



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 Introduzione	1
1.1 Specifiche	1
2 Database	3
2.1 Analisi Testuale	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 Applicazione	9
3.1 Analisi Requisiti Applicazione	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 di Interazioni	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 | Introduzione

1.1. Specifiche

Esercizio 2: Gestione Preventivi

Un'applicazione web consente la gestione di richieste di preventivi per prodotti personalizzati. L'applicazione supporta registrazione e login di clienti e impiegati mediante una pagina pubblica con opportune form. La registrazione controlla l'unicità dello username. Un preventivo è associato a un prodotto, al cliente che l'ha richiesto e all'impiegato che l'ha gestito. Il preventivo comprende una o più opzioni per il prodotto a cui è associato, che devono essere tra quelle disponibili per il prodotto. Un prodotto ha un codice, un'immagine e un nome. Un'opzione ha un codice, un tipo (“normale”, “in offerta”) e un nome. Un preventivo ha un prezzo, definito dall'impiegato. Quando l'utente (cliente o impiegato) accede all'applicazione, appare una LOGIN PAGE, mediante la quale l'utente si autentica con username e password. Quando un cliente fa login, accede a una pagina HOME PAGE CLIENTE che contiene una form per creare un preventivo e l'elenco dei preventivi creati dal cliente. Selezionando uno dei preventivi il cliente ne visualizza i dettagli. Mediante la form di creazione di un preventivo l'utente per prima cosa sceglie il prodotto; scelto il prodotto, la form mostra le opzioni di quel prodotto. L'utente sceglie le opzioni (almeno una) e conferma l'invio del preventivo mediante il bottone INVIA PREVENTIVO. Quando un impiegato effettua il login, accede a una pagina HOME PAGE IMPIEGATO che contiene l'elenco dei preventivi gestiti da lui in precedenza e quello dei preventivi non ancora associati a nessun impiegato. Quando l'impiegato seleziona un elemento dall'elenco dei preventivi non ancora associati a nessuno, compare una pagina PREZZA PREVENTIVO che mostra i dati del cliente (username) e del preventivo e una form per inserire il prezzo del preventivo. Quando l'impiegato inserisce il prezzo e invia i dati con il bottone INVIA PREZZO, compare di nuovo la pagina HOME PAGE IMPIEGATO con gli elenchi dei preventivi aggiornati. Il prezzo definito dall'impiegato risulta visibile al cliente quando questi accede all'elenco dei propri preventivi e visualizza i dettagli del preventivo. La pagina PREZZA PREVENTIVO contiene anche un collegamento per tornare alla HOME PAGE IMPIEGATO. L'applicazione consente il logout dell'utente.

2 | Database

2.1. Analisi Testuale

1. Entities
2. Attributes
3. Relationships

Un'applicazione web consente la gestione di richieste di preventivi per prodotti personalizzati. L'applicazione supporta registrazione e login di **clienti** e **impiegati** mediante una pagina pubblica con opportune form. La registrazione controlla l'unicità dello username. **Un preventivo è associato a un prodotto, al cliente che l'ha richiesto e all'impiegato che l'ha gestito.** Il preventivo comprende una o più opzioni per il prodotto a cui è associato, che devono essere tra quelle disponibili per il prodotto. **Un prodotto** ha un **codice**, un'**immagine** e un **nome**. **Un'opzione** ha un **codice**, un **tipo** (“normale”, “in offerta”) e un **nome**. **Un preventivo** ha un **prezzo**, **definito dall'impiegato**. Quando l'utente (cliente o impiegato) accede all'applicazione, appare una LOGIN PAGE, mediante la quale **l'utente** si autentica con **username** e **password**. Quando un cliente fa login, accede a una pagina HOME PAGE CLIENTE che contiene una form per creare un preventivo e l'elenco dei preventivi creati dal cliente. Selezionando uno dei preventivi il cliente ne visualizza i dettagli. Mediante la form di creazione di un preventivo l'utente per prima cosa sceglie il prodotto; scelto il prodotto, la form mostra le opzioni di quel prodotto. L'utente sceglie le opzioni (almeno una) e conferma l'invio del preventivo mediante il bottone INVIA PREVENTIVO. Quando un impiegato effettua il login, accede a una pagina HOME PAGE IMPIEGATO che contiene l'elenco dei preventivi gestiti da lui in precedenza e quello dei preventivi non ancora associati a nessun impiegato. Quando l'impiegato seleziona un elemento dall'elenco dei preventivi non ancora associati a nessuno, compare una pagina PREZZA PREVENTIVO che mostra i dati del cliente (username) e del preventivo e una form per inserire il prezzo del preventivo. Quando l'impiegato inserisce il prezzo e invia i dati con il bottone INVIA PREZZO, compare di nuovo la pagina HOME PAGE IMPIEGATO con gli elenchi dei preventivi aggiornati. Il prezzo definito dall'impiegato risulta

visibile al cliente quando questi accede all'elenco dei propri preventivi e visualizza i dettagli del preventivo. La pagina PREZZA PREVENTIVO contiene anche un collegamento per tornare alla HOME PAGE IMPIEGATO. L'applicazione consente il logout dell'utente

2.2. Database Design

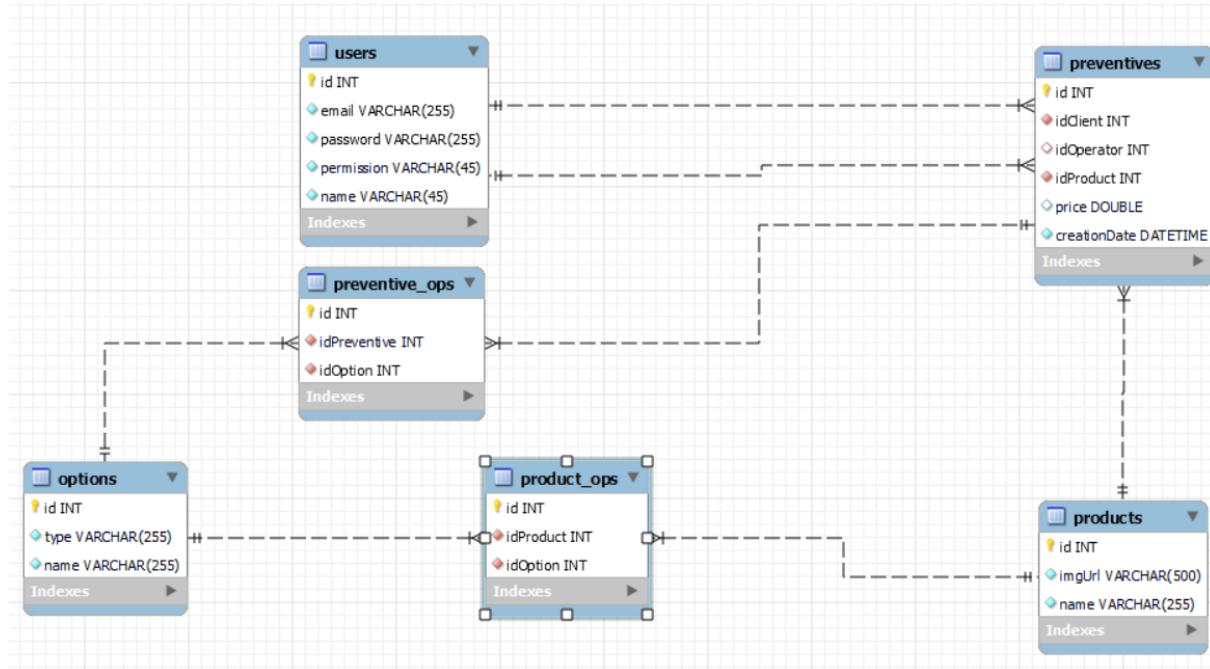


Figure 2.1: Diagramma Entita relazionale

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 | Applicazione

3.1. Analisi Requisiti Applicazione

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

Un'applicazione web consente la gestione di richieste di preventivi per prodotti personalizzati. L'applicazione supporta registrazione e login di clienti e impiegati mediante una pagina pubblica con opportune form. La registrazione controlla l'unicità dello username. Un preventivo è associato a un prodotto, al cliente che l'ha richiesto e all'impiegato che l'ha gestito. Il preventivo comprende una o più opzioni per il prodotto a cui è associato, che devono essere tra quelle disponibili per il prodotto. Un prodotto ha un codice, un'immagine e un nome. Un'opzione ha un codice, un tipo ("normale", "in offerta") e un nome. Un preventivo ha un prezzo, definito dall'impiegato. Quando l'utente (cliente o impiegato) accede all'applicazione, appare una **LOGIN PAGE**, mediante la quale l'utente si autentica con **username e password**. Quando un cliente fa **login**, accede a una pagina **HOME PAGE CLIENTE** che contiene una **form** per **creare un preventivo** e l'**elenco dei preventivi creati dal cliente**. **Selezionando uno dei preventivi** il cliente ne **visualizza i dettagli**. Mediante la form di **creazione di un preventivo** l'utente per prima cosa **sceglie il prodotto**; scelto il prodotto, la form mostra le opzioni di quel prodotto. L'utente sceglie le opzioni (almeno una) e conferma l'invio del preventivo mediante il **bottone INVIA PREVENTIVO**. Quando un impiegato effettua il login, accede a una pagina **HOME PAGE IMPIEGATO** che contiene l'**elenco dei preventivi gestiti da lui in precedenza** e quello dei **preventivi non ancora associati a nessun impiegato**. Quando l'impiegato **seleziona un elemento dall'elenco dei preventivi non ancora associati a nessuno**, compare una pagina **PREZZA PREVENTIVO** che **mostra i dati del cliente (username)** e del preventivo e **una form** per inserire il prezzo del preventivo. Quando l'impiegato inserisce il prezzo e invia i

dati con il bottone INVIA PREZZO, compare di nuovo la pagina HOME PAGE IMPIEGATO con gli elenchi dei preventivi aggiornati. Il prezzo definito dall'impiegato risulta visibile al cliente quando questi accede all'elenco dei propri preventivi e visualizza i dettagli del preventivo. La pagina PREZZA PREVENTIVO contiene anche un collegamento per tornare alla HOME PAGE IMPIEGATO. L'applicazione consente il logout dell'utente

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 (Incluso nelle pagine dopo il login)

5 | Design Applicativo(IFML)

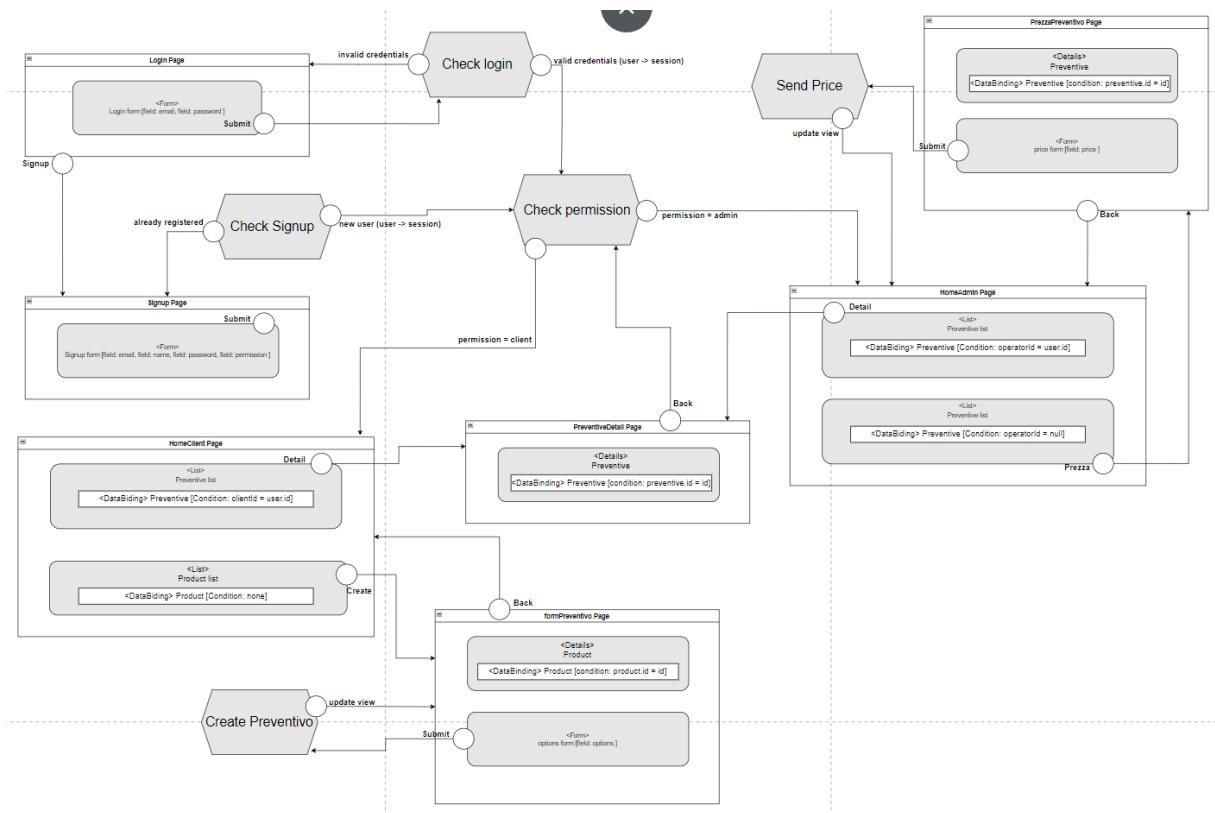


Figure 5.1: Design Applicativo(IFML)

6 | Sequence Diagrams di Interazioni

6.1. CheckLoggedUser

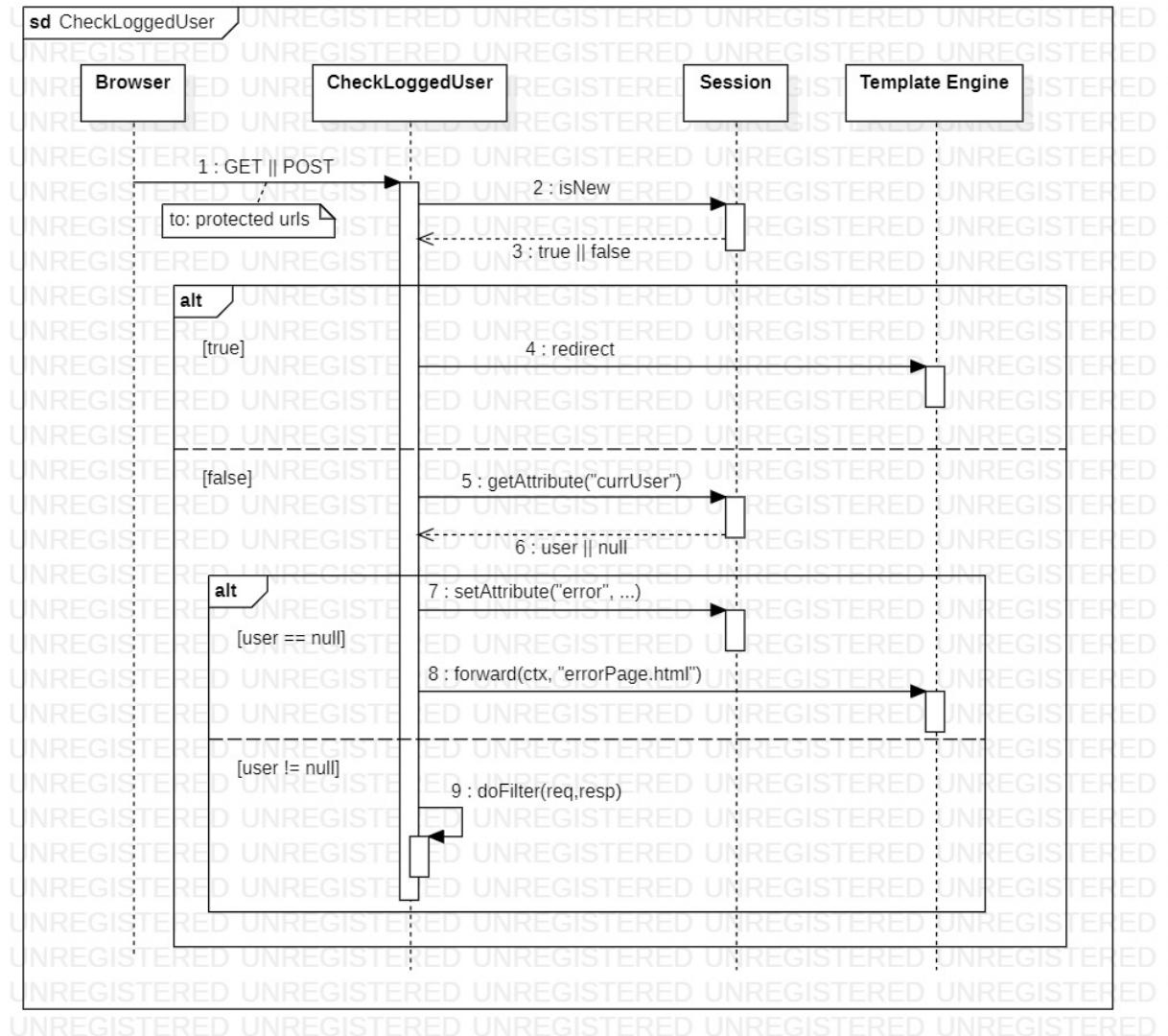


Figure 6.1: CheckLoggedUser

6.2. CheckNotLoggedUser

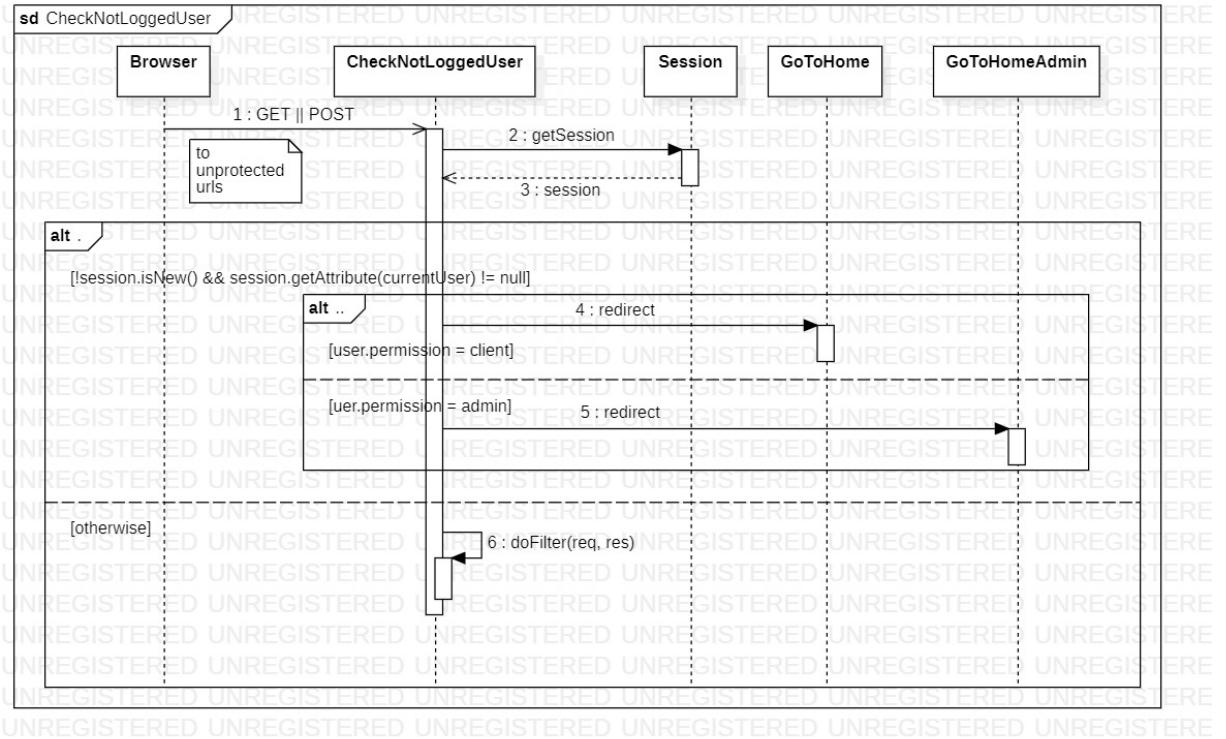


Figure 6.2: CheckNotLoggedUser

6.3. CheckLoggedAdmin

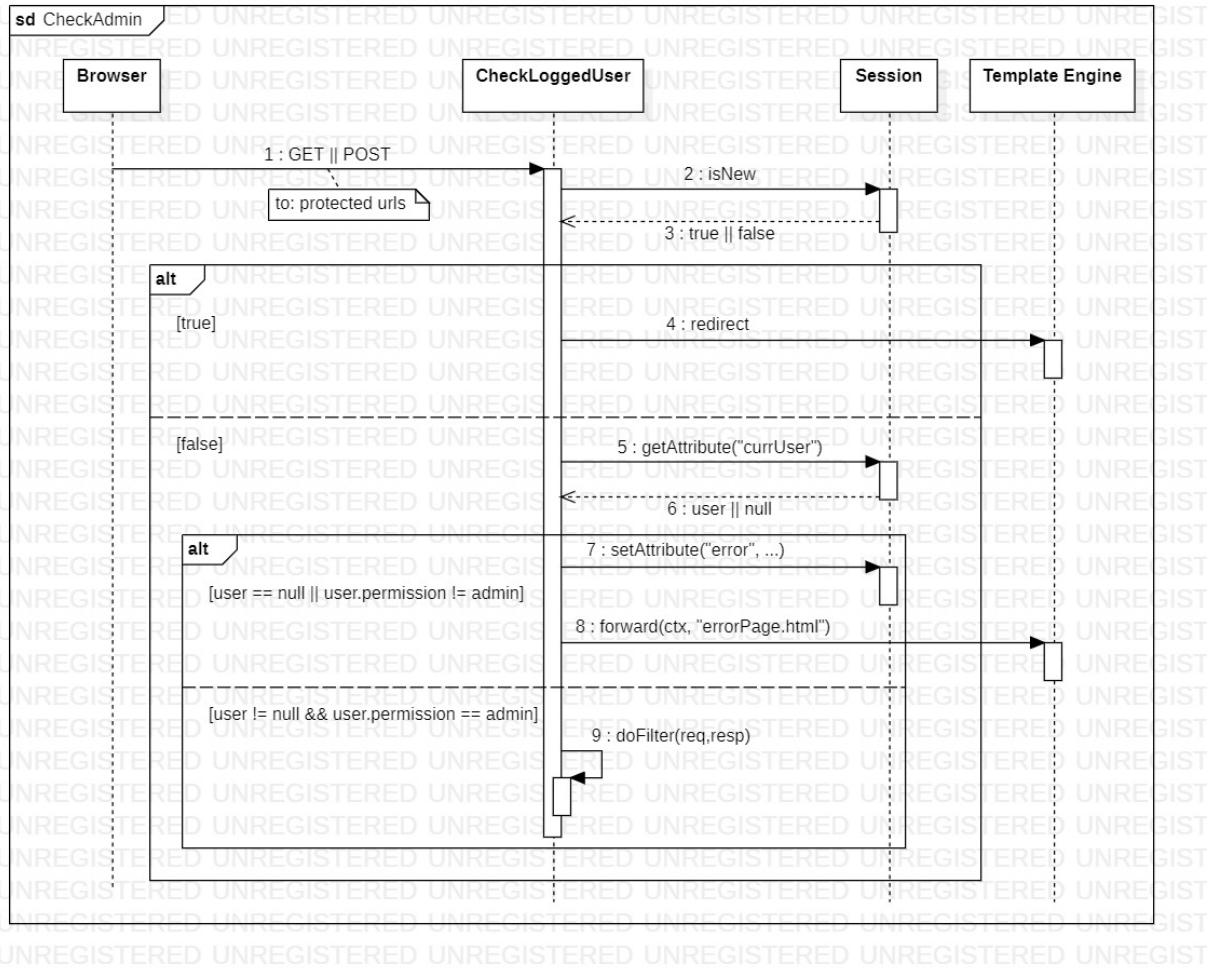


Figure 6.3: CheckLoggedAdmin

6.4. CheckLoggedClient

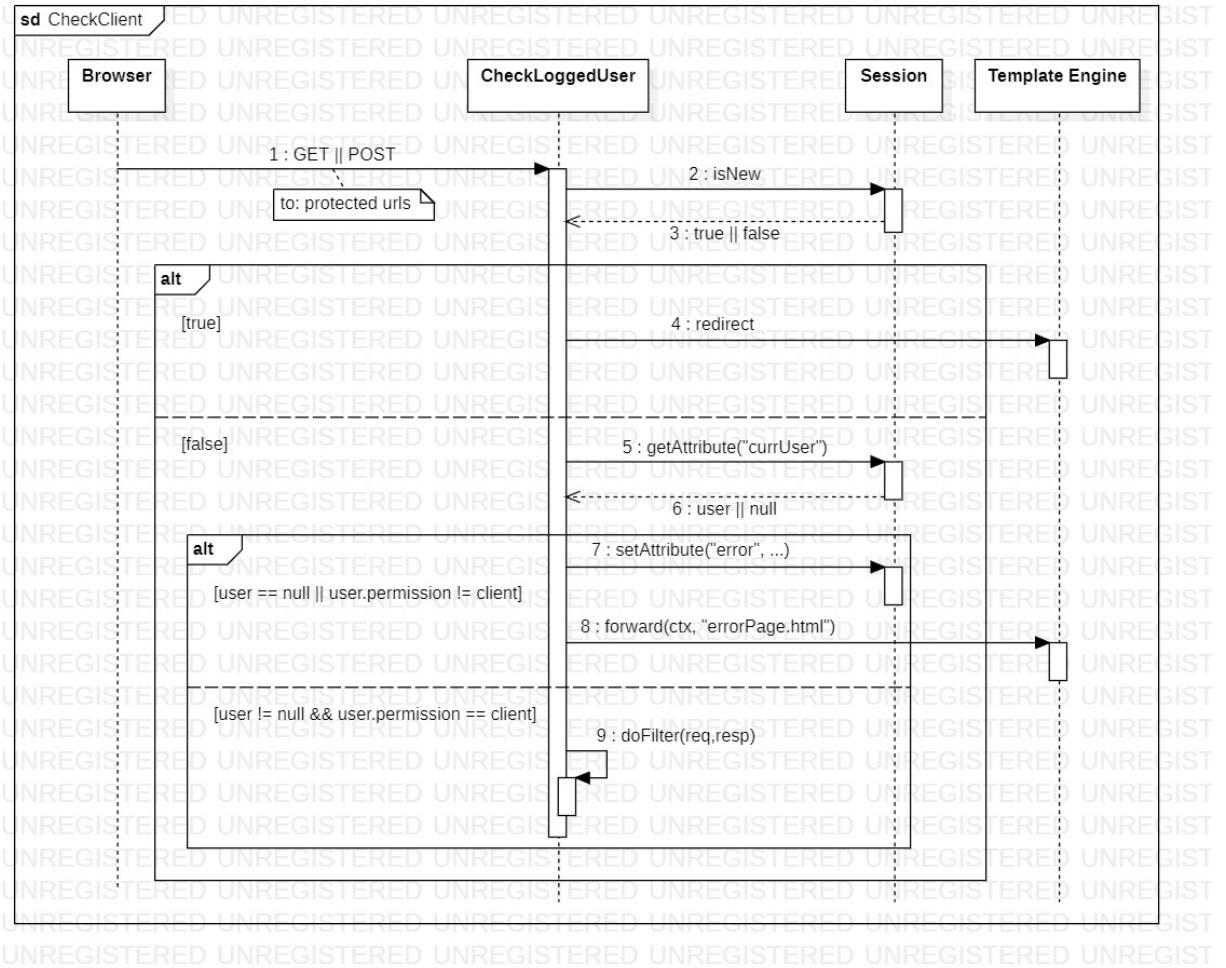


Figure 6.4: CheckLoggedClient

6.5. Login

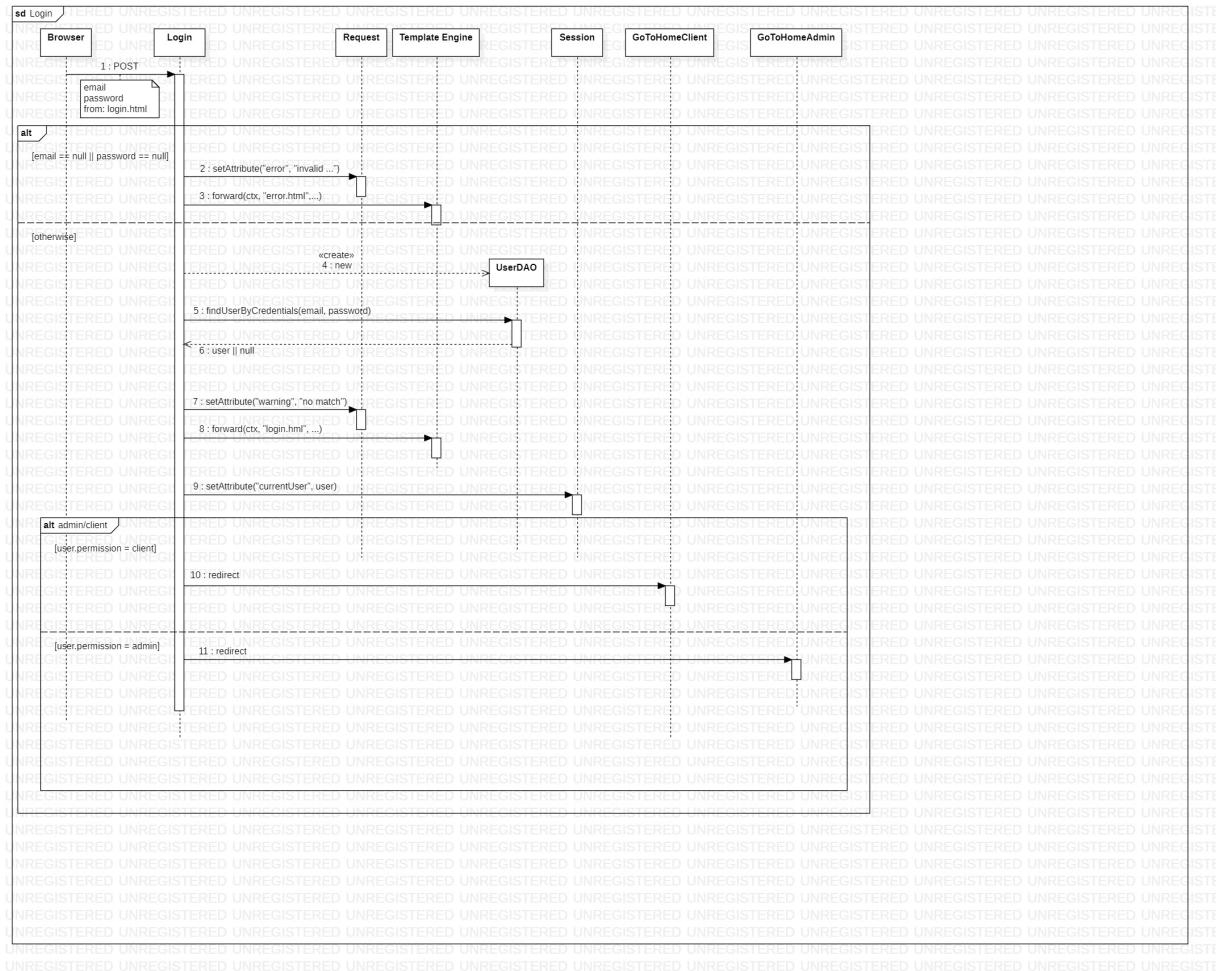


Figure 6.5: Login

6.6. Logout

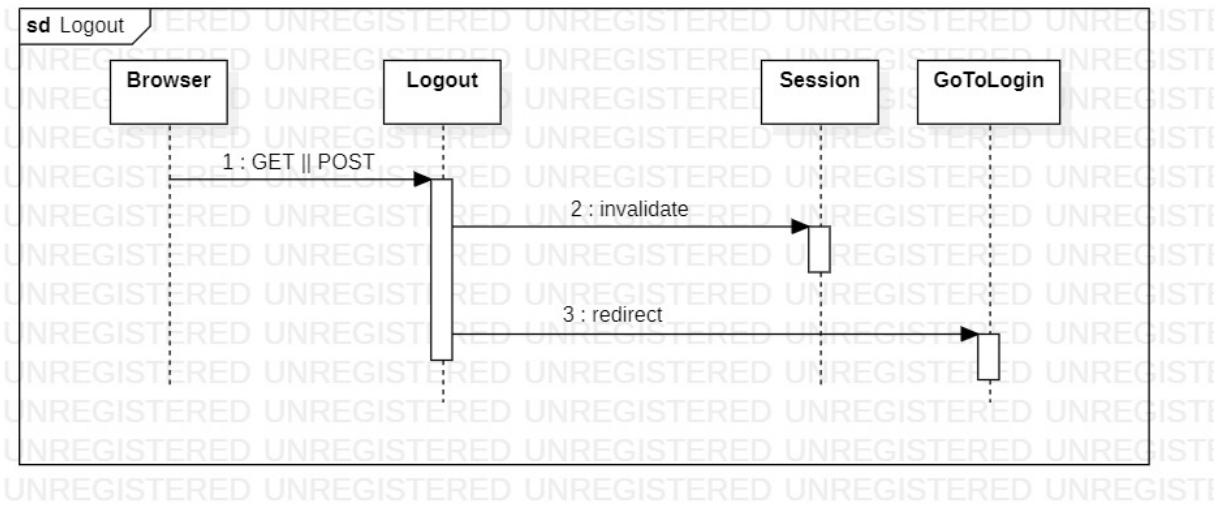


Figure 6.6: Logout

6.7. Signup

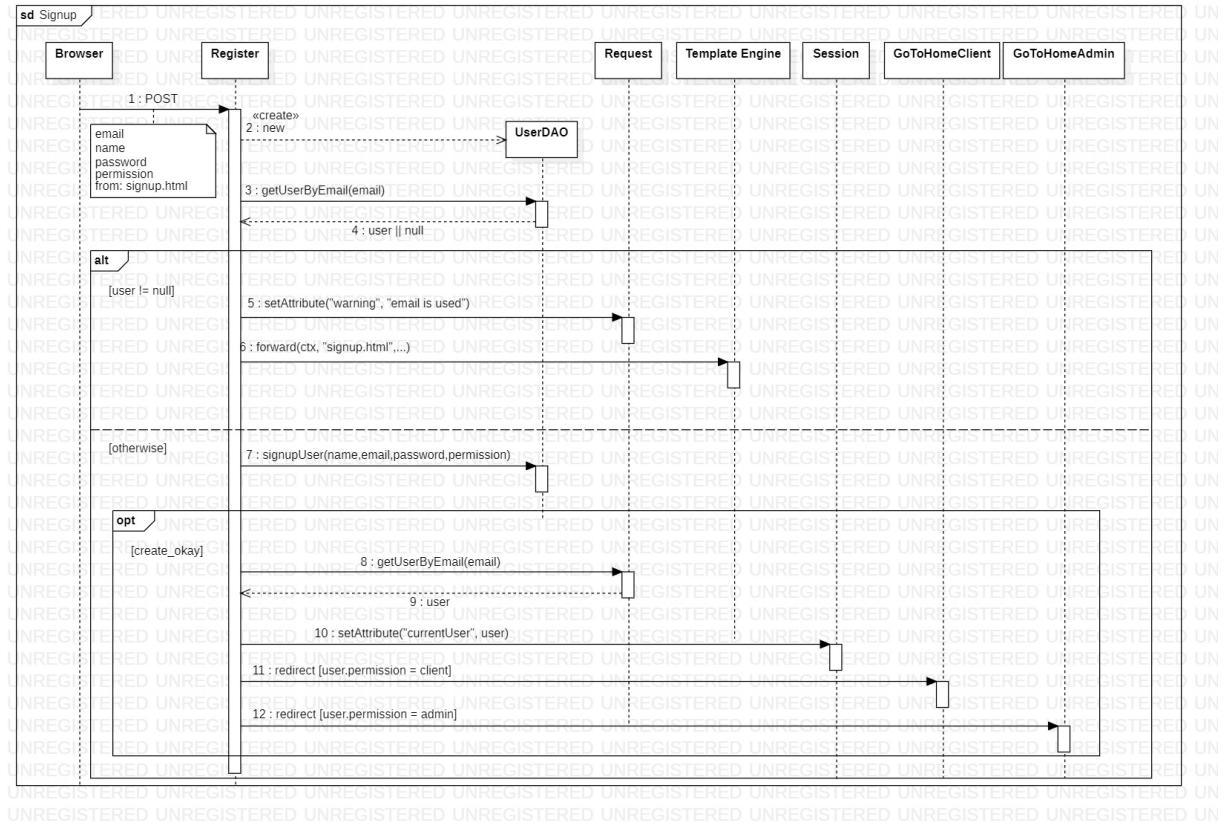


Figure 6.7: Signup

6.8. GoToPreventivoCreate

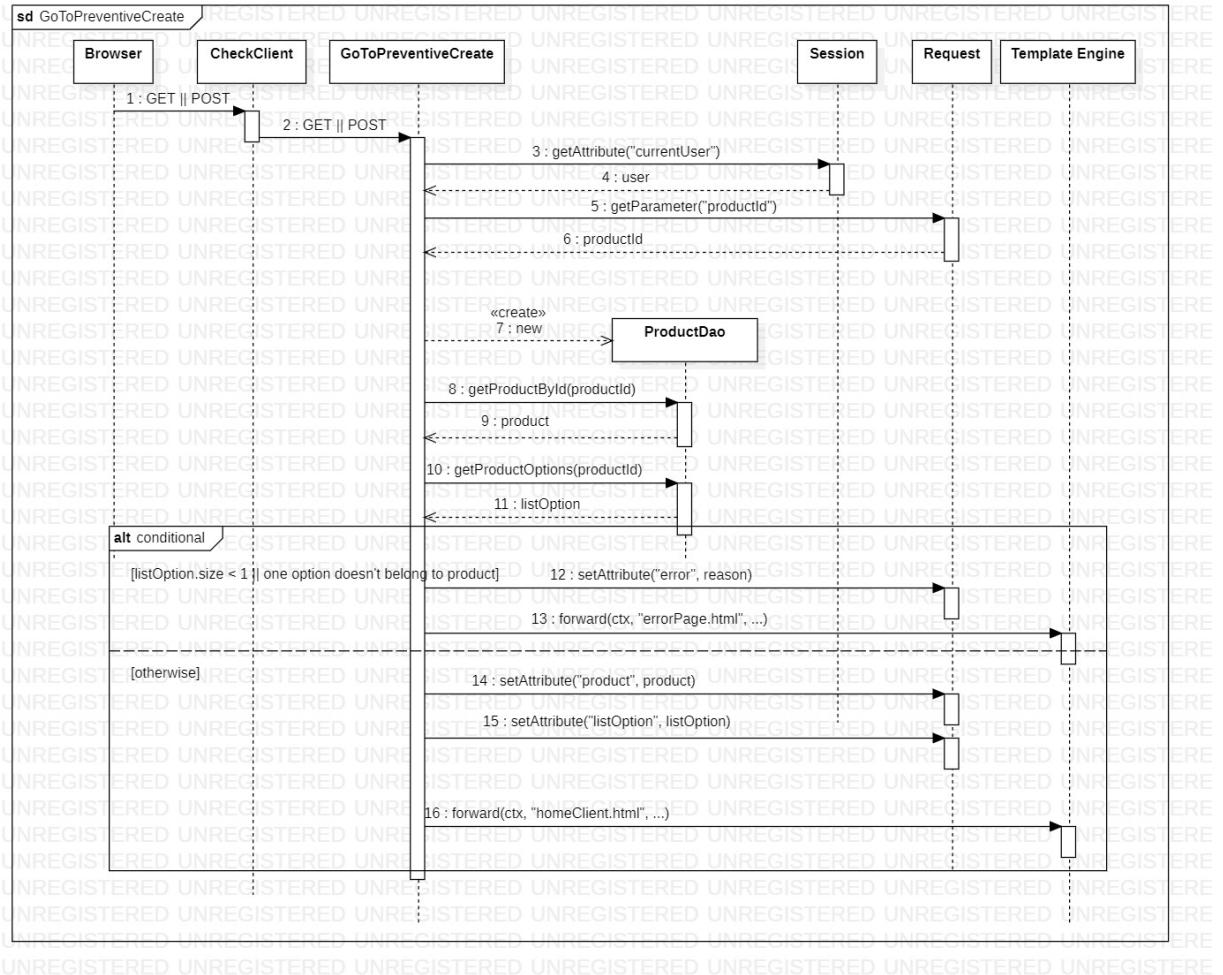


Figure 6.8: GoToPreventivoCreate

6.9. GoToHomeClient

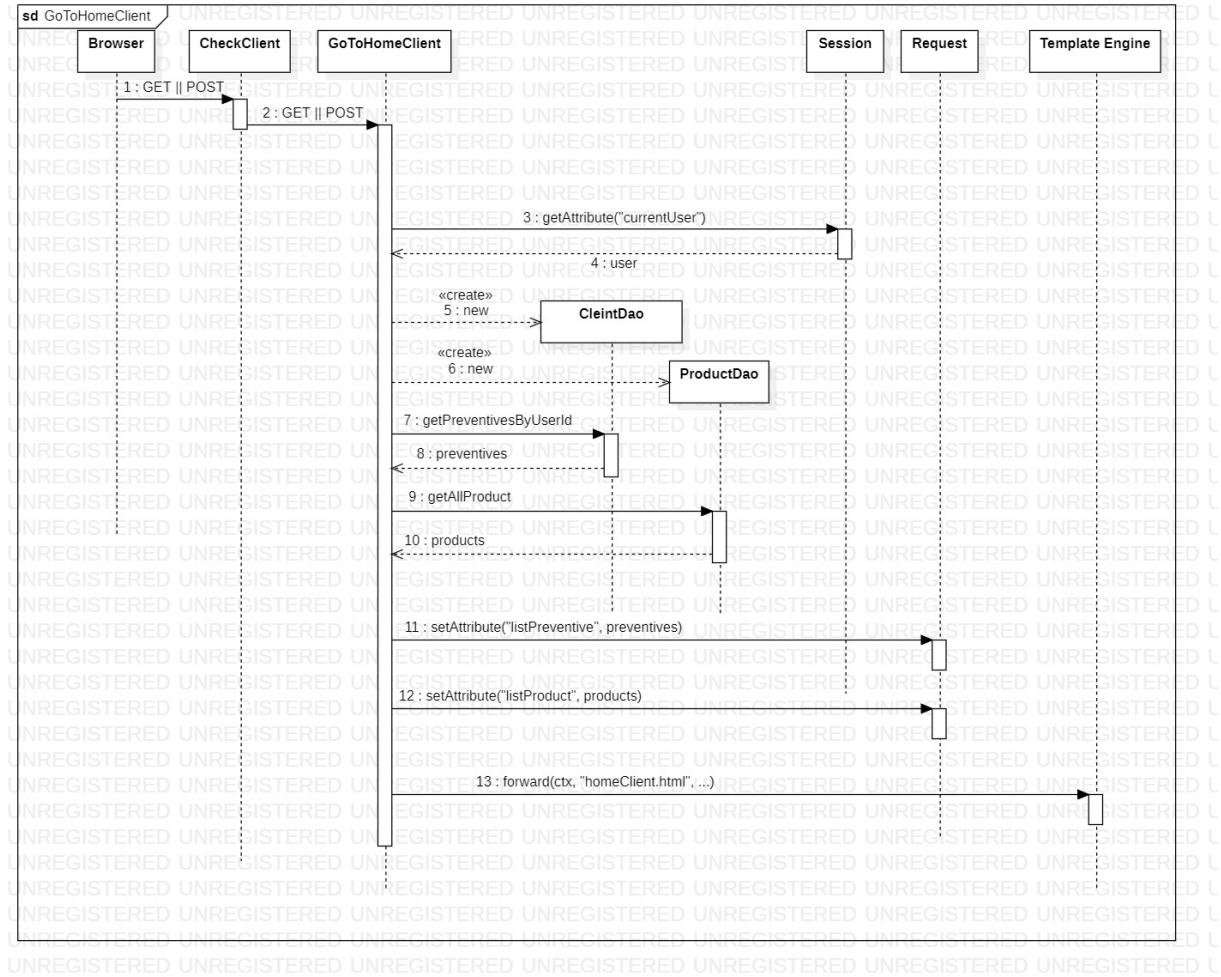
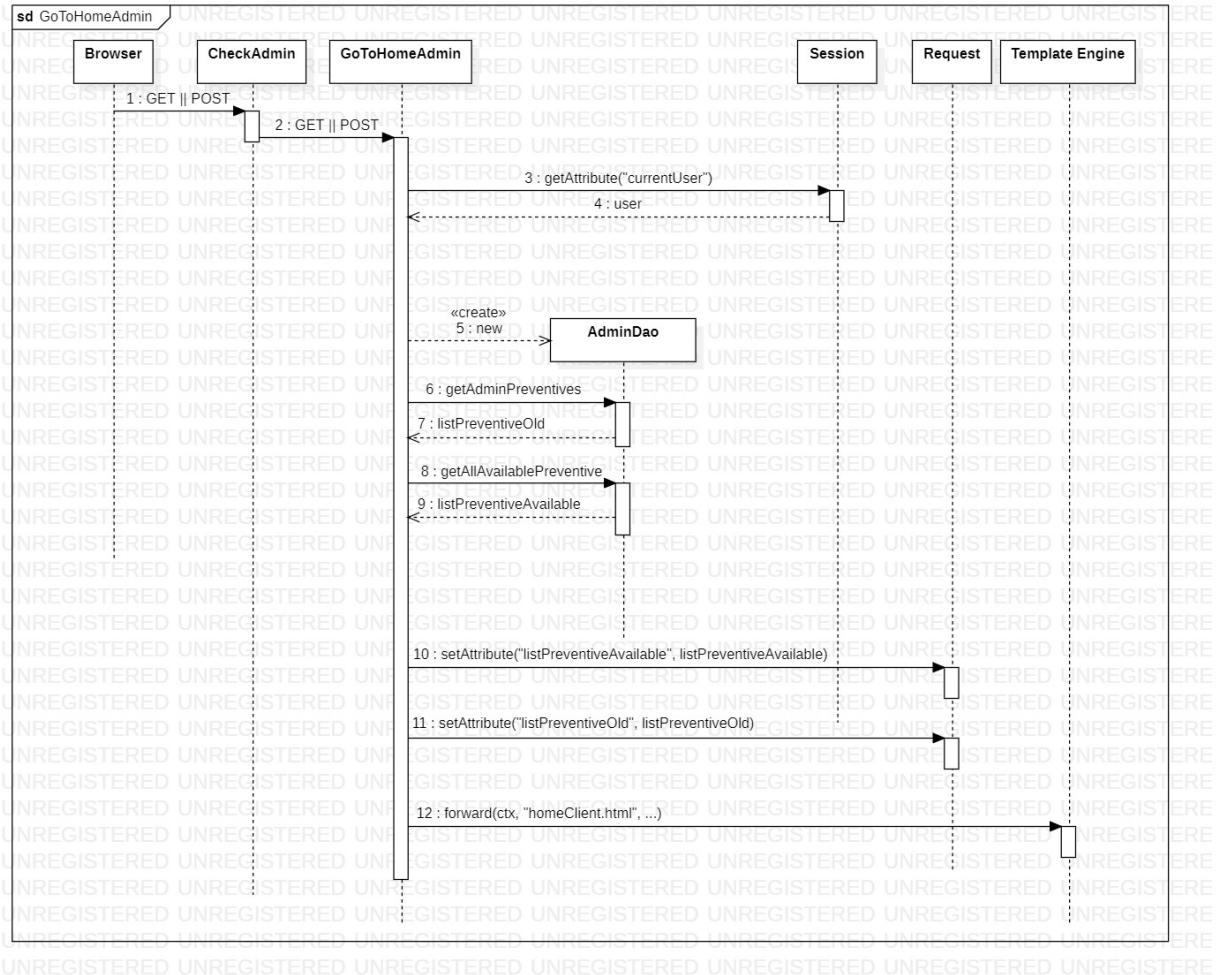


Figure 6.9: GoToHomeClient

6.10. GoToHomeAdmin



6.11. GoToPreventivoDetail

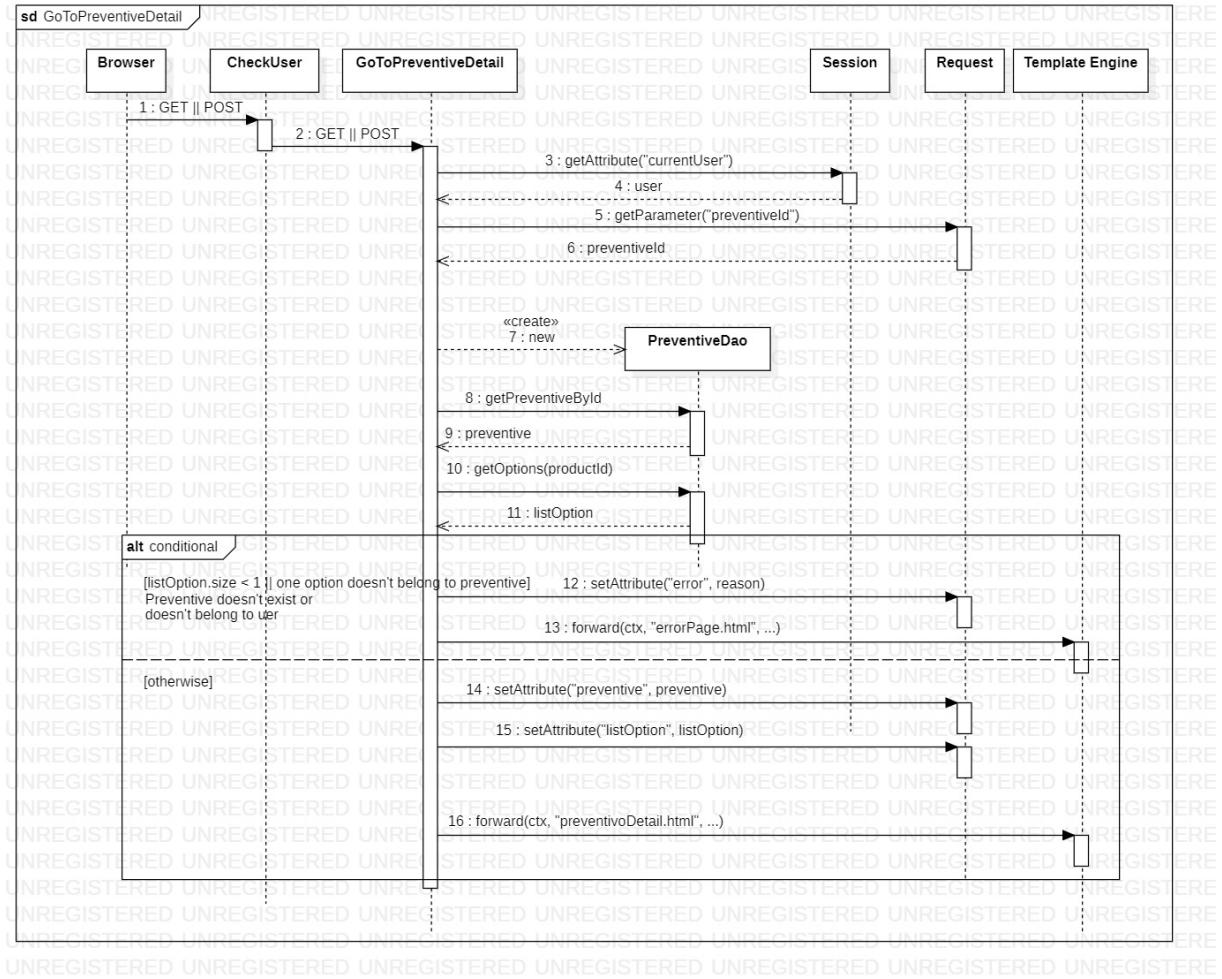


Figure 6.10: GoToPreventivoDetail

6.12. GoToPreventivoPrezza

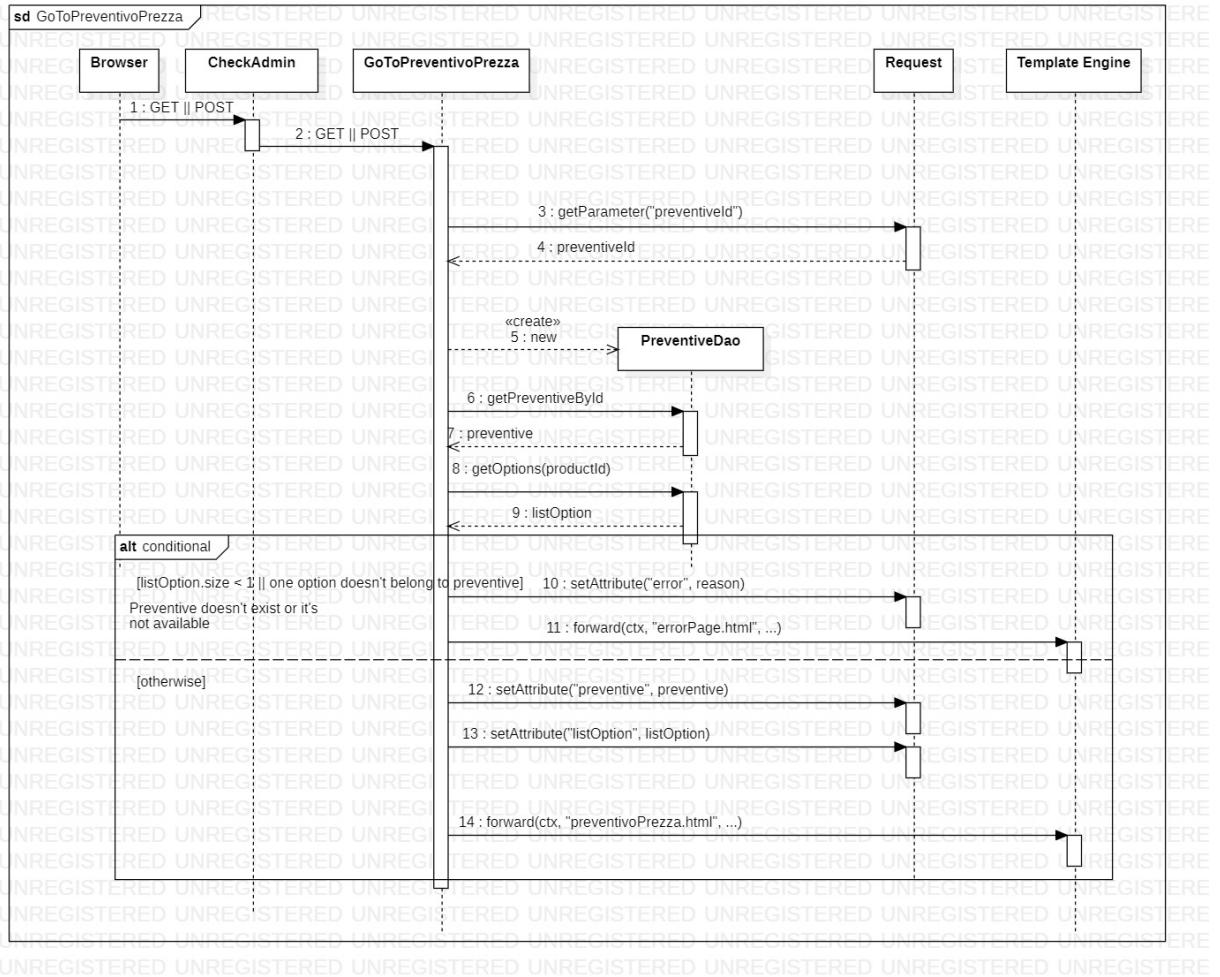


Figure 6.11: GoToPreventivoPrezza

6.13. GoToSignup

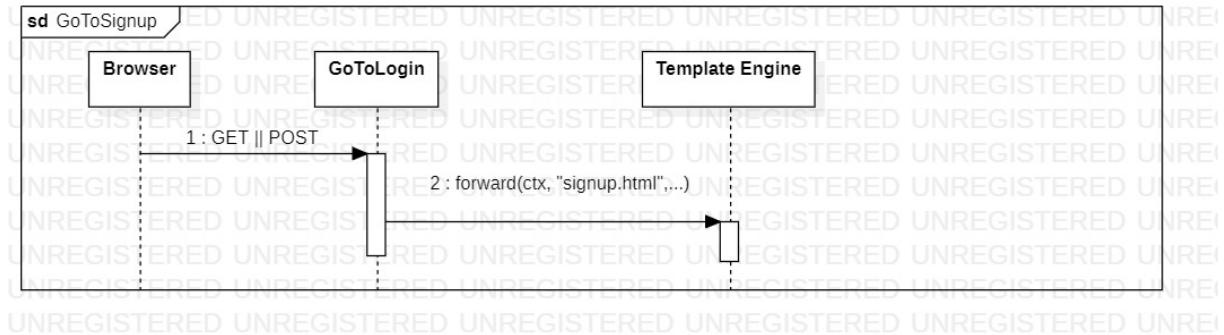


Figure 6.12: GoToSignup

6.14. SendPreventivo

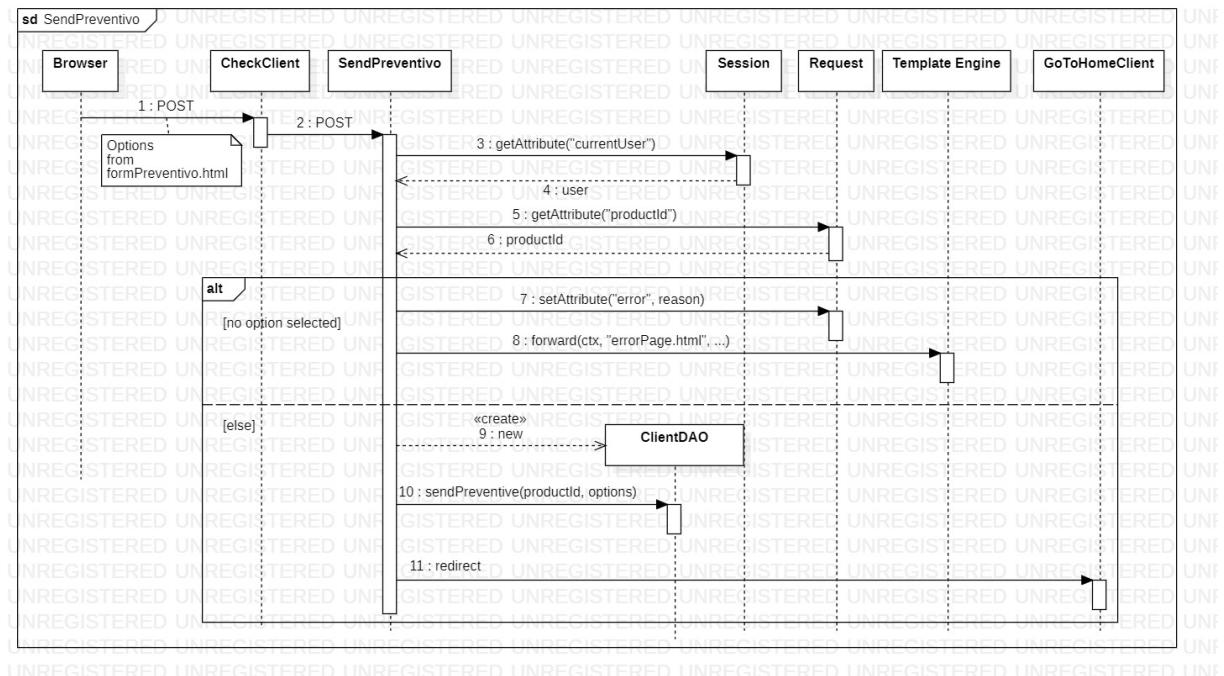


Figure 6.13: SendPreventivo

6.15. SendPrezzo

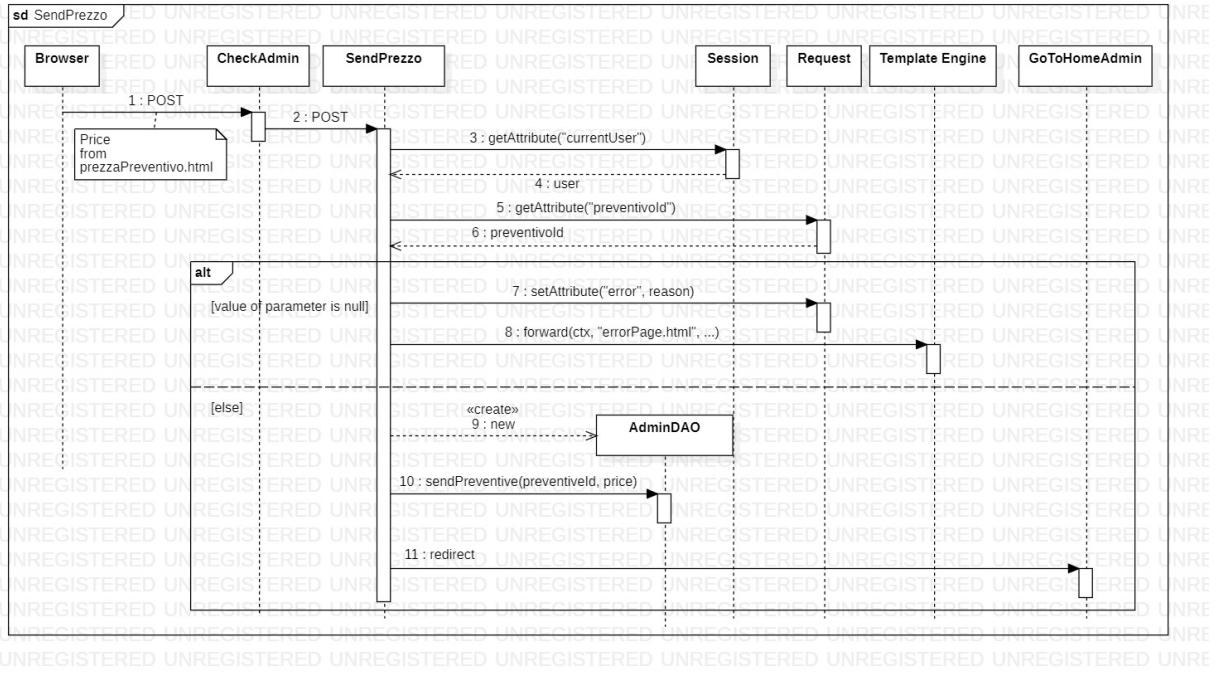


Figure 6.14: SendPrezzo