



POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Progetto TIW versione HTML

PROGETTO DI TIW
INGEGNERIA INFORMATICA

Studente: **Yanfeng Li**

Codice Persona: 10650552
Prof: Prof. Fraternali Piero, Prof. Federico Milani
Anno Accademico: 2021-2022

Contents

Contents	i
1 Introduction	1
1.1 Specifications	1
2 Database	3
2.1 Textual Analysis	3
2.2 Database Design	4
2.3 Database Schema	4
2.3.1 Users	5
2.3.2 Preventives	5
2.3.3 Products	6
2.3.4 Options	6
2.3.5 Preventive_ops	6
2.3.6 Product_ops	7
3 Application	9
3.1 Application Requirements Analysis	9
4 Componenti	11
4.1 Model Objects (Beans)	11
4.2 Data Access Objects (DAO)	11
4.3 Filters	12
4.4 Controllers (servlets) [access rights]	12
4.5 Views (Templates)	12
5 Design Applicativo(IFML)	15
6 Sequence Diagrams of Interactions	17
6.1 CheckLoggedUser	18

6.2	CheckNotLoggedUser	19
6.3	CheckLoggedAdmin	20
6.4	CheckLoggedClient	21
6.5	Login	22
6.6	Logout	23
6.7	Signup	24
6.8	GoToPreventivoCreate	25
6.9	GoToHomeClient	26
6.10	GoToHomeAdmin	27
6.11	GoToPreventivoDetail	28
6.12	GoToPreventivoPrezza	29
6.13	GoToSignup	30
6.14	SendPreventivo	30
6.15	SendPrezzo	31

1 | Introduction

1.1. Specifications

EX 2: Quotation Management

A web application allows the management of quotation requests for customized products. The application supports registration and login for customers and employees through a public page with appropriate forms. Registration checks the uniqueness of the username. A quotation is associated with a product, the customer who requested it, and the employee who handled it. The quotation includes one or more options for the associated product, which must be among those available for the product. A product has a code, an image, and a name. An option has a code, a type ("standard", "on offer"), and a name. A quotation has a price, determined by the employee.

When the user (customer or employee) accesses the application, a LOGIN PAGE appears, where the user authenticates with a username and password. When a customer logs in, they are directed to a HOME PAGE CLIENTE, which contains a form to create a quotation and a list of quotations created by the customer. By selecting one of the quotations, the customer can view its details. Through the quotation creation form, the user first selects the product; once the product is chosen, the form displays the options available for that product. The user selects the options (at least one) and confirms the submission of the quotation by clicking the INVIA PREVENTIVO button.

When an employee logs in, they are directed to a HOME PAGE IMPIEGATO, which contains a list of quotations previously managed by them and a list of quotations not yet assigned to any employee. When the employee selects an item from the list of unassigned quotations, a PREZZA PREVENTIVO page appears, showing the customer's data (username) and the quotation details, along with a form to enter the quotation price. When the employee enters the price and submits the data by clicking the INVIA PREZZO button, the HOME PAGE IMPIEGATO reappears with updated lists of quotations. The price set by the employee becomes visible to the customer when they access their list of quotations and view the quotation details.

The PREZZA PREVENTIVO page also includes a link to return to the HOME PAGE IMPIEGATO. The application allows the user to log out.

2 | Database

2.1. Textual Analysis

1. Entities
2. Attributes
3. Relationships

A web application allows the management of quotation requests for customized products. The application supports registration and login of **customers** and **employees** through a public page with appropriate forms. Registration checks the uniqueness of the username. A quotation is associated with a product, the customer who requested it, and the employee who handled it. The quotation includes one or more options for the associated product, which must be among those available for the product. A product has a **code**, an **image**, and a **name**. An option has a **code**, a **type** (“standard”, “on offer”), and a **name**. A quotation has a **price**, determined by the employee.

When the user (customer or employee) accesses the application, a LOGIN PAGE appears, where **the user** authenticates with a **username** and **password**. When a customer logs in, they are directed to a HOME PAGE CLIENTE, which contains a form to create a quotation and a list of quotations created by the customer. By selecting one of the quotations, the customer can view its details. Through the quotation creation form, the user first selects the product; once the product is chosen, the form displays the options available for that product. The user selects the options (at least one) and confirms the submission of the quotation by clicking the INVIA PREVENTIVO button.

When an employee logs in, they are directed to a HOME PAGE IMPIEGATO, which contains a list of quotations previously managed by them and a list of quotations not yet assigned to any employee. When the employee selects an item from the list of unassigned quotations, a PREZZA PREVENTIVO page appears, showing the customer's data (username) and the quotation details, along with a form to enter the quotation price. When the employee enters the price and submits the data by clicking the INVIA PREZZO but-

ton, the HOME PAGE IMPIEGATO reappears with updated lists of quotations. The price set by the employee becomes visible to the customer when they access their list of quotations and view the quotation details.

The PREZZA PREVENTIVO page also includes a link to return to the HOME PAGE IMPIEGATO. The application allows the user to log out.

2.2. Database Design

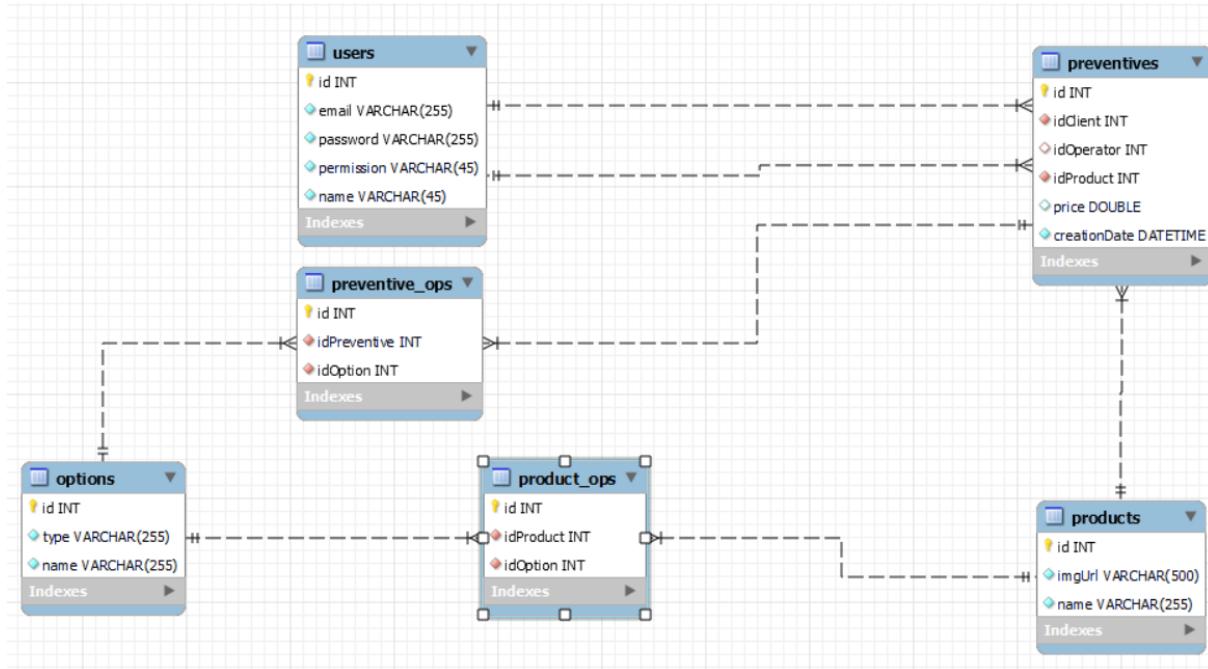


Figure 2.1: Entity-Relationship Diagram

2.3. Database Schema

2.3.1. Users

```
CREATE TABLE `users` (
  `id` int NOT NULL AUTO_INCREMENT,
  `email` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `permission` varchar(45) NOT NULL,
  `name` varchar(45) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `email_UNIQUE` (`email`)
) ENGINE=InnoDB AUTO_INCREMENT=12 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Figure 2.2: Users

2.3.2. Preventives

```
CREATE TABLE `preventives` (
  `id` int NOT NULL AUTO_INCREMENT,
  `idClient` int NOT NULL,
  `idOperator` int DEFAULT NULL,
  `idProduct` int NOT NULL,
  `price` double DEFAULT NULL,
  `creationDate` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`),
  KEY `idOperator_idx` (`idOperator`),
  KEY `id_product_idx` (`idProduct`),
  KEY `idClient_idx` (`idClient`),
  CONSTRAINT `idClient` FOREIGN KEY (`idClient`) REFERENCES `users` (`id`) ON UPDATE CASCADE,
  CONSTRAINT `idOperator` FOREIGN KEY (`idOperator`) REFERENCES `users` (`id`) ON UPDATE CASCADE,
  CONSTRAINT `idProduct` FOREIGN KEY (`idProduct`) REFERENCES `products` (`id`) ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=34 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Figure 2.3: Preventives

2.3.3. Products

```
CREATE TABLE `products` (
  `id` int NOT NULL AUTO_INCREMENT,
  `imgUrl` varchar(500) NOT NULL,
  `name` varchar(255) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Figure 2.4: Products

2.3.4. Options

```
CREATE TABLE `options` (
  `id` int NOT NULL AUTO_INCREMENT,
  `type` varchar(255) NOT NULL,
  `name` varchar(255) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Figure 2.5: Options

2.3.5. Preventive_ops

```
CREATE TABLE `preventive_ops` (
  `id` int NOT NULL AUTO_INCREMENT,
  `idPreventive` int NOT NULL,
  `idOption` int NOT NULL,
  PRIMARY KEY (`id`),
  KEY `idPreventive_idx` (`idPreventive`),
  KEY `idOp_idx` (`idOption`),
  CONSTRAINT `idOp` FOREIGN KEY (`idOption`) REFERENCES `options` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `idPreventive` FOREIGN KEY (`idPreventive`) REFERENCES `preventives` (`id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=51 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Figure 2.6: Preventive_ops

2.3.6. Product_ops

```
CREATE TABLE `product_ops` (
  `id` int NOT NULL AUTO_INCREMENT,
  `idProduct` int NOT NULL,
  `idOption` int NOT NULL,
  PRIMARY KEY (`id`),
  KEY `id_product_idx` (`idProduct`),
  KEY `idOption_idx` (`idOption`),
  CONSTRAINT `id_product` FOREIGN KEY (`idProduct`) REFERENCES `products` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `idOption` FOREIGN KEY (`idOption`) REFERENCES `options` (`id`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

Figure 2.7: Product_ops

3 | Application

3.1. Application Requirements Analysis

1. Pages (view)
2. view components
3. events
4. actions

A web application allows the management of quotation requests for customized products. The application supports registration and login for customers and employees through a public page with appropriate forms. Registration checks the uniqueness of the username. A quotation is associated with a product, the customer who requested it, and the employee who handled it. The quotation includes one or more options for the associated product, which must be among those available for the product. A product has a code, an image, and a name. An option has a code, a type (“standard”, “on offer”), and a name. A quotation has a price, determined by the employee.

When the user (customer or employee) accesses the application, a **LOGIN PAGE** appears, where **the user authenticates with a username and password**. When a customer performs **login**, they are directed to a **HOME PAGE CLIENTE**, which contains a **form to create a quotation** and a **list of quotations created by the customer**. By selecting one of the quotations, the customer **views its details**. Through the **quotation creation form**, the user first **selects the product**; once the product is chosen, **the form displays the options available for that product**. The user selects the options (at least one) and confirms the submission of the quotation by clicking the **INVIA PREVENTIVO** button.

When an employee logs in, they are directed to a **HOME PAGE IMPIEGATO**, which contains a **list of quotations previously managed by them** and a **list of quotations not yet assigned to any employee**. When the employee **selects an item from the list of unassigned quotations**, a **PREZZA PREVENTIVO** page appears, which **displays the customer's data (username)** and the **quotation details** and includes a **form to enter the quotation price**.

When the employee enters the price and submits the data by clicking the INVIA PREZZO button, the HOME PAGE IMPIEGATO reappears with updated lists of quotations. The price set by the employee becomes visible to the customer when they access their list of quotations and view the quotation details.

The PREZZA PREVENTIVO page also includes a link to return to the HOME PAGE IMPIEGATO. The application allows the user to log out.

4 | Componenti

4.1. Model Objects (Beans)

- User
- Preventive
- Product
- Option

4.2. Data Access Objects (DAO)

- UserDAO
 - findUserByCredentials(email, password) : User
 - signupUser(name, email, password, permission) : void
 - getUserByEmail(email) : User
- ClientDAO
 - getPreventivesByUserId() : List<Preventive>
 - sendPreventive(productId, options) : void
- AdminDAO
 - getAdminPreventives() : List<Preventive>
 - getAllAvailablePreventive() : List<Preventive>
 - sendPreventive(preventiveId, price) : void
- PreventiveDAO
 - getPreventiveById() : Preventive
 - getOptionsById() : List<Option>

- ProductDao
 - getAllProduct() : List<Product>
 - getProductById(productId) : Product
 - getProductOptions(productId) : List<Option>

4.3. Filters

- CheckAdmin
- CheckClient
- CheckNotLoggedUser
- CheckLoggedUser

4.4. Controllers (servlets) [access rights]

- Login [Not Logged User]
- Logout [ALL]
- Signup [Not Logged User]
- SendPreventivo [User, Client]
- SendPrezzo [User, Admin]
- GoToPreventivoCreate [User, Client]
- GoToHomeClient [User, Client]
- GoToHomeAdmin [User, Admin]
- GoToPreventivoPrezza [User, Admin]
- GoToPreventivoDetail [User]
- GoToLogin [Not Logged User]
- GoToSignUp [Not Logged User]

4.5. Views (Templates)

- errorPage.html

- formPreventivo.html
- homeAdmin.html
- homeClient.html
- login.html
- preventivoDetail.html
- prezzaPreventivo.html
- prezzaPreventivo.html
- signup.html
- userStatus.html (Included in the pages after login)

5 | Design Applicativo(IFML)

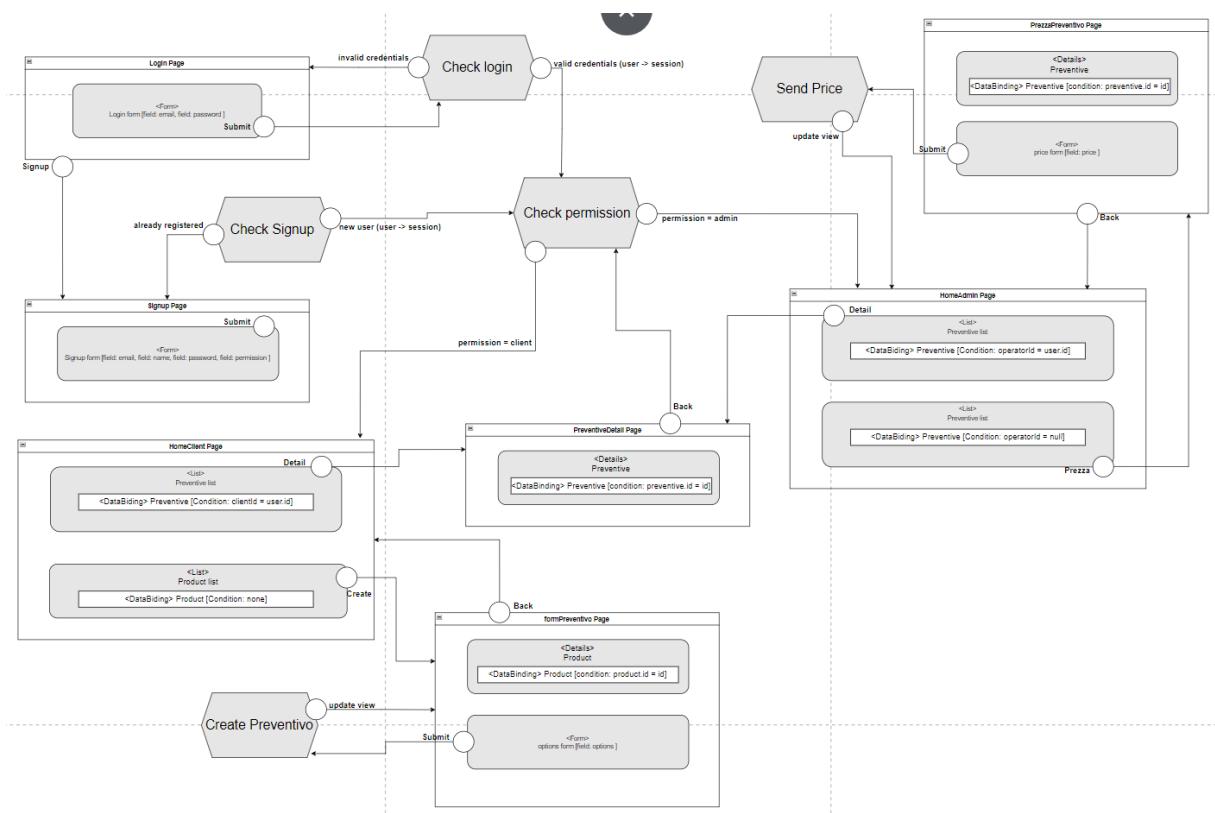


Figure 5.1: Design Applicativo(IFML)

6 | Sequence Diagrams of Interactions

6.1. CheckLoggedUser

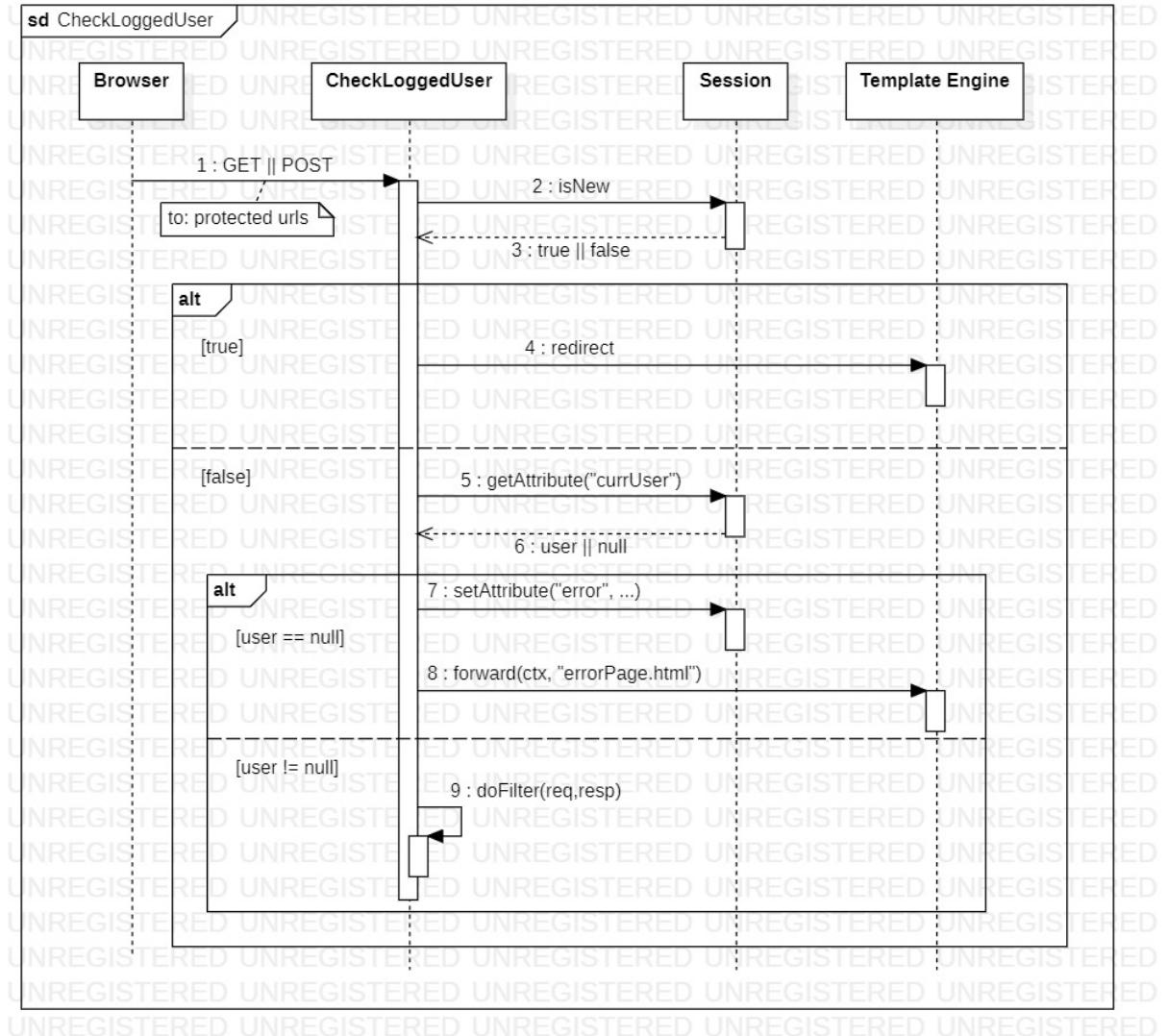


Figure 6.1: CheckLoggedUser

6.2. CheckNotLoggedUser

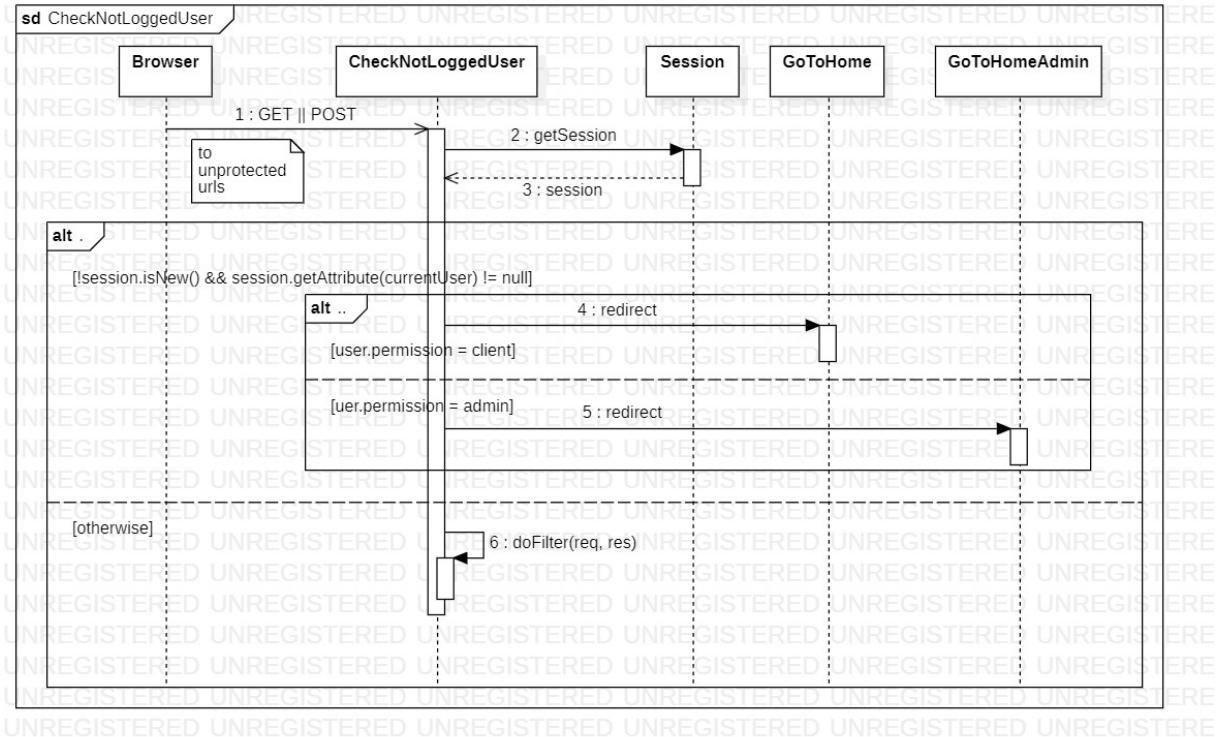


Figure 6.2: CheckNotLoggedUser

6.3. CheckLoggedAdmin

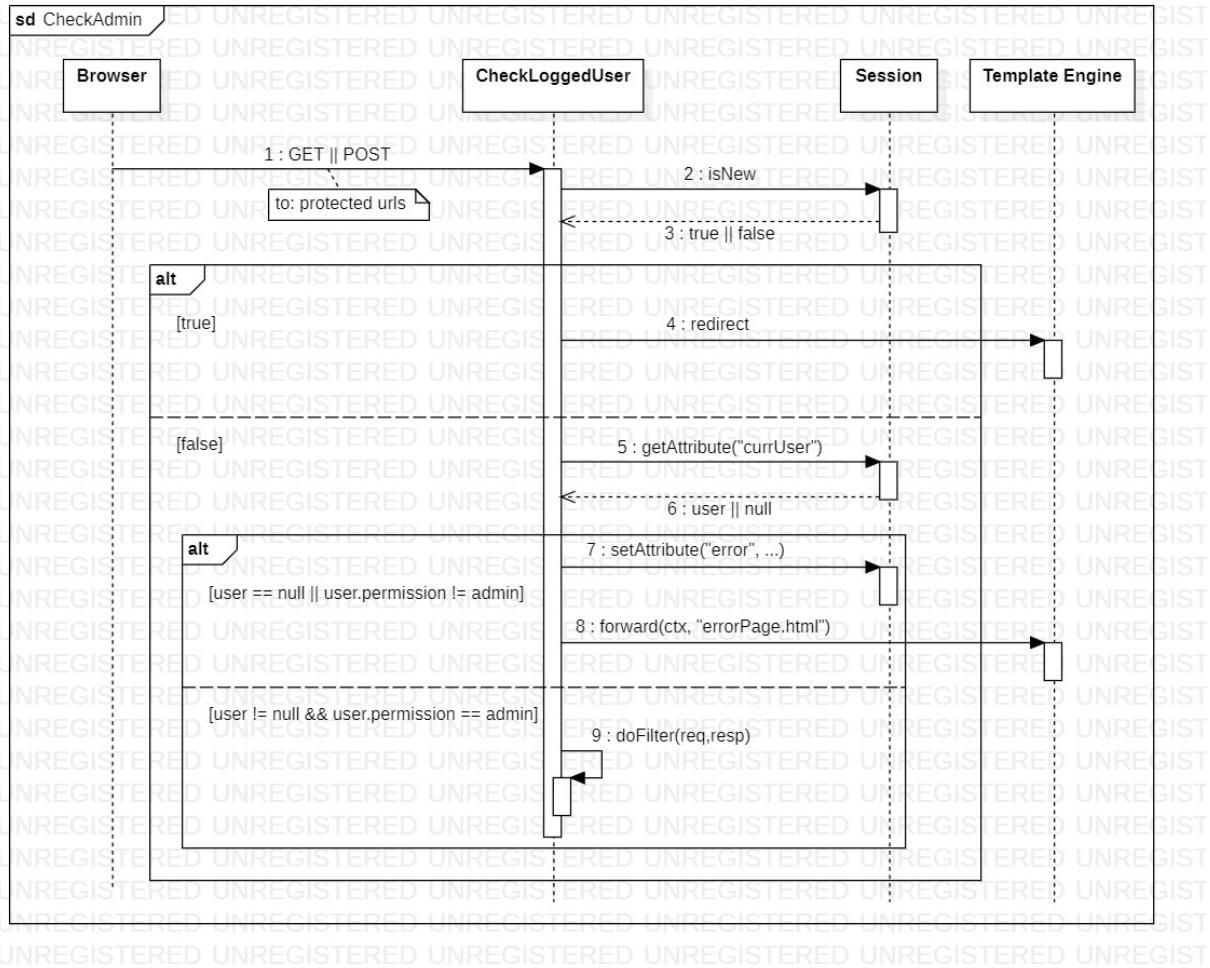


Figure 6.3: CheckLoggedAdmin

6.4. CheckLoggedClient

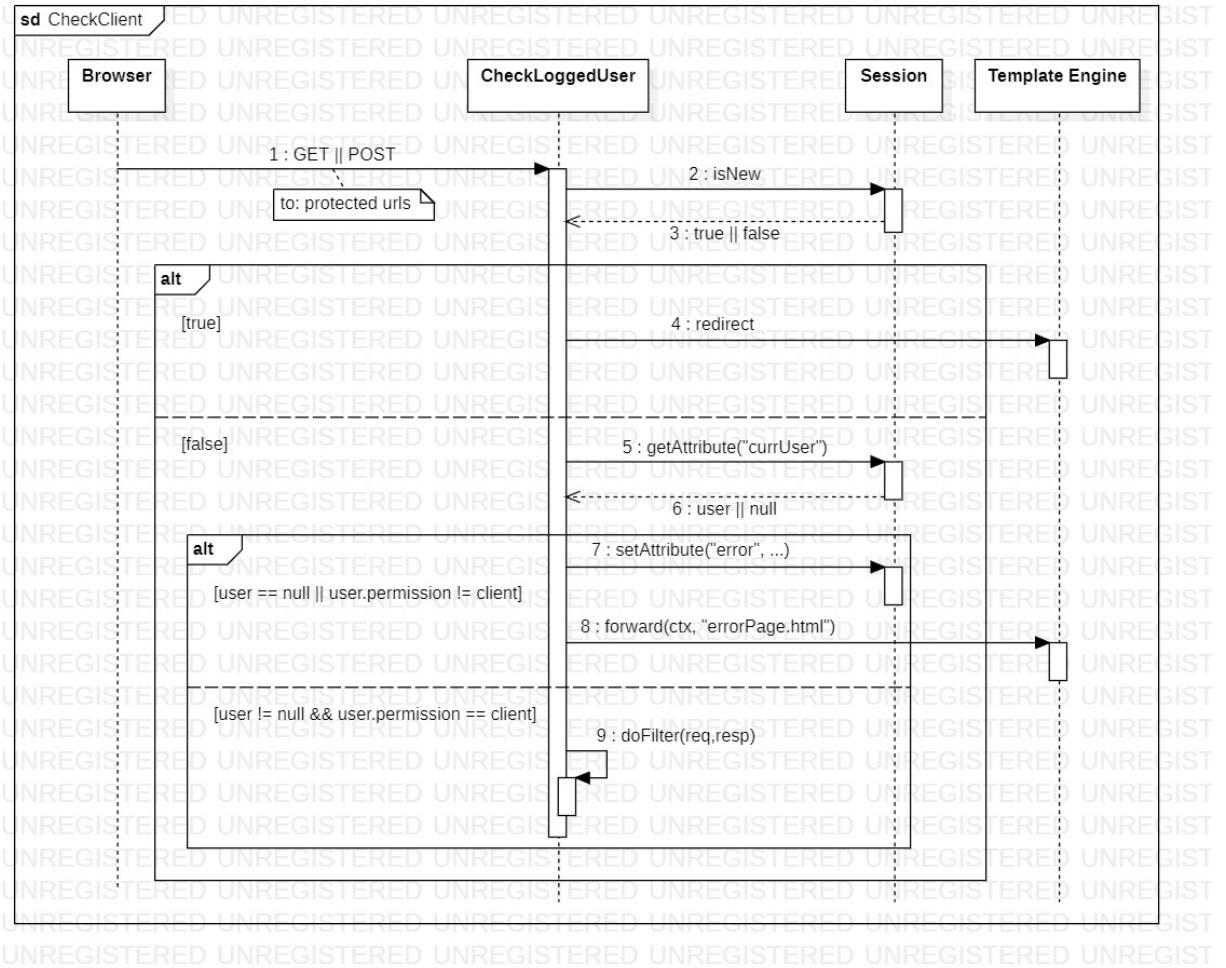


Figure 6.4: CheckLoggedClient

6.5. Login

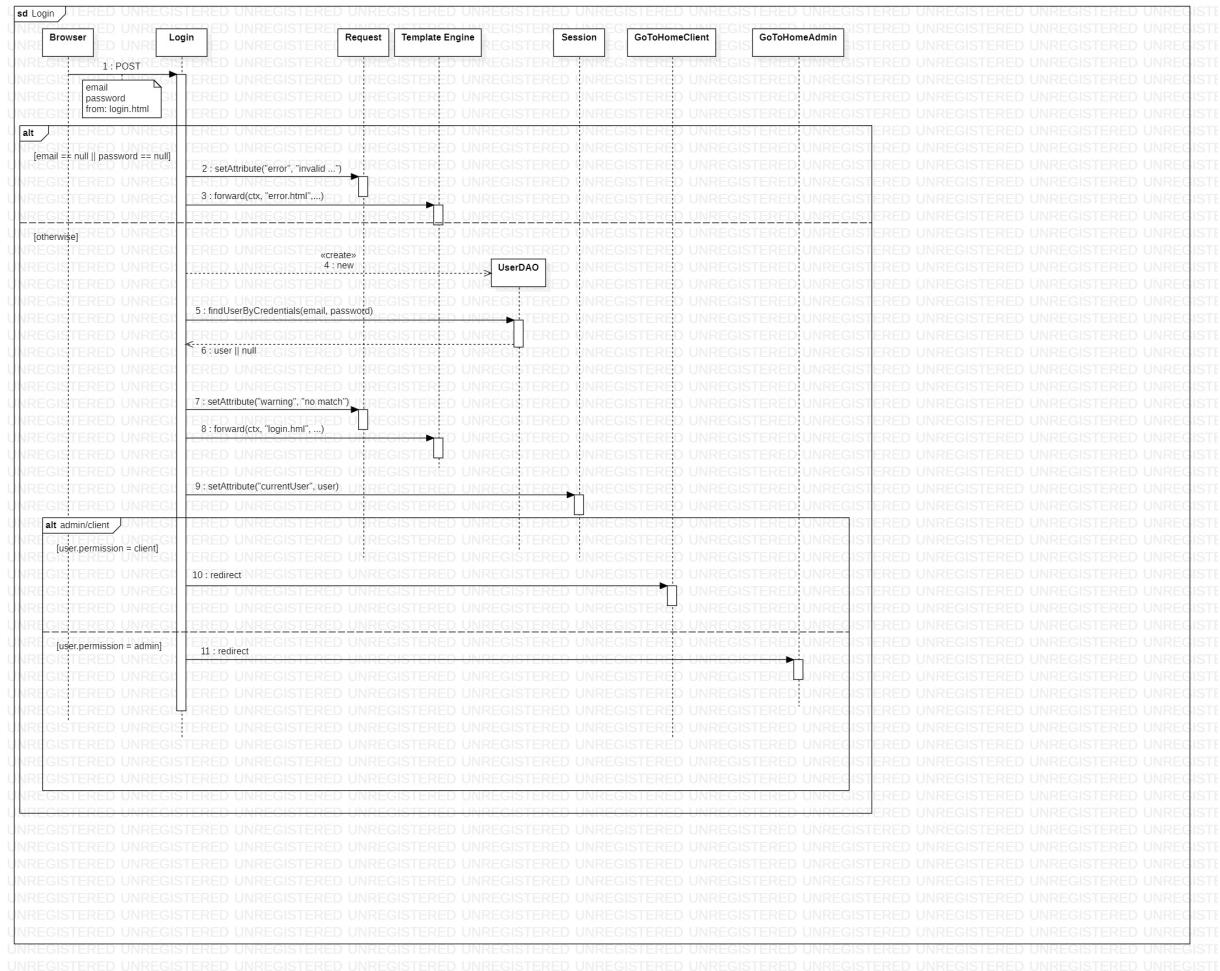


Figure 6.5: Login

6.6. Logout

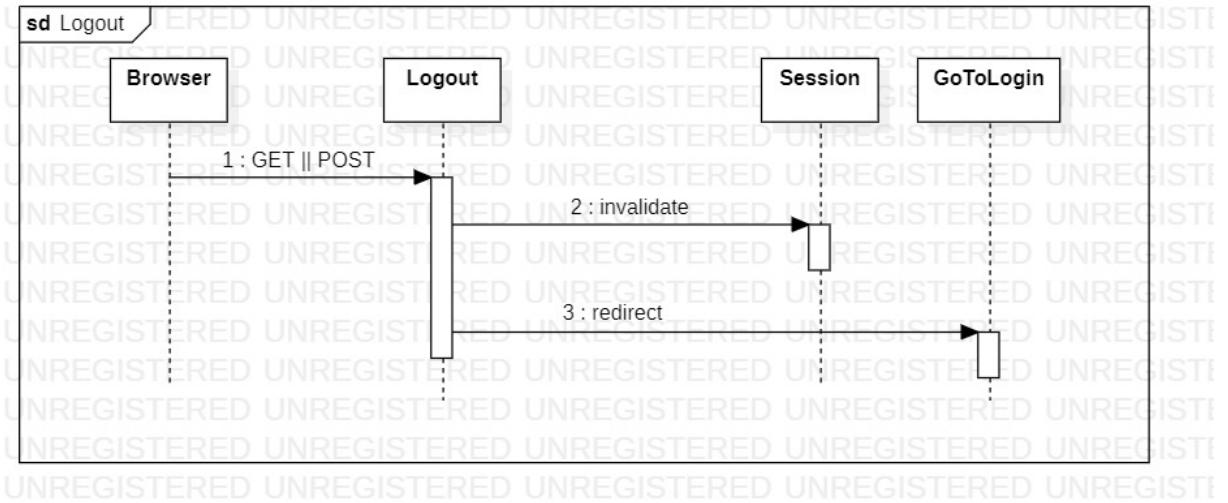


Figure 6.6: Logout

6.7. Signup

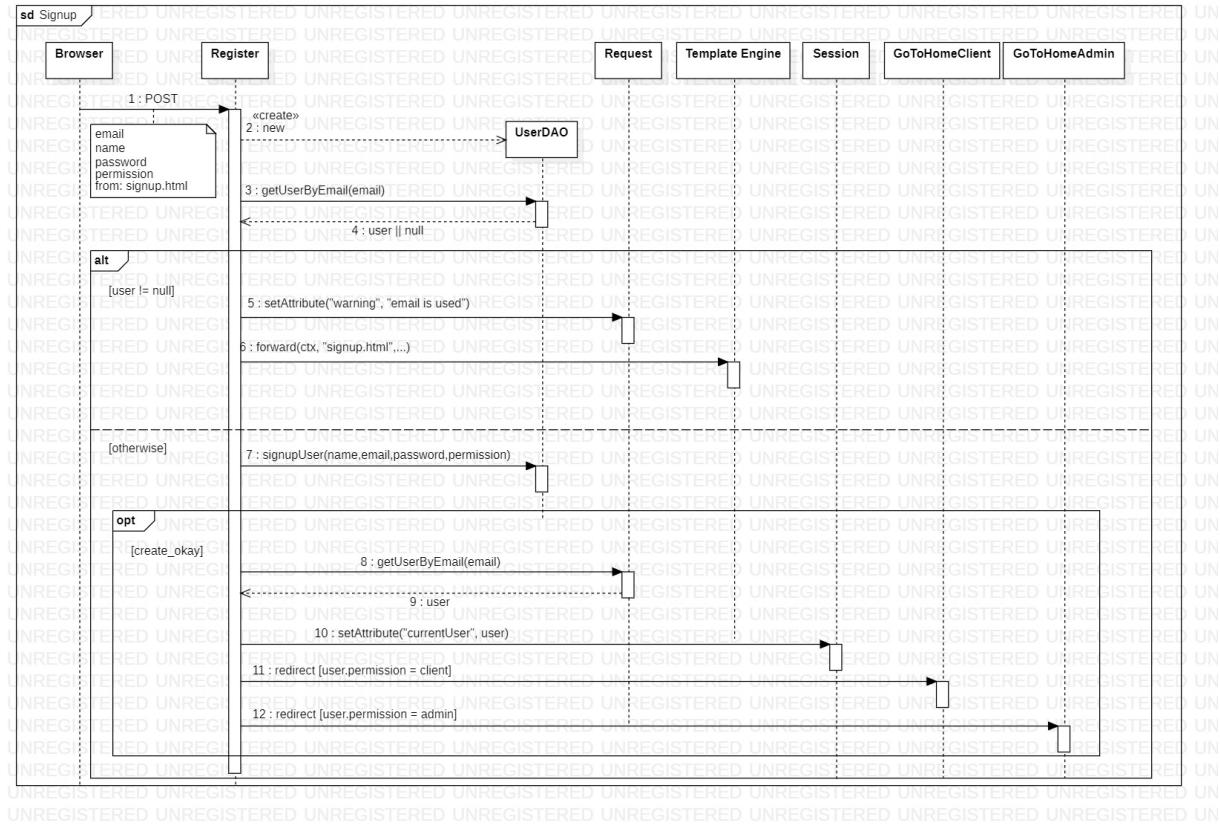


Figure 6.7: Signup

6.8. GoToPreventivoCreate

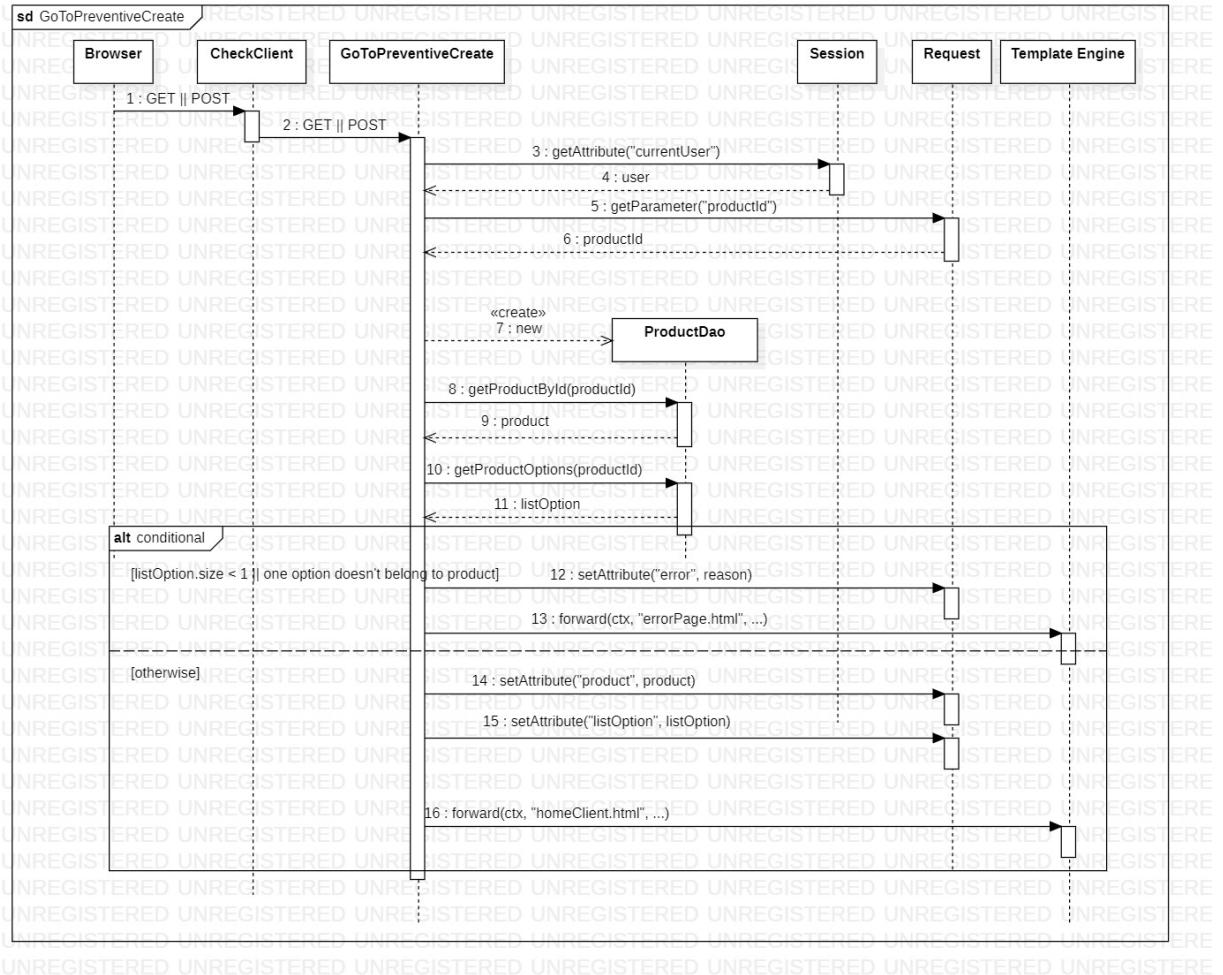


Figure 6.8: GoToPreventivoCreate

6.9. GoToHomeClient

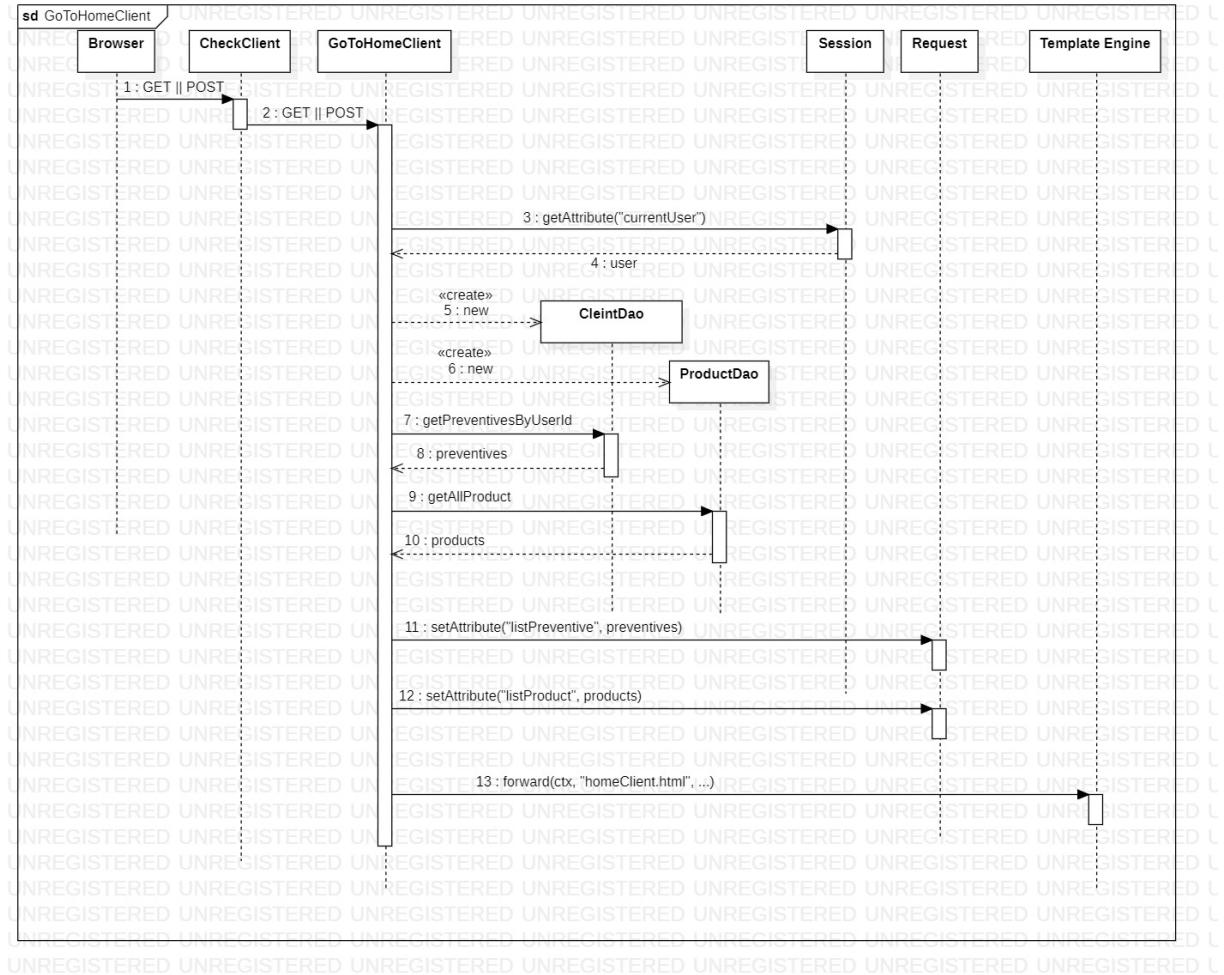


Figure 6.9: GoToHomeClient

6.10. GoToHomeAdmin

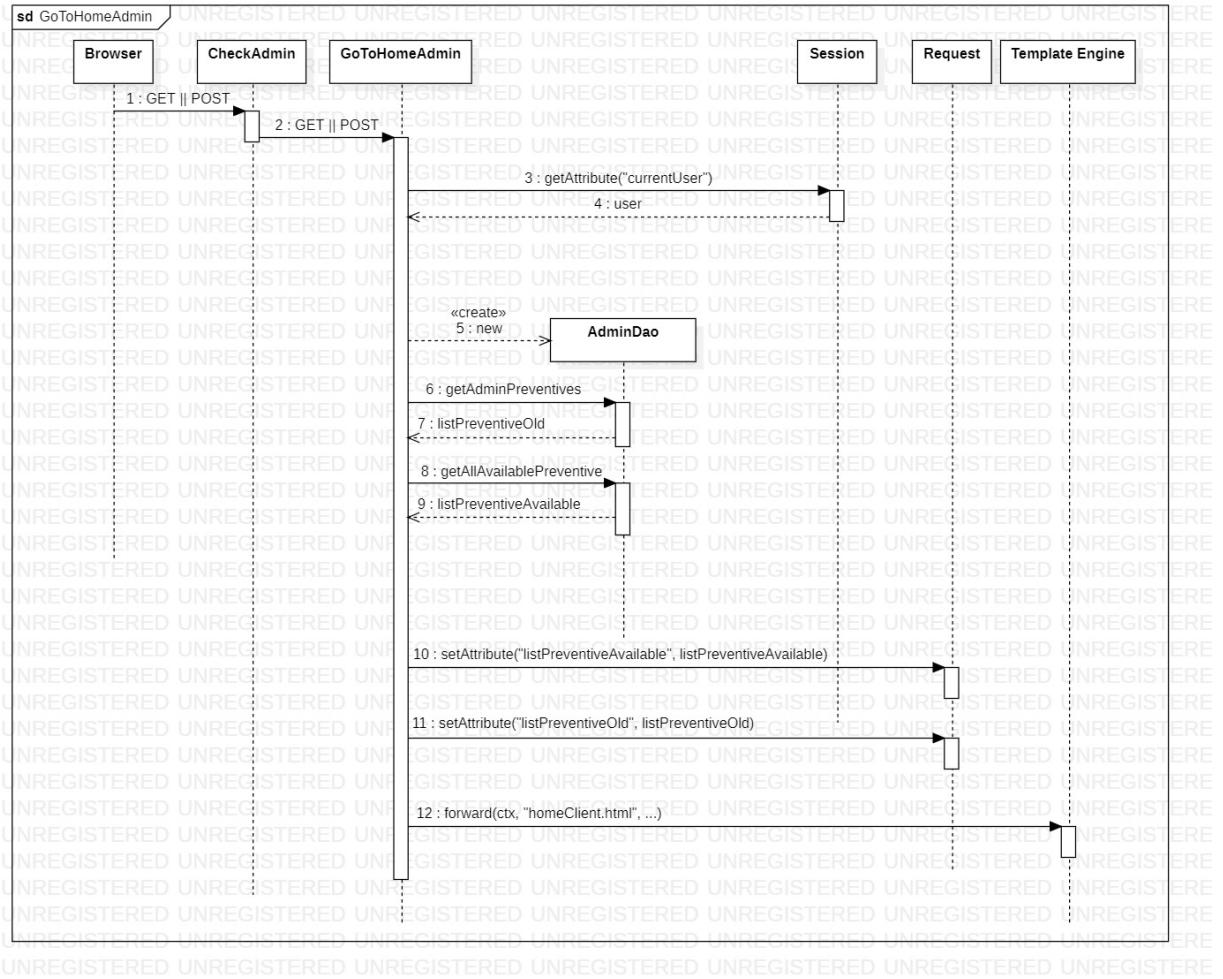


Figure 6.10: GoToHomeAdmin

6.11. GoToPreventivoDetail

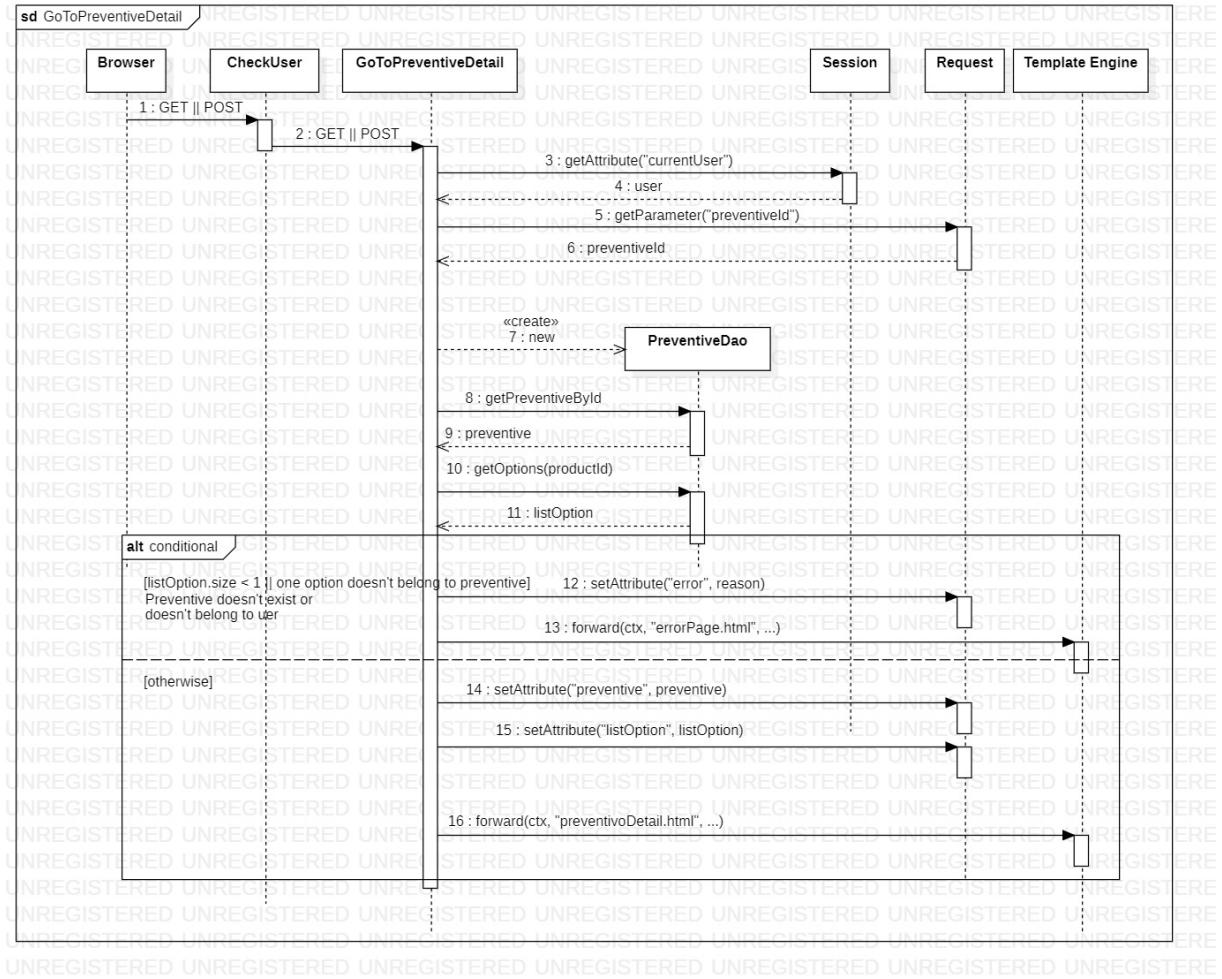


Figure 6.11: GoToPreventivoDetail

6.12. GoToPreventivoPrezza

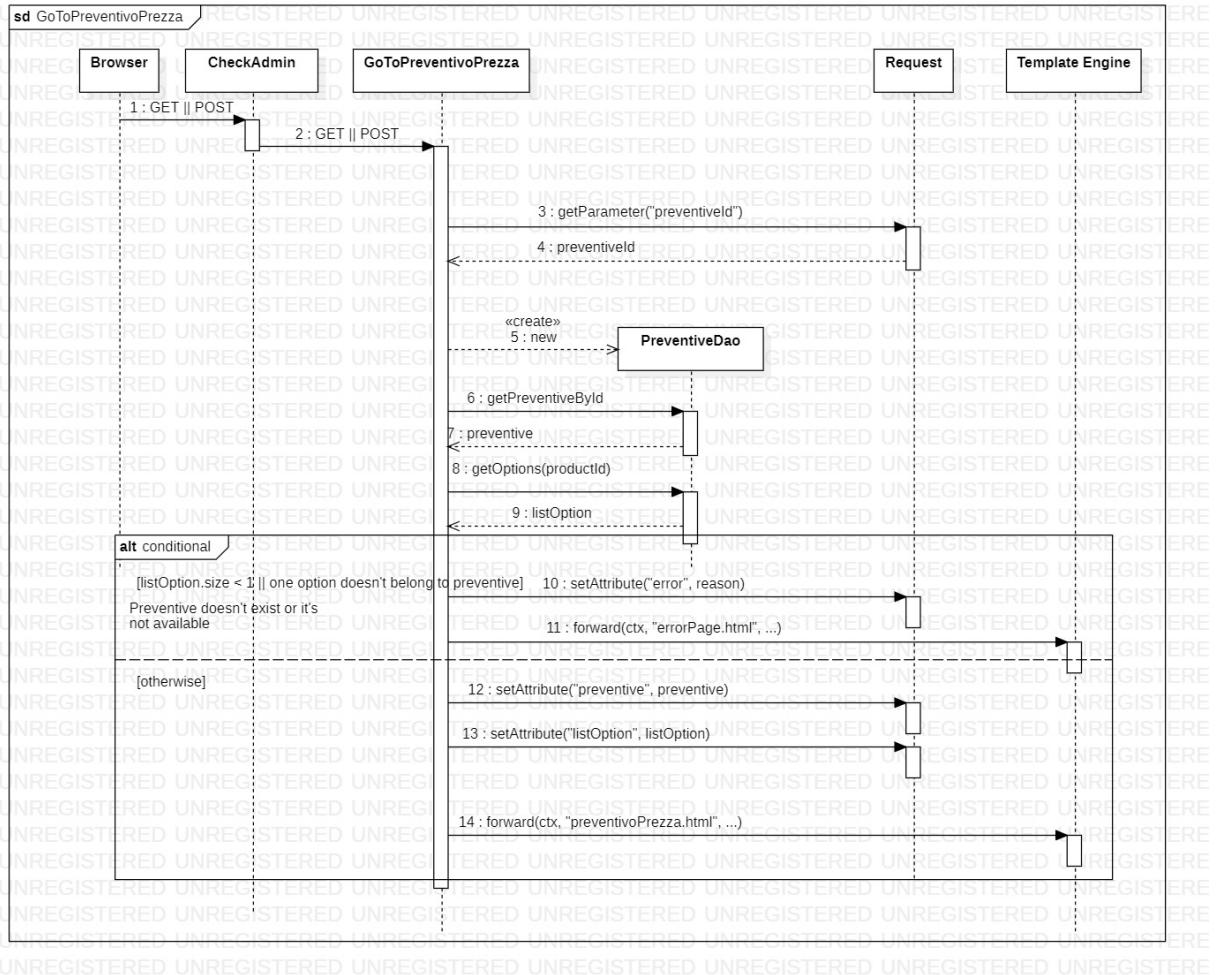


Figure 6.12: GoToPreventivoPrezza

6.13. GoToSignup

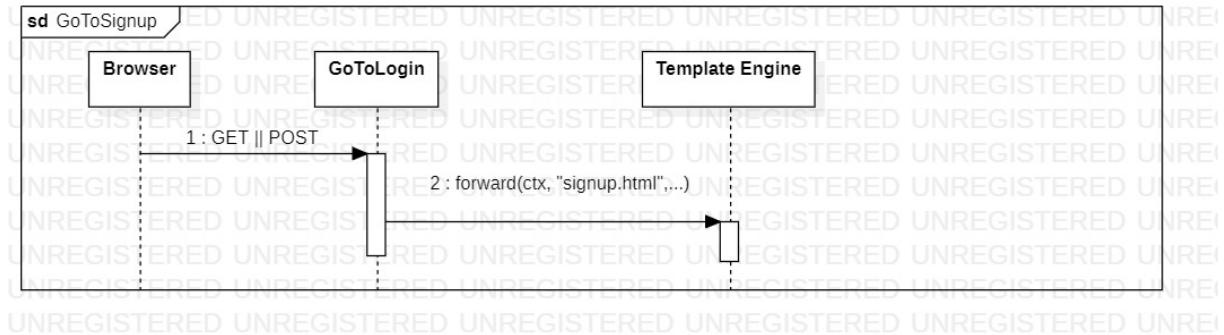


Figure 6.13: GoToSignup

6.14. SendPreventivo

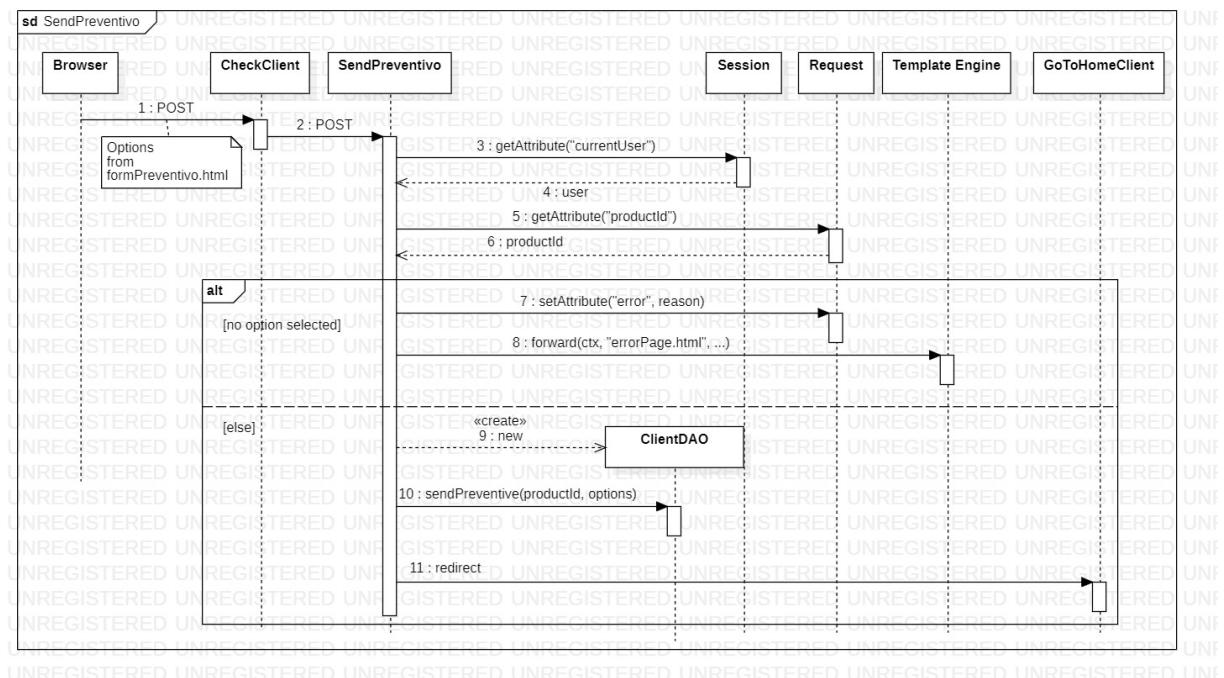


Figure 6.14: SendPreventivo

6.15. SendPrezzo

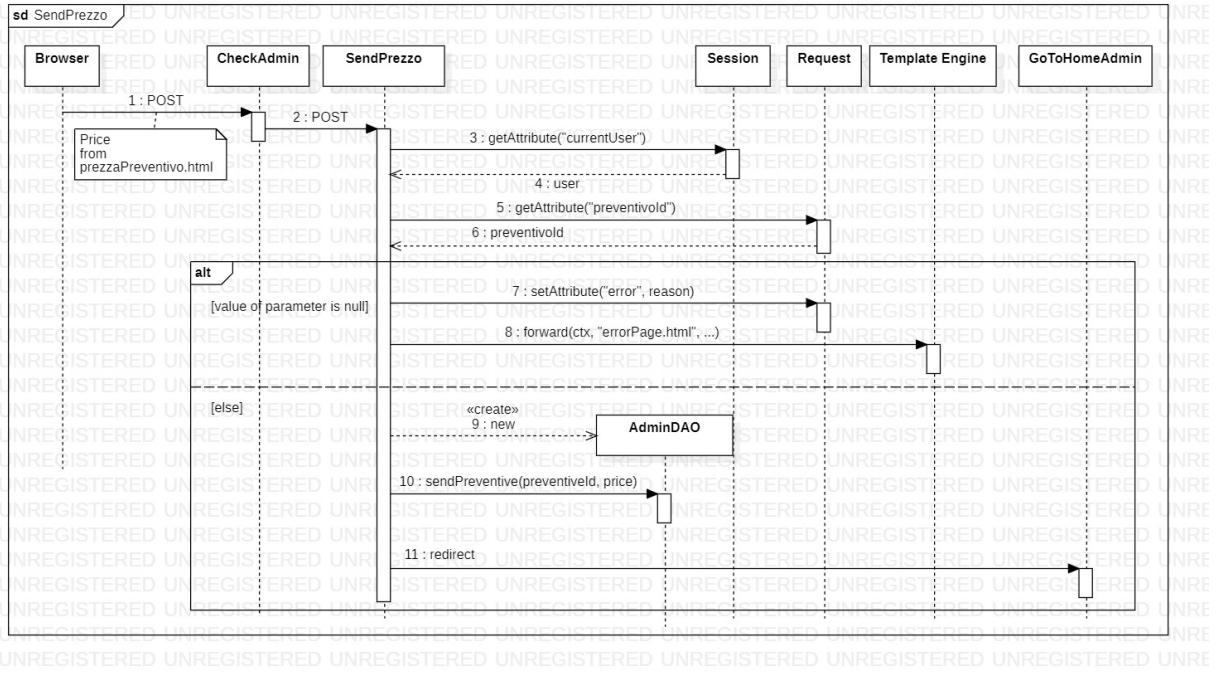


Figure 6.15: SendPrezzo