

# Gestione Preventivi

Esercizio d'esame di Tecnologie Informatiche per il Web

---

Riccardo Paltrinieri

Giugno 2020

10626923

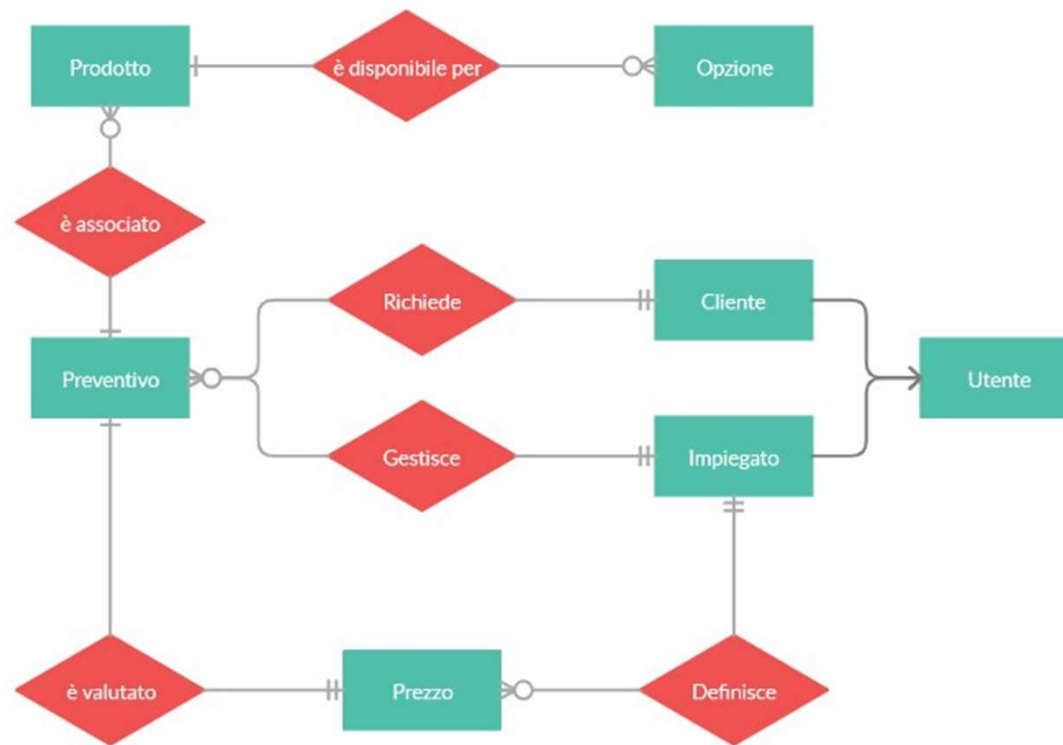
# Analisi dei dati

Un'applicazione web consente la gestione di richieste di preventivi per prodotti personalizzati.

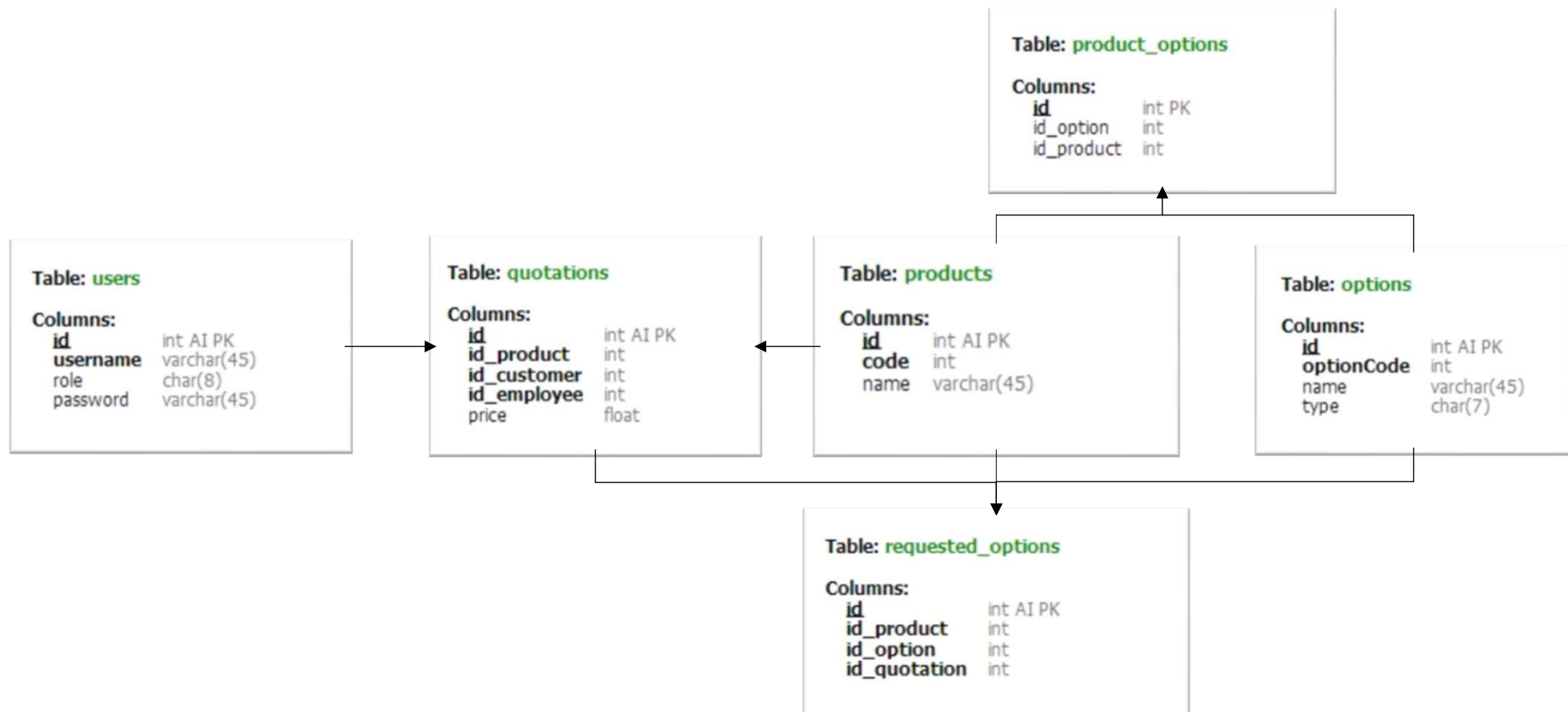
Un preventivo è associato a un prodotto, al cliente che l'ha richiesto e all'impiegato che l'ha gestito. Il preventivo comprende una o più opzioni per il prodotto a cui è associato, che devono essere tra quelle disponibili per il prodotto. Un prodotto ha un codice, un'immagine e un nome. Un'opzione ha 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 si autentica con username e password.

Entities, attributes, relationships

# Database Design



# Database Design



# Analisi dei requisiti

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. Mediante la form 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 fa 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.

**Pages, components, events, actions**

# Html Components



## **Model Objects (Beans)**

- User
- Product
- Quotation
- Option



## **Controllers (Servlets)**

- Login
- Signup
- Logout
- HomeCustomer
- HomeEmployee
- PriceQuote
- GetImage



## **Data Access Objects (DAO)**

- UserDao
- ProductDAO
- QuotationDAO
- OptionDAO



## **Filters**

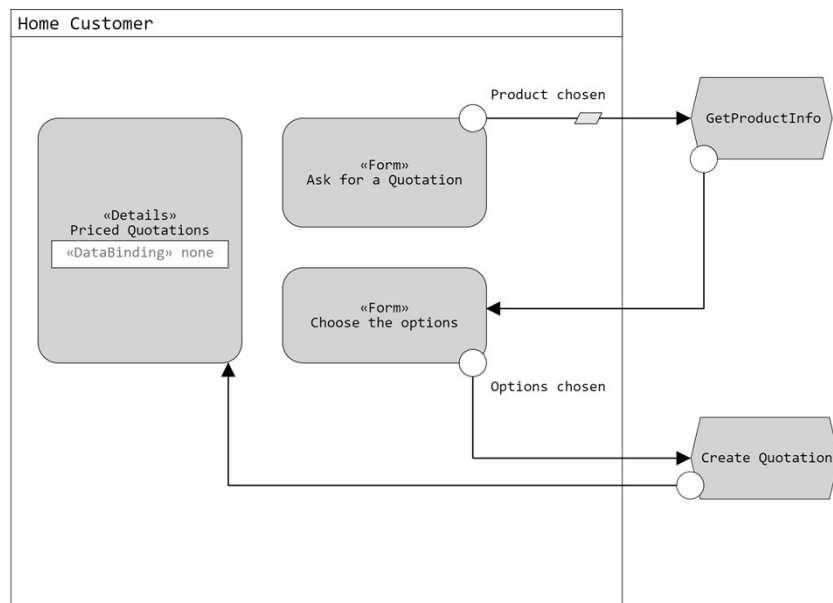
- UserChecker
- EmployeeChecker



## **View (Templates)**

- index
- HomeCustomer
- HomeEmployee
- PriceQuote
- SignUp

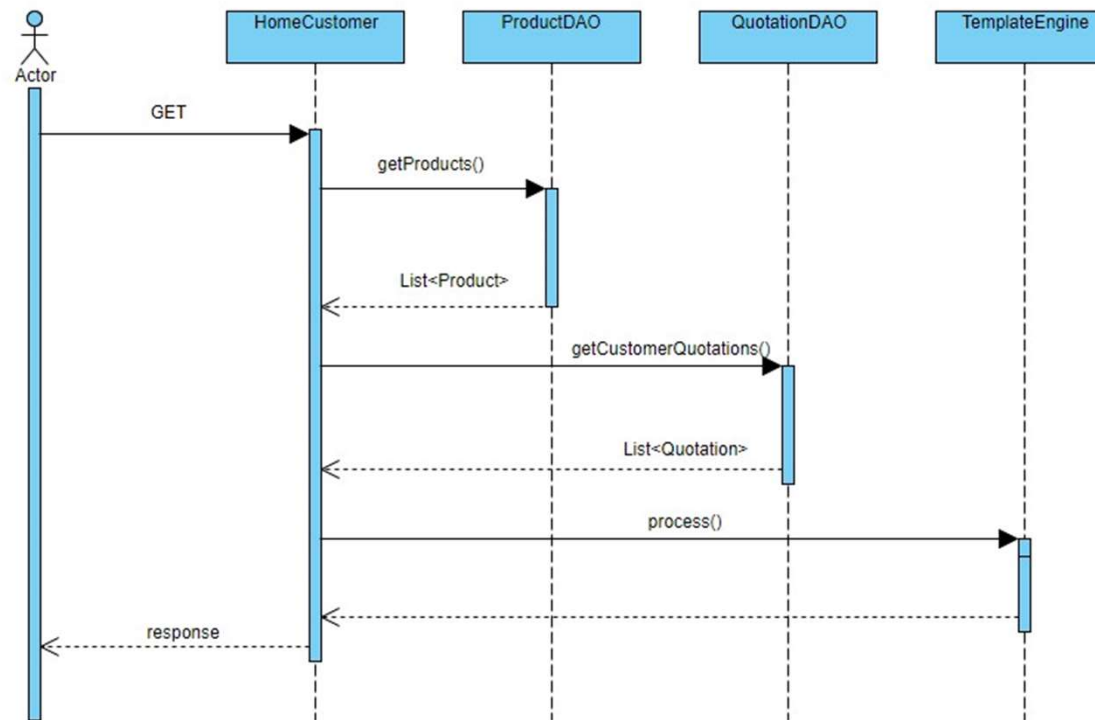
# Application Design



Quando un cliente fa login, accede a una pagina **Home Customer** che contiene una form per creare un preventivo e l'elenco dei preventivi creati dal cliente.

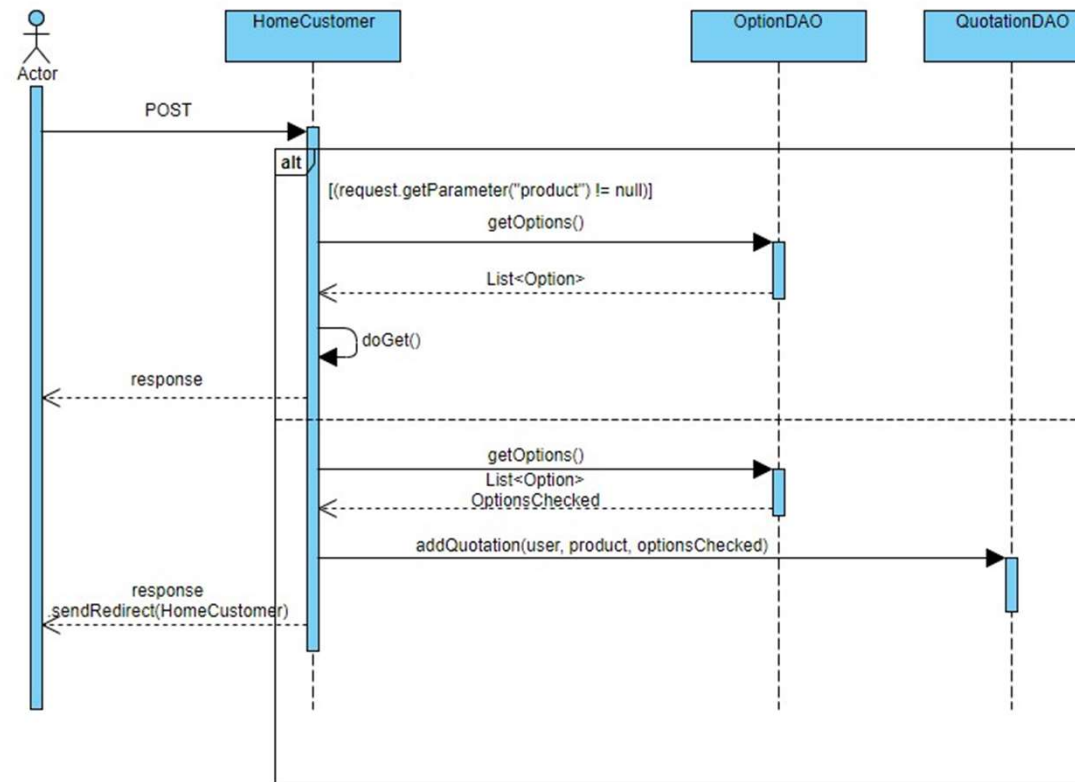
Mediante la form l'utente per prima cosa sceglie il prodotto; scelto il prodotto, la form mostra le opzioni di quel prodotto. L'utente sceglie le opzioni (almeno una) e conferma l'invio del preventivo.

# Evento: Apertura home page cliente

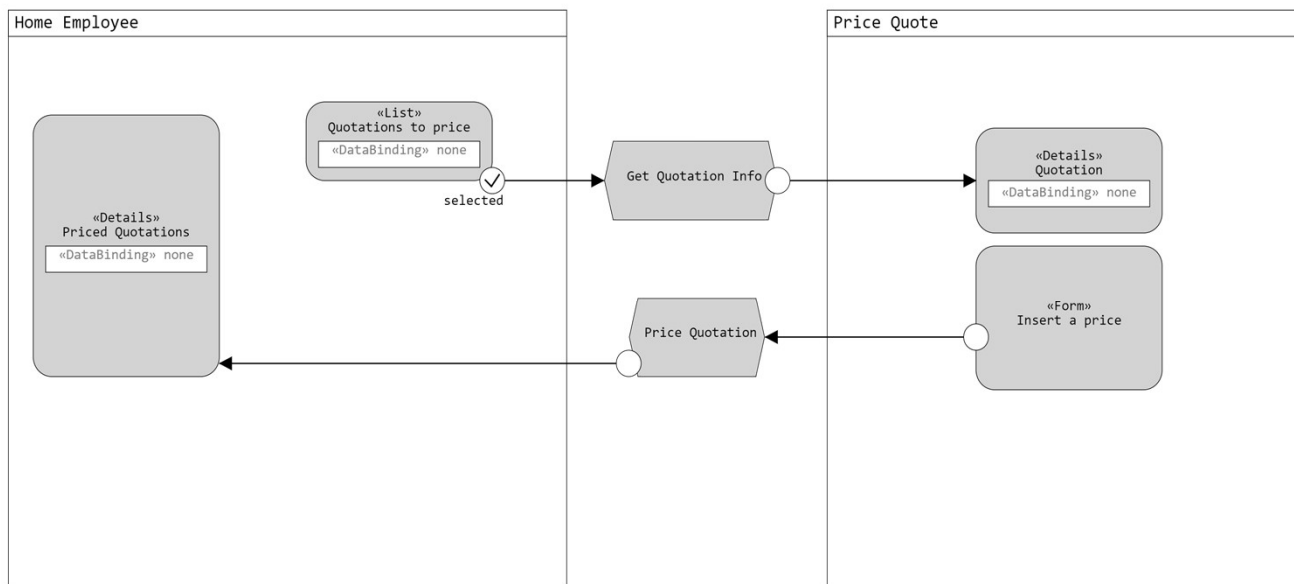




# Evento: Richiesta di un preventivo



# Application Design

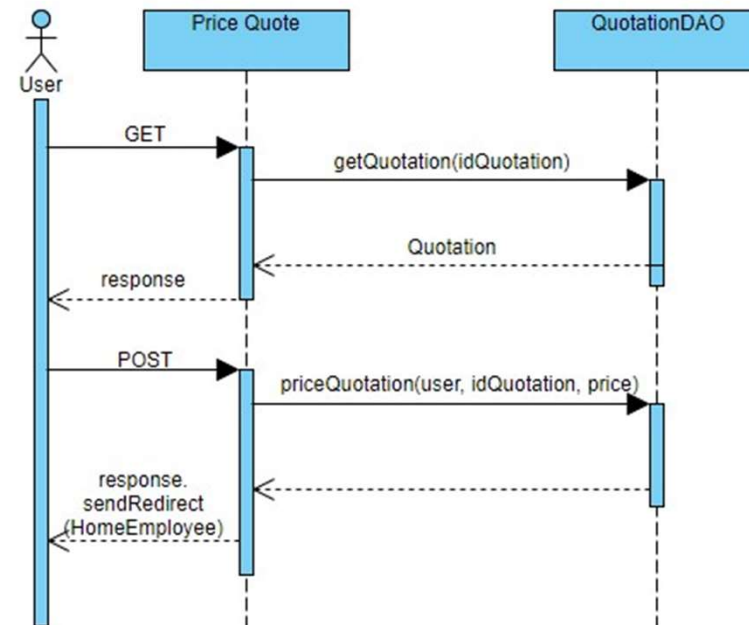
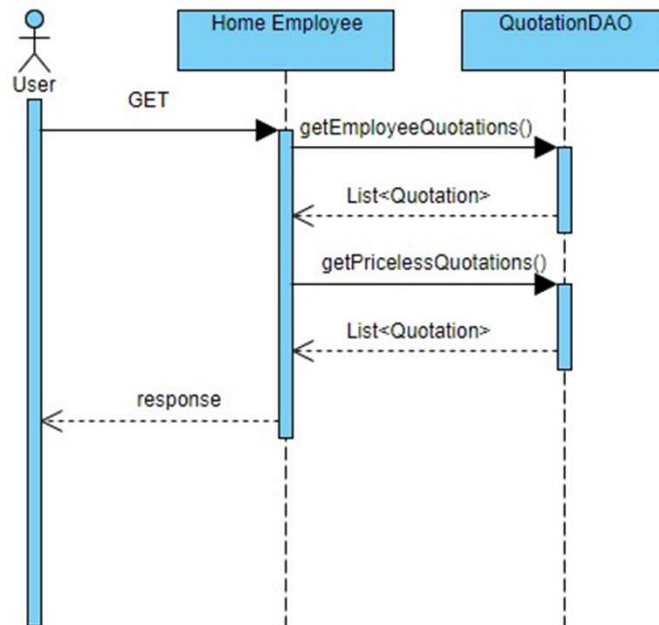


Quando un impiegato fa login, accede a una pagina **Home Employee** che contiene l'elenco dei preventivi gestiti da lui in precedenza e quello dei preventivi non ancora associati a nessun impiegato.

Quando l'impiegato seleziona un elemento dall'elenco dei preventivi non ancora associati a nessuno, compare una pagina **Price Quote** 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, compare di nuovo la pagina Home Employee con gli elenchi dei preventivi aggiornati.

# Evento: Apertura Home Page Impiegato e inserimento prezzo di un preventivo



# Completamento delle specifiche

## Versione con JavaScript

Si realizzi un'applicazione client server web che modifica le specifiche precedenti come segue:

- L'applicazione supporta registrazione e login mediante una pagina pubblica con opportune form. La registrazione controlla la validità sintattica dell'indirizzo di email e l'uguaglianza tra i campi "password" e "ripeti password", anche a lato client. La registrazione controlla l'unicità dello username.
- Dopo il login, l'intera applicazione è realizzata con un'unica pagina per ciascuno dei ruoli: una pagina singola per il ruolo di cliente e una pagina singola per il ruolo di impiegato.
- Ogni interazione dell'utente è gestita senza ricaricare completamente la pagina, ma produce l'invocazione asincrona del server e l'eventuale modifica del contenuto da aggiornare a seguito dell'evento.
- Nella pagina del cliente, la scelta del prodotto comporta la successiva visualizzazione delle opzioni senza produrre un'ulteriore chiamata al server. L'invio del preventivo da parte del cliente deve produrre la verifica dei dati anche a lato client (almeno un'opzione scelta).
- Nella pagina dell'impiegato, il controllo del prezzo (non nullo e maggiore di zero) deve essere fatto anche a lato client.
- Eventuali errori a lato server devono essere segnalati mediante un messaggio di allerta all'interno della pagina del cliente o dell'impiegato.

# RIA Components



## **Controllers (Servlets)**

- HomeCustomerJS
- HomeEmployeeJS
- GetQuotations
- GetOptions
- GetImage



## **View (Single Pages)**

- index
- Signup
- HomeCustomerJS
- HomeEmployeeJS



## **Scripts (JavaScript)**

- Sign Up
- HomeCustomer
- HomeEmployee