

TIW – progetto #2

«Gestione preventivi»

Francesco Scroccarello

Matricola: 936982, CP: 10723028

Prof. Piero Fraternali

1 Analisi delle specifiche

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**.

Database analysis:

Entities

Attributes

Relationships

Application analysis:

Pages

View components

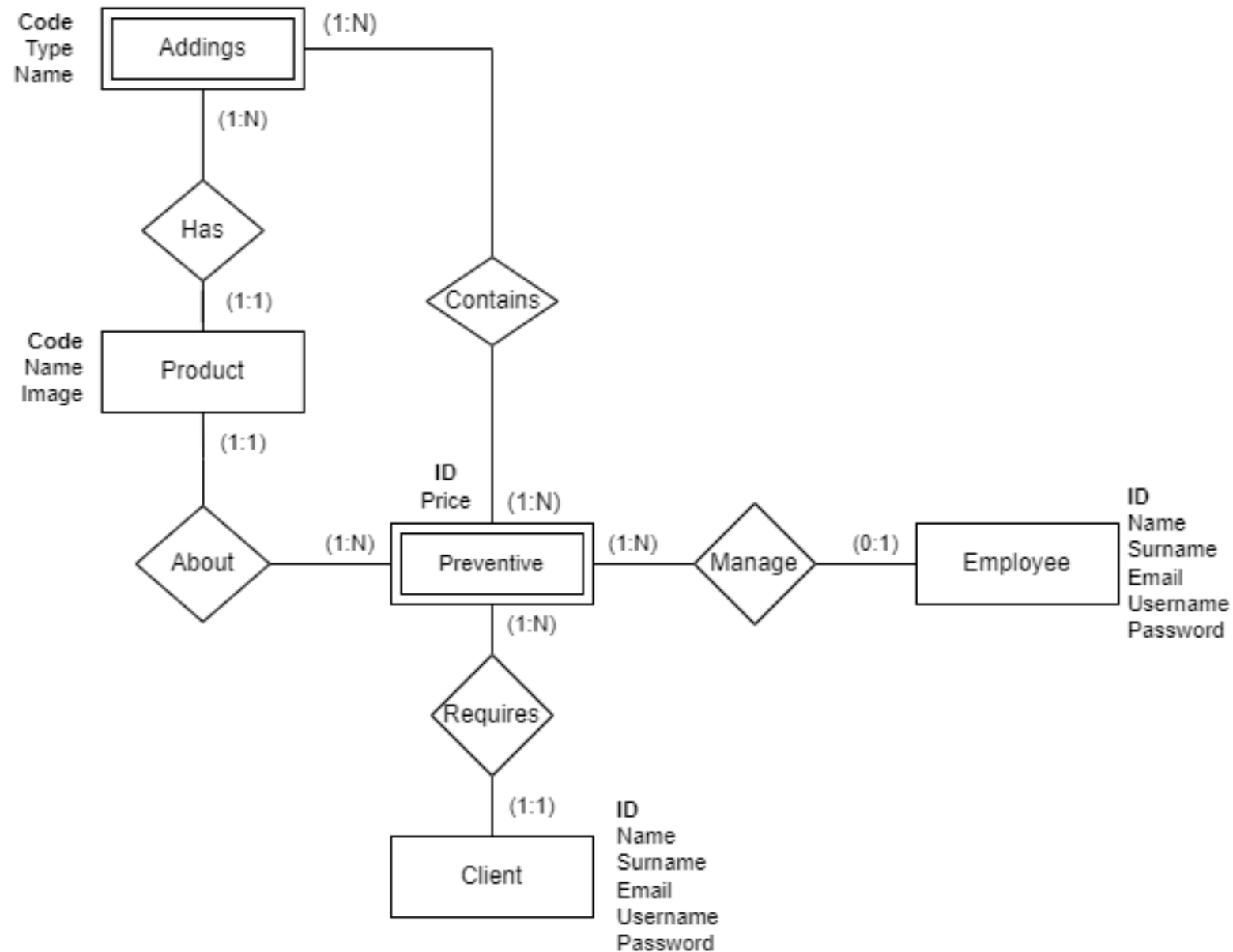
Events

Actions

1.1 Completamento delle specifiche

- La pagina di default contiene il link ad una pagina aggiuntiva per la registrazione.
- Nel caso in cui un preventivo non sia ancora stato completato, il cliente visualizzerà un messaggio provvisorio.
- Il logout dell'utente riporta alla pagina di login.

1.2.1 Database design: Entità-Relazione



1.2.2 Database design: Schema Logico

```
CREATE TABLE `client` (  
  `ID` int NOT NULL AUTO_INCREMENT,  
  `Name` varchar(45) NOT NULL,  
  `Surname` varchar(45) NOT NULL,  
  `Email` varchar(45) NOT NULL,  
  `Username` varchar(45) NOT NULL,  
  `Password` varchar(45) NOT NULL,  
  PRIMARY KEY (`ID`),  
  
  UNIQUE KEY `Username_UNIQUE`  
  (`Username`)  
)
```

```
CREATE TABLE `employee` (  
  `ID` int NOT NULL AUTO_INCREMENT,  
  `Name` varchar(45) NOT NULL,  
  `Surname` varchar(45) NOT NULL,  
  `Email` varchar(45) NOT NULL,  
  `Username` varchar(45) NOT NULL,  
  `Password` varchar(45) NOT NULL,  
  PRIMARY KEY (`ID`),  
  
  UNIQUE KEY `Username_UNIQUE`  
  (`Username`)  
)
```

```
CREATE TABLE `product` (  
  `Code` int NOT NULL AUTO_INCREMENT,  
  `Name` varchar(45) NOT NULL, `Image`  
  longtext NOT NULL, PRIMARY KEY  
  (`Code`)  
)
```

```
CREATE TABLE `preventive` (  
  `ID` int NOT NULL AUTO_INCREMENT,  
  `ClientID` int NOT NULL,  
  `EmployeeID` int DEFAULT NULL,  
  `ProductCode` int NOT NULL,  
  `ProductName` varchar(45) NOT NULL,  
  `Price` int DEFAULT NULL,  
  PRIMARY KEY  
  (`ID`,`ClientID`,`ProductCode`), KEY  
  `ClientID_idx` (`ClientID`), KEY  
  `EmployeeID_idx` (`EmployeeID`), KEY  
  `ProductCode_idx` (`ProductCode`),  
  CONSTRAINT `ClientID` FOREIGN KEY  
  (`ClientID`) REFERENCES `client` (`ID`) ON  
  DELETE CASCADE ON UPDATE CASCADE,  
  CONSTRAINT `EmployeeID` FOREIGN KEY  
  (`EmployeeID`) REFERENCES `employee`  
  (`ID`) ON DELETE SET NULL ON UPDATE  
  CASCADE, CONSTRAINT `ProductID`  
  FOREIGN KEY (`ProductCode`) REFERENCES  
  `product` (`Code`) ON DELETE CASCADE ON  
  UPDATE CASCADE)
```

```
CREATE TABLE `addings` (  
  `Code` int NOT NULL AUTO_INCREMENT,  
  `ProductCode` int NOT NULL,  
  `Type` varchar(45) NOT NULL,  
  `Name` varchar(45) NOT NULL,  
  PRIMARY KEY (`Code`), KEY  
  `ProductCode_idx` (`ProductCode`),  
  CONSTRAINT `ProductCode` FOREIGN KEY  
  (`ProductCode`) REFERENCES `product`  
  (`Code`) ON DELETE CASCADE ON UPDATE  
  CASCADE  
)
```

```
CREATE TABLE `optionselected` (  
  `OptionCode` int NOT NULL,  
  `PreventiveID` int NOT NULL,  
  PRIMARY KEY  
  (`OptionCode`,`PreventiveID`), KEY  
  `PrevCode_idx` (`PreventiveID`),  
  CONSTRAINT `OptionCode` FOREIGN KEY  
  (`OptionCode`) REFERENCES `addings`  
  (`Code`) ON DELETE CASCADE ON UPDATE  
  CASCADE, CONSTRAINT `PrevCode`  
  FOREIGN KEY (`PreventiveID`)  
  REFERENCES `preventive` (`ID`)  
)
```

1.3.1 Componenti (versione pure HTML)

Model Objects (beans):

- User
- Client
- Employee
- Product
- Preventive
- Option

Data Access Objects:

- ConnectionHandler
- UserDAO
- ClientDAO
- EmployeeDAO
- ProductDAO
- OptionDAO

Controllers (servlets):

- LoginManager
- Logout
- RegistrationHandler
- HomeClientHadler
- HomeEmployeeHandler
- GetPreventiveDetails
- CreatePreventve
- CompletePreventive
- PricePreventive
- BackToHome

Views (template pages):

- login
- register
- homeClient
- homeEmployee
- pricePrevntive

1.3.1 Componenti (versione RIA)

Model Objects (beans):

- User
- Client
- Employee
- Product
- Preventive
- Option

Data Access Objects:

- ConnectionHandler
- UserDAO
- ClientDAO
- EmployeeDAO
- ProductDAO
- OptionDAO

Controllers (servlets):

- CheckLogin
- Logout
- RegistrationManager
- CreatePreventive
- CompletePreventive
- BackToHome
- GetOptions
- GetPreventives
- GetProducts
- GetUnhandled

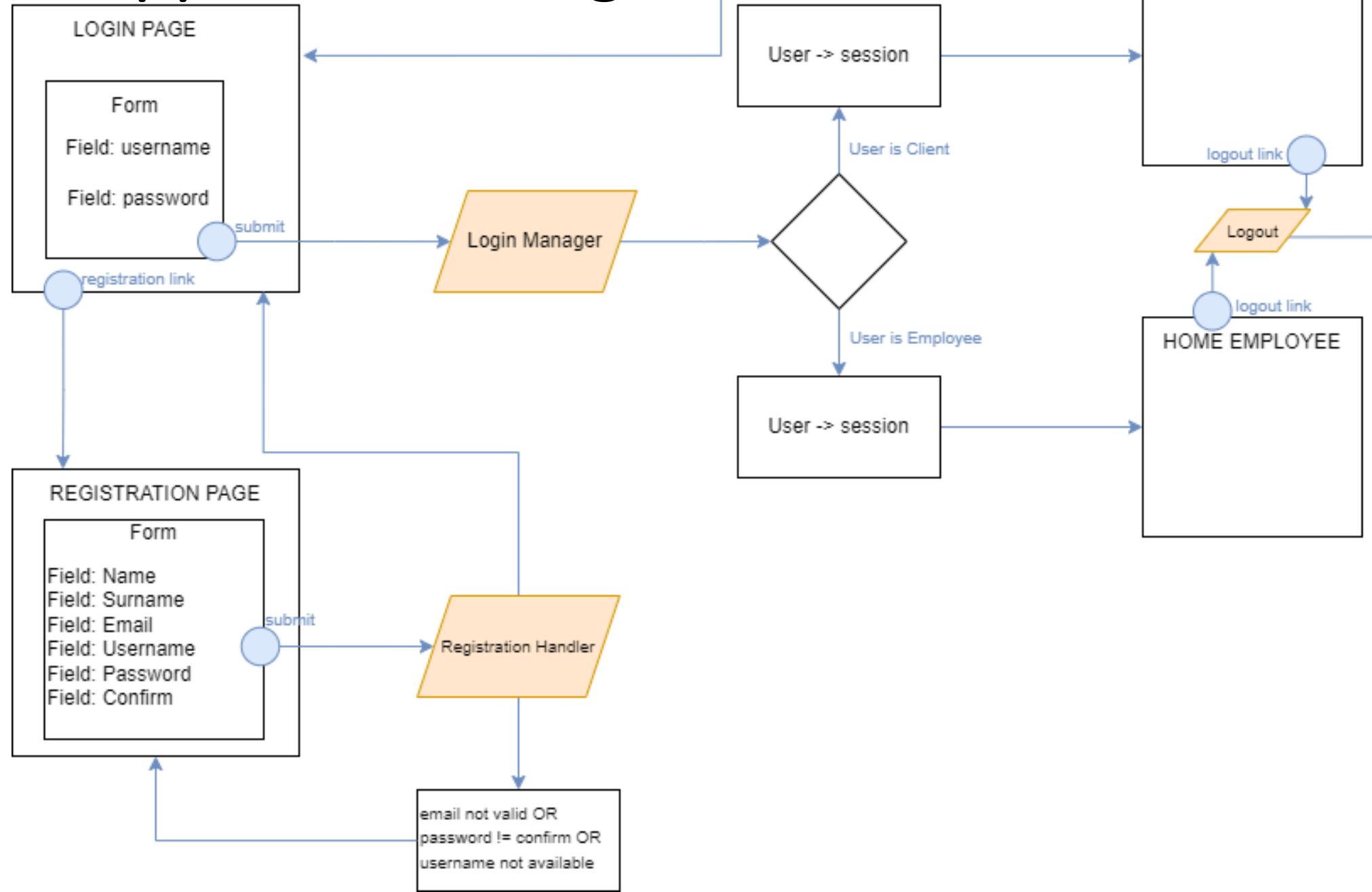
Views (pages):

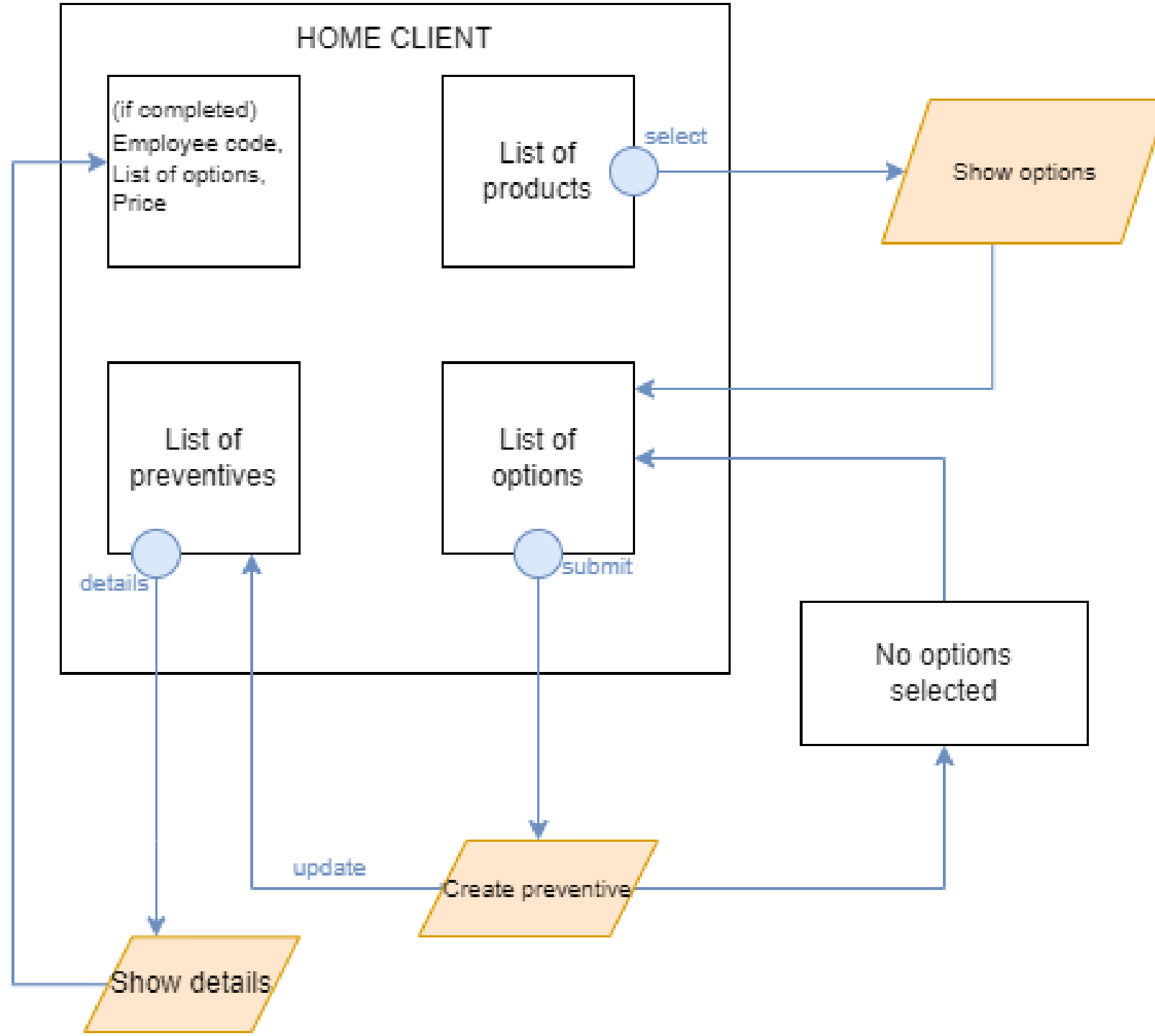
- Login
- Registration
- HomeClient
- HomeEmployee

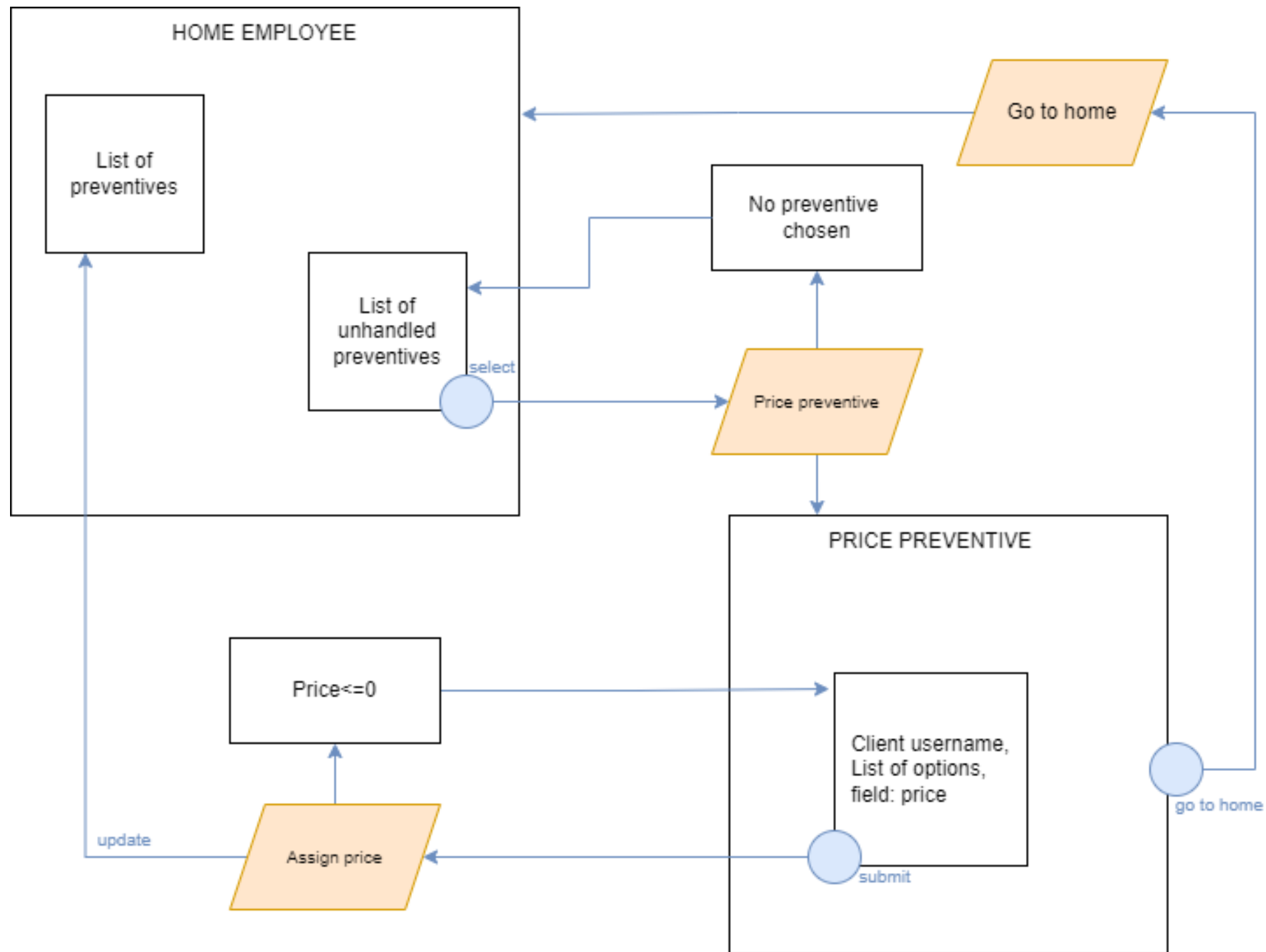
Client side scripts:

- LoginHandler
- RegistrationHandler
- HomeClientHandler
- HomeEmployeeHandler

1.4 Application design







Si noti che poiché una delle richieste per la versione RIA sia di realizzare l'intera applicazione con una pagina singola per ciascuno dei due ruoli, il design dell'applicazione per quanto riguarda la pagina impiegato è stato modificato come segue:

