | Telephone Number |
|---|---|---|---|
| sup1 | su1@pd.it | here in padova | 099999 |
| sup2 | su2@pd.it | here in padova | 0888888 |
| sup0 | su0@pd.it | pd | 00000 |

Figure 10: List of Supplier

- **URL:**
  `/supplier`
- **Method:**
  `POST`
- **Data Parameters:**


```
{
    "name": "sup4",
    "address": "here in padova",
```

```
    "email": "su4@pd.it",
    "telephoneNumber": "0888888"
}
```

- **Success Response:**

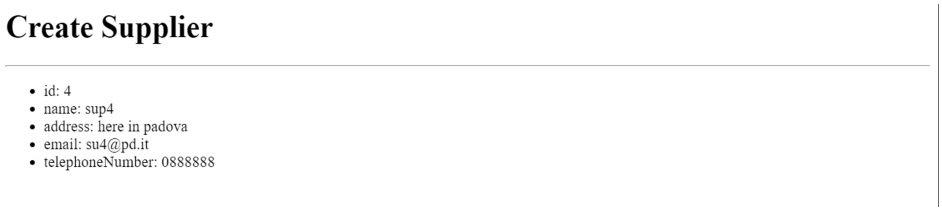


Figure 11: Success Response of Supplier

- **Error Response**:

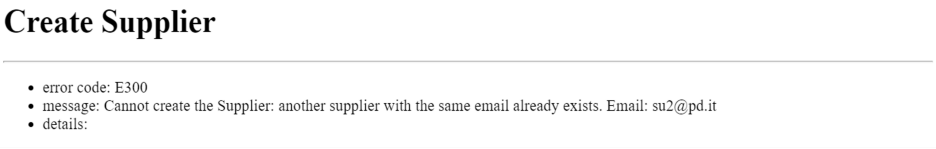


Figure 12: Error Response of Supplier

### EDIT SUPPLIER

The following endpoint modify storage in the system.

- **URL:**
  `/storage/id`
- **Method:**
  `PUT`
- **Success Response:**

```
success response:
{
   msg: "OK",
   data: true
}
```

- **Error Response:**

```
{
    "msg": "permission denied",
    "data": "Token :eyJhbGciOiJIUzI1NiJ9.
       eyJzdWIiOiIxIiwiUm9sZSI6ImFkbWluIiwiaWF0IjoxNjgyMjgyNDg0LCJleHAiOjE3MTgyODI0
       .L8IXQCVJdk_05IYLRSnkkB5kqrSGFS1cPnA7j04kdG8"
}
```

## DELETE SUPPLIER

The following endpoint delete storage in the system.

- **URL:**
  `/storage/id`
- **Method:**
  `DELETE`
- **Success Response:**

```
success response:
{
   msg: "OK",
   data: true
}
```

## GET SUPPLIER

The following endpoint modify storage in the system.

- **URL:**
  `/storage/id`
- **Method:**
  `GET`
- **Data Parameters:**

```
{
    "msg": "OK",
    "data": {
        "id": 1,
        "name": "John",
        "address": "Main 123",
        "email": "john.doe@gmail.com",
        "telephoneNumber": "3558549953",
        "drugs": [],
        "materials": []
    }
}
```

- **Success Response:**

```
{
    "msg": "Supplier Not Found with id1",
    "data": null
}
```

## PHARMACY ACTIVATION

The following endpoint activates a pharmacy in the system.

- **URL:**
  `/pharmacy/id`
- **Method:**
  `PATCH`
- **Data Parameters:**

```
{
    "status" : "ACTIVE"
}
```

- **Success Response:**

```
{
    "msg": "OK",
    "data": {
        "id": 1,
        "name": "pharmacy1",
        "address": "padova",
        "telephoneNumber": "09999999",
        "time_table": [],
        "logo": null,
        "storage": [],
        "staff": [
            {
                "id": 1,
                "name": "jeremy",
                "lastName": "jervis",
                "gender": "MALE",
                "birthDate": null,
                "phoneNumber": "08888888",
                "address": "unipd",
                "role": null,
                "email": "hi@uni.com",
                "accountStatus": "ACTIVE",
                "profilePicture": null
            }
        ],
        "status": "ACTIVE",
        "orders": []
    }
}
```

## PHARMACY CREATION

The following endpoint create a pharmacy in the system.

- **URL:**
  `/pharmacy`
- **Method:**
  `POST`
- **Data Parameters:**

```
{
  "name": "Example Pharmacy1",
  "address": "123 Main St",
  "tellNumber": "3389929820",
  "timeTable": [],
  "logoPath": [],
  "storage": [],
  "staff": [],
  "status": "ACTIVE"
}
```

- **Success Response:**

```
{
    "msg": "OK",
    "data": {
        "id": 2,
        "name": "Example Pharmacy1",
        "address": "123 Main St",
        "telephoneNumber": "3389929820",
        "time_table": null,
        "logo": "",
        "storage": null,
        "staff": null,
        "status": "ACTIVE",
        "orders": null
    }
}
```

- **Error Response:**

```
{
    "msg": "Telephone number should contain only digits",
    "data": "com.unipd.semicolon.business.exception.
       IllegalArgumentException"
}
```

## PHARMACY EDIT

The following endpoint modification a pharmacy in the system.

- **URL:**
  `/pharmacy/id`

- **Method:**
  `PUT`
- **Data Parameters:**

```
{
    "name": "Hpp's Pharmacy2222",
    "tell_number": "3389338922"
}
```

- **Success Response:**

```
{
    "msg": "OK",
    "data": true
}
```

- **Error Response:**

```
{
    "msg": "Telephone number should contain only digits",
    "data": "com.unipd.semicolon.business.exception.
       IllegalArgumentException"
}
```

### PHARMACY DELETE

The following endpoint delete a pharmacy in the system.

- **URL:**
  `/pharmacy/save`
- **Method:**
  `POST`
- **Success Response:**

```
{
    "msg": "OK",
    "data": true
}
```

### PHARMACY GET

The following endpoint get a pharmacy in the system.

- **URL:**
  `/pharmacy/id`
- **Method:**
  `GET`
- **Success Response:**

```
{
    "msg": "OK",
    "data": {
        "id": 1,
        "name": "Hpp's Pharmacy2222",
        "address": "123 Main St",
        "telephoneNumber": "3389338922",
        "time_table": [],
        "logo": "",
        "storage": [],
        "staff": [
            {
                "id": 4,
                "name": "John",
                "lastName": "Doe",
                "gender": "MALE",
                "birthDate": "1995-06-30T00:00:00",
                "phoneNumber": "3358962012",
                "address": "123 Main St.",
                "role": {
                    "id": 2,
                    "role": "user"
                },
                "email": "john.doe@example.com",
                "accountStatus": "ACTIVE",
                "profilePicture": null
            }
        ],
        "status": "ACTIVE",
        "orders": []
    }
}
```


## ACCOUNT LOGIN

The following endpoint allows users to log in to their account in the pharmacy system.

- **URL:**
  `/account`
- **Method:**
  `GET`
- **Data Parameters:**

```
{
    "username" : "STRING",
    "password" : "STRING"
}
```

- **Success Response:**

```
{
    "msg": "OK",
    "data": {
        "username": "STRING",
        "lastLoginDate": "2023-04-27T00:48:27",
        "user": {
            "id":  LONG ,
            "name": "STRING",
            "lastName": "STRING",
            "gender": MALE,
            "birthDate": LOCALDATE,
            "phoneNumber": "STRING",
            "address": "STRING",
            "role": {
                "id": LONG,
                "role": "STRING"
            },
            "email": "STRING",
            "accountStatus":  "STRING",
            "profilePicture": byte[]
        },
        "token": "STRING"
    }
}
```

## ACCOUNT LOGOUT

The following endpoint allows users to log out of their account in the pharmacy system.

- **URL:**
  `/account/token`
- **Method:**
  `POST`
- **Success Response:**

```
{
    "msg": "OK",
    "data": true
}
```

# 6 Group Members Contribution

Throughout the project lifecycle, the team conducted multiple meetings to ensure that the project was executed successfully. The initial stages involved selecting a pertinent subject and devising an appropriate implementation strategy. The team engaged in extensive deliberations to determine the project requirements and defined all relevant classes and entities. Subsequently, tasks were assigned to individual team members using the Trello platform. Throughout the project duration, each team member fulfilled specific responsibilities that were critical to the success of the endeavor. Through collaborative efforts, the team developed a dependable and robust system that met the expectations of both users and stakeholders. Their commitment to working together allowed for effective completion of various tasks, leading to the successful execution of the project. In the following paragraphs, a detailed account of each team member's responsibilities is provided.

**Reihaneh Baghishani** She created Sequence Diagrams and built entities like Drug for the project. She implemented servlets, JSPs, and RESTful APIs using Spring Boot framework, designed DAO and service layers, and created filters to enhance the functionality of Drugs and Materials modules. Additionally, she contributed to exception handling, documented APIs, modified the designs to conform to RESTful API standards, and helped implement certain methods for the Drug and Material entities.

**Ali Erfanian Omidvar** He had a diverse set of responsibilities, such as defining the system structure and creating a response helper for APIs. Additionally, he was responsible for fixing any issues related to the relationship between entities and creating a security service for the system. He also contributed to refactoring the order entity, service, and repository and created a log service to ensure system accuracy. Finally, he reviewed the merge request on git and refactored the URLs.

**Irem Goksu Celik** In her role, she was responsible for implementing the supplier and login entity and repository. Additionally, she developed the order CRUD API and prepared the class diagram. She also contributed to the implementation of storage reports for the system.

**Hamideh Haeri** Her role encompassed various responsibilities, including creating the Entity Relationship diagram, defining the attributes and methods of the Pharmacy entity, and developing the Receipt API. She was also responsible for creating the project document and fixing API issues, as well as adding validation checks to ensure system accuracy. Additionally, she contributed to the implementation of some methods in the User module.

**Ferdos Hajizadeh Chavari** Her role encompassed various tasks, such as implementing and designing the material entity and repository, developing the drug API, and implementing the payment method. Additionally, she was responsible for editing the project document to ensure system accuracy.

**Hossein Hosseinpour** As part of his role, he was responsible for defining the attributes and methods of the Storage entity and developing the Pharmacy API. He also created the sequence diagram for the Pharmacy save API and was responsible for fixing any related API issues and adding validation checks to the system. Additionally, he contributed to the implementation of some methods in the User module.

**Gulce Sirvanci** She had multiple responsibilities, including implementing the user entity and repository, supplier repository, and supplier CRUD API. Furthermore, she was responsible for the implementation of storage reports for the amount of each drug and material, as well as the threshold parameters.

**Ali Tahvildari** He had a diverse set of responsibilities, which included designing and implementing the order entity and fixing any related issues. He also collaborated with others to implement Swagger and design and implement the wireframe for the system layout and contributed to the implementation of the material and user API methods, as well as some methods in the User module.

**Mohammad Torabi** As part of his responsibilities, he created the Receipt entity and implemented the Storage module of the back-end system using Spring Boot. To achieve this, he had to design and implement the API, repository, and service layers for the Storage entity. In addition, he contributed to the implementation of some methods in the User and Receipt modules and edited the project documentation.