

Concepteur Développeur Informatique

Soutenance de stage

Stagiaire:
Houssam EL ATMANI

Sommaire

- Présentation stagiaire
- Présentation entreprise Easy Cab
- Présentation projet de stage

Présentation stagiaire

Présentation entreprise

Présentation projet

Projet EasyCab

- Cahier des charges
- Conception avec UML
- Persistance des données
- Architecture n-tiers
- IHM (interface homme-machine)

Cahier des charges



Conception avec UML



Conception avec UML

Package Client

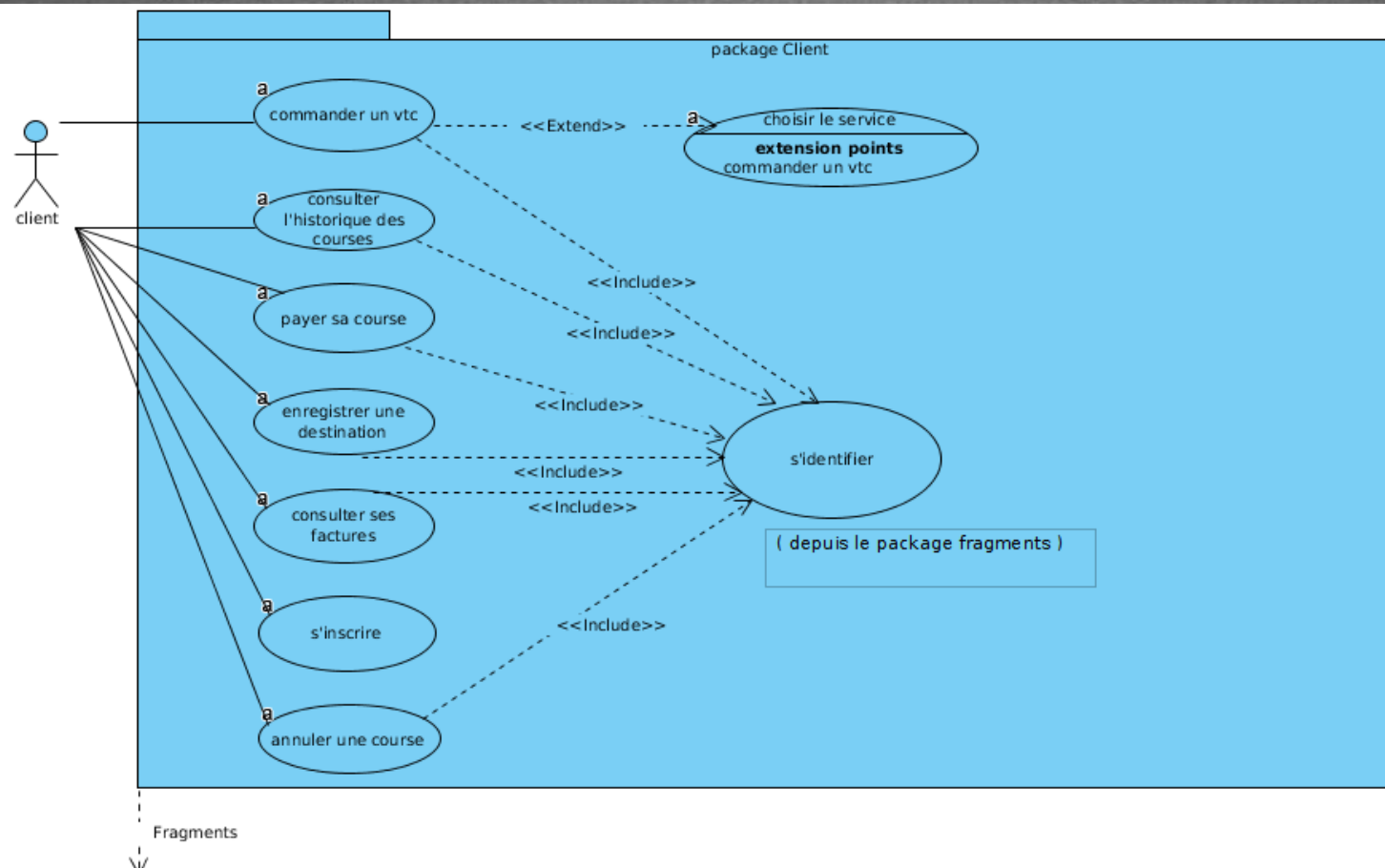
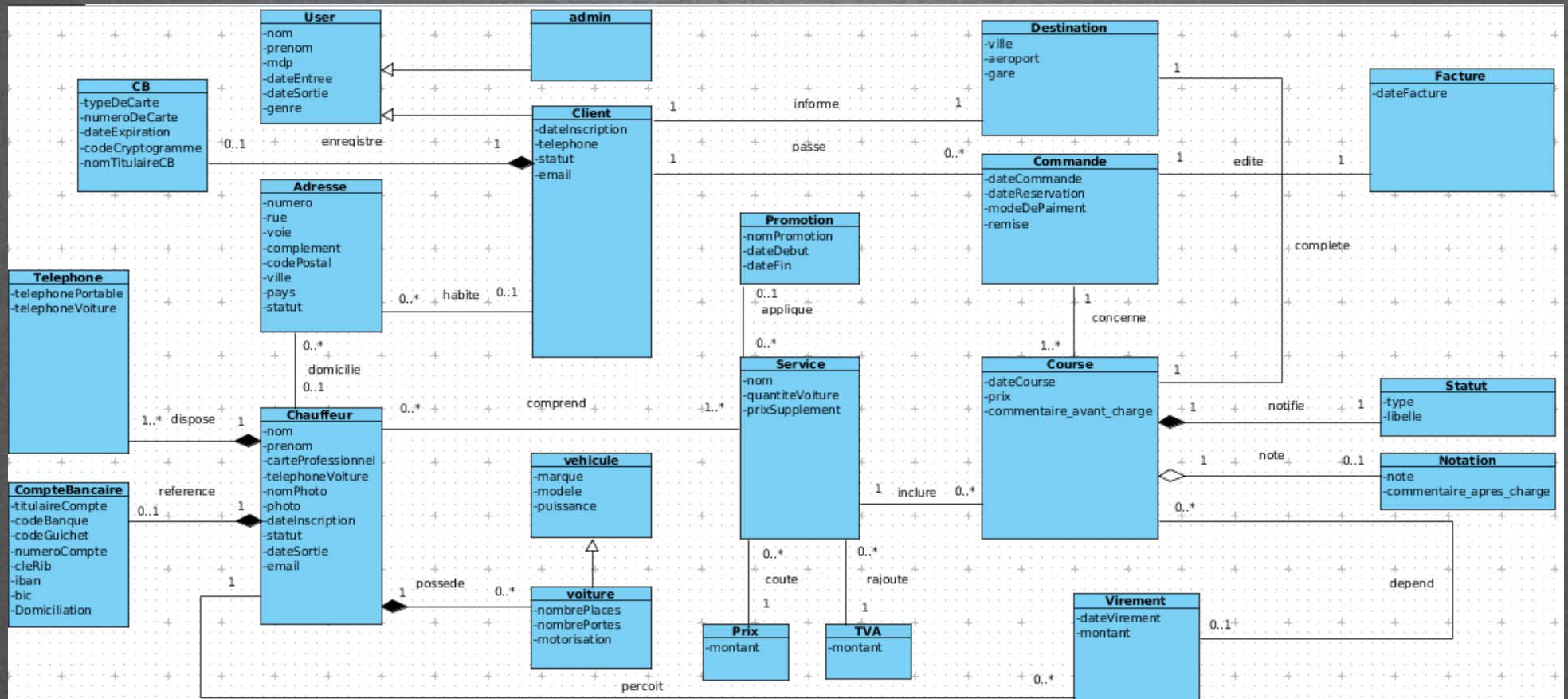
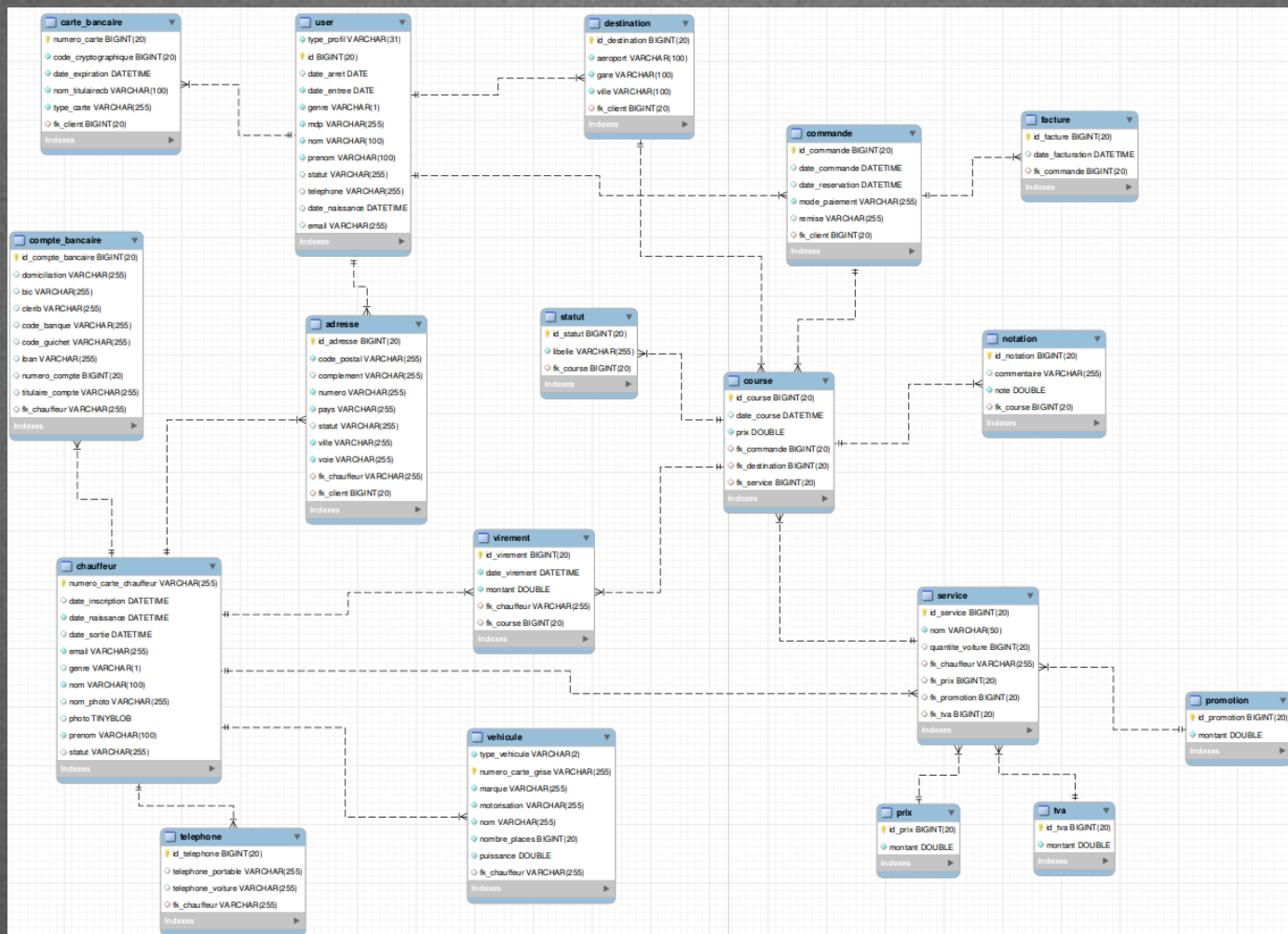


Diagramme de classe



Conception avec UML

Modèle logique de données



Persistance des données




Persistance des données

- MySQL
- JPA
- Jeu de données

Persistance des données

Classe Virement

```
 @Entity
23 public class Virement implements Serializable {
24     @Id
25     @GeneratedValue()
26     private Long idVirement;
27
28     @NotEmpty
29     @Temporal(TemporalType.TIMESTAMP)
30     private Date dateVirement;
31
32     @NotEmpty
33     private Double montant;
34
35     @OneToOne
36     @JoinColumn(name="fk_course")
37     private Course course;
38
39     @ManyToOne
40     @JoinColumn(name="fk_chauffeur")
41     private Chauffeur chauffeur;
42
43
44     public Virement() {
45     }
46
47     public Virement(Date dateVirement, Double montant,
48     this.dateVirement = dateVirement;
49     this.montant = montant;
```


Architecture n-tiers



Architecture n-tiers

- Couche DAO
- Couche Metier
- Couche Web

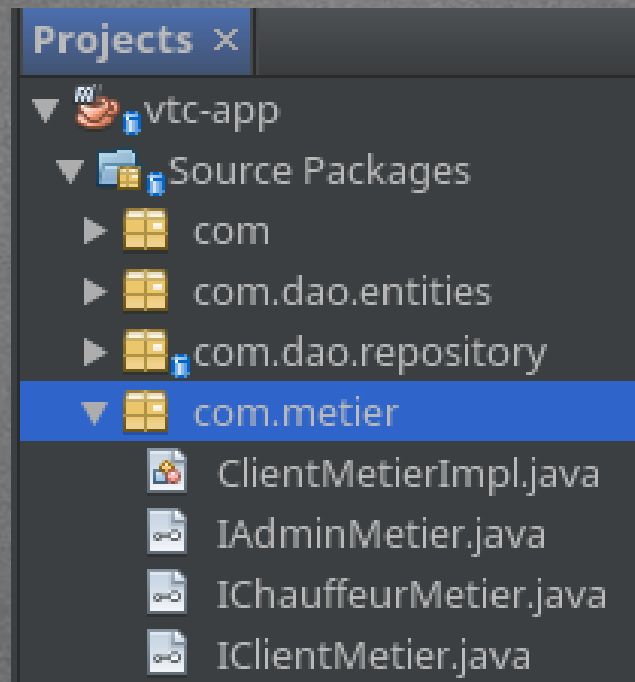
Architecture n-tiers

Couche DAO



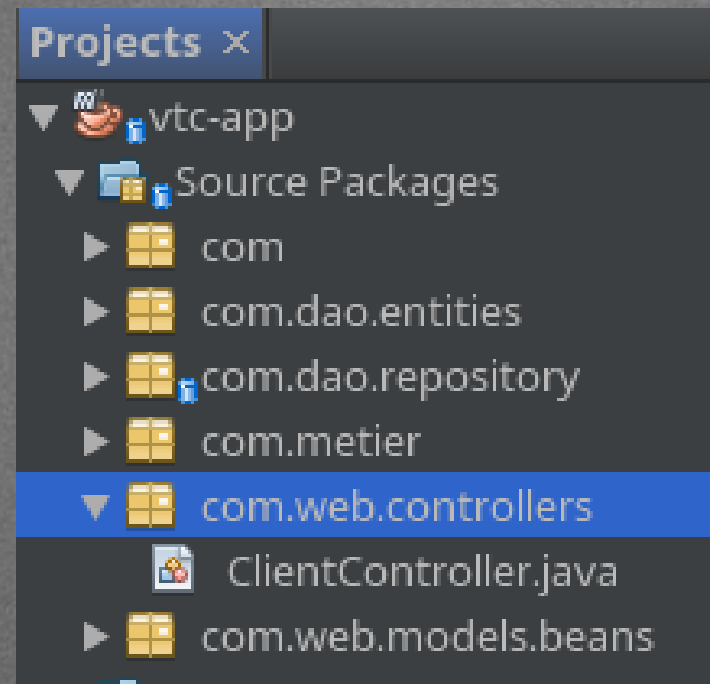
Architecture n-tiers

Couche Metier



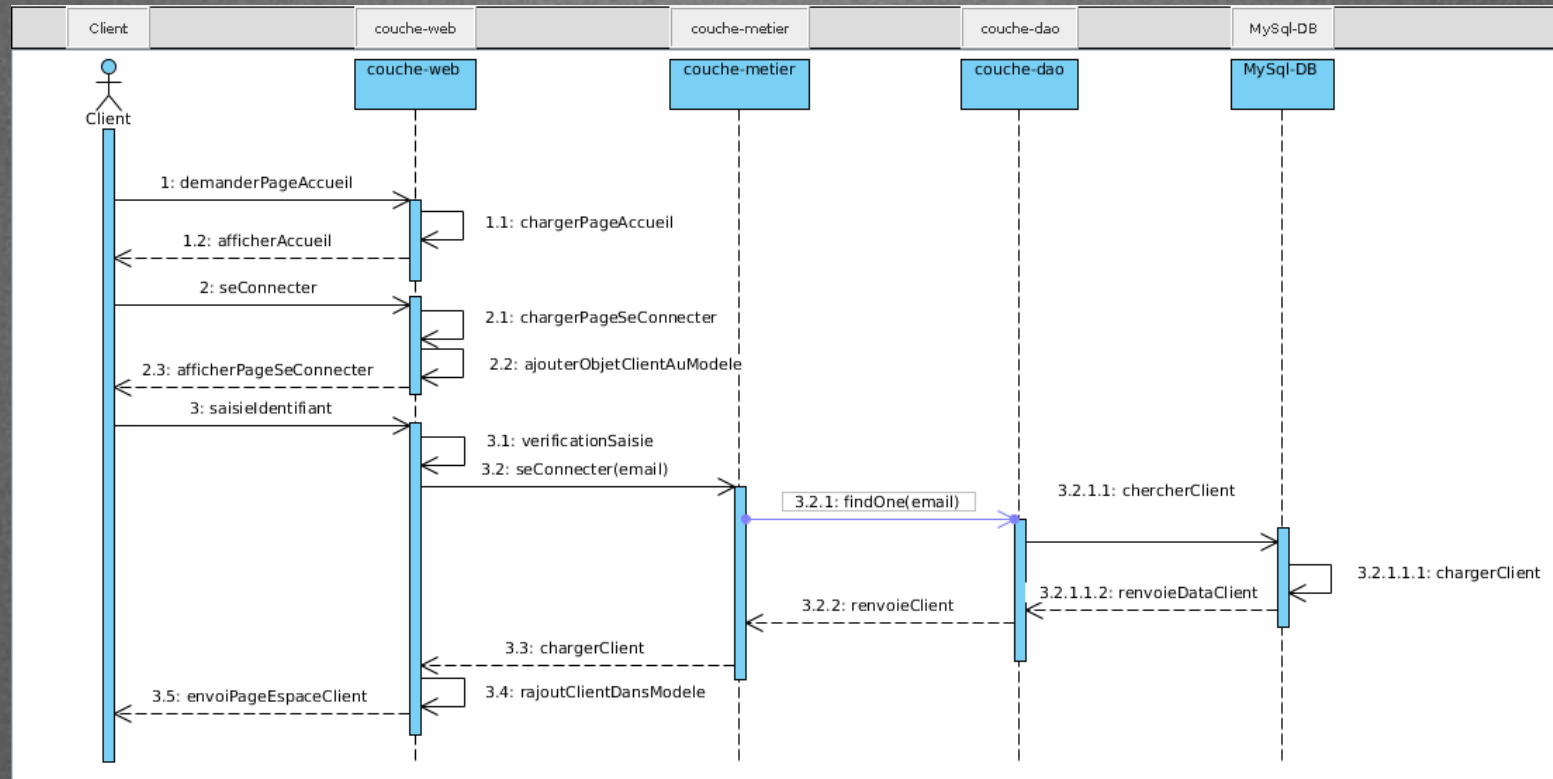
Architecture n-tiers

Couche web

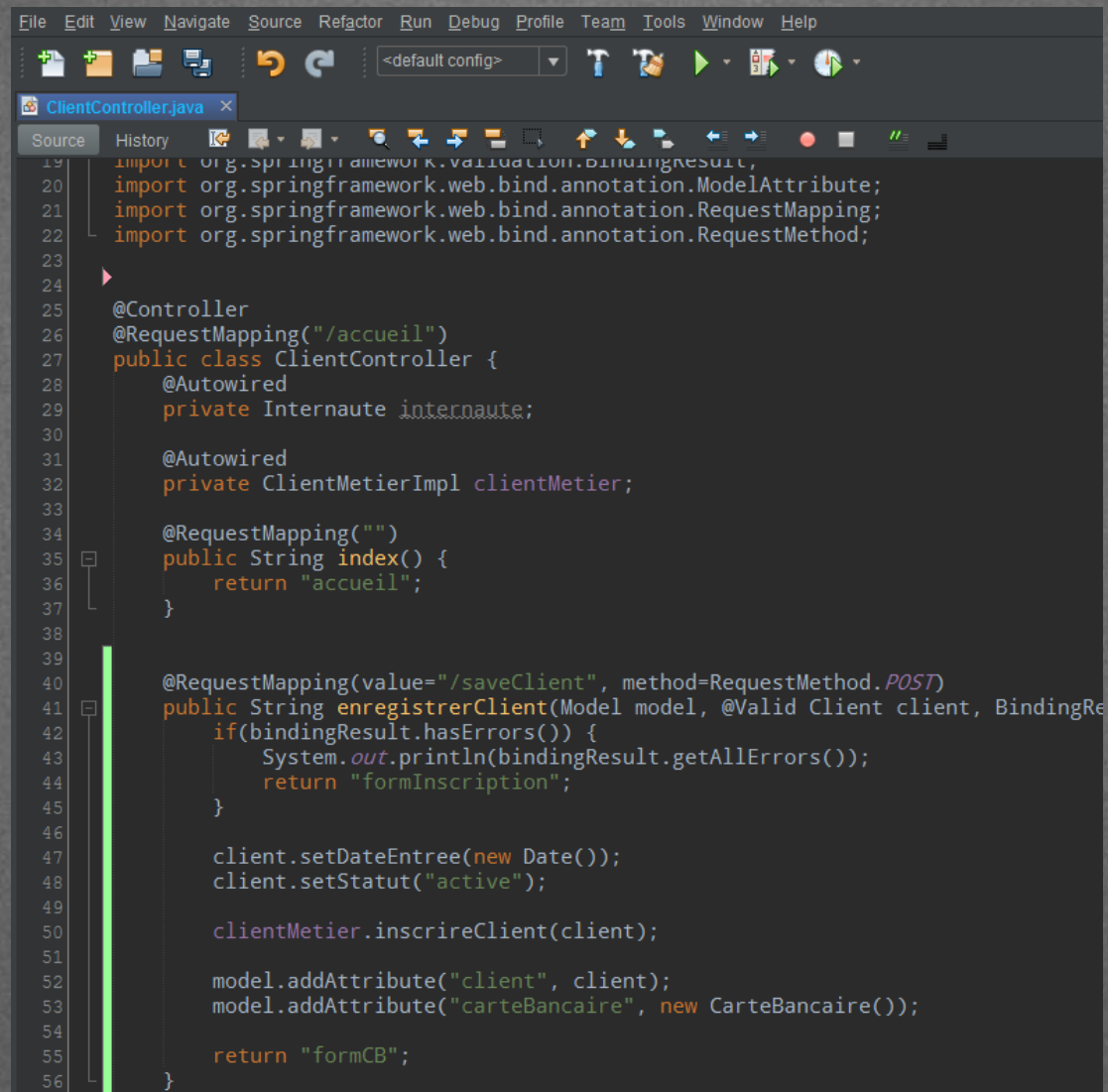


Architecture n-tiers

Diagramme de séquence



Architecture n-tiers



```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config>
ClientController.java x
Source History
19 import org.springframework.validation.BindingResult;
20 import org.springframework.web.bind.annotation.ModelAttribute;
21 import org.springframework.web.bind.annotation.RequestMapping;
22 import org.springframework.web.bind.annotation.RequestMethod;
23
24
25 @Controller
26 @RequestMapping("/accueil")
27 public class ClientController {
28     @Autowired
29     private Internaute internaute;
30
31     @Autowired
32     private ClientMetierImpl clientMetier;
33
34     @RequestMapping("")
35     public String index() {
36         return "accueil";
37     }
38
39
40     @RequestMapping(value="/saveClient", method=RequestMethod.POST)
41     public String enregistrerClient(Model model, @Valid Client client, BindingResult bindingResult) {
42         if(bindingResult.hasErrors()) {
43             System.out.println(bindingResult.getAllErrors());
44             return "formInscription";
45         }
46
47         client.setDateEntree(new Date());
48         client.setStatut("active");
49
50         clientMetier.inscrireClient(client);
51
52         model.addAttribute("client", client);
53         model.addAttribute("carteBancaire", new CarteBancaire());
54
55         return "formCB";
56     }
}
```

Architecture n-tiers

```
1 package com.metier;
2
3 import com.dao.entities.Client;
4 import com.dao.repository.IClientRepository;
5 import javax.transaction.Transactional;
6 import org.springframework.beans.factory.annotation.Autowired;
7 import org.springframework.stereotype.Service;
8
9 @Service
10 @Transactional
11 public class ClientMetierImpl implements IClientMetier {
12     @Autowired
13     IClientRepository clientRepository;
14
15     @Override
16     public Client inscrireClient(Client client) {
17         if(client != null) {
18             clientRepository.save(client);
19             return client;
20         }
21         else {
22             throw new RuntimeException("client invalide ");
23         }
24     }
25
26     @Override
27     public Client connecterClient(String email) {
28         if(email != null || !email.isEmpty()) {
29             Client c = clientRepository.findByEmail(email);
30             return c;
31         }
32         else {
33             throw new RuntimeException("client introuvable ");
34         }
35     }
36
37 }
38 }
```

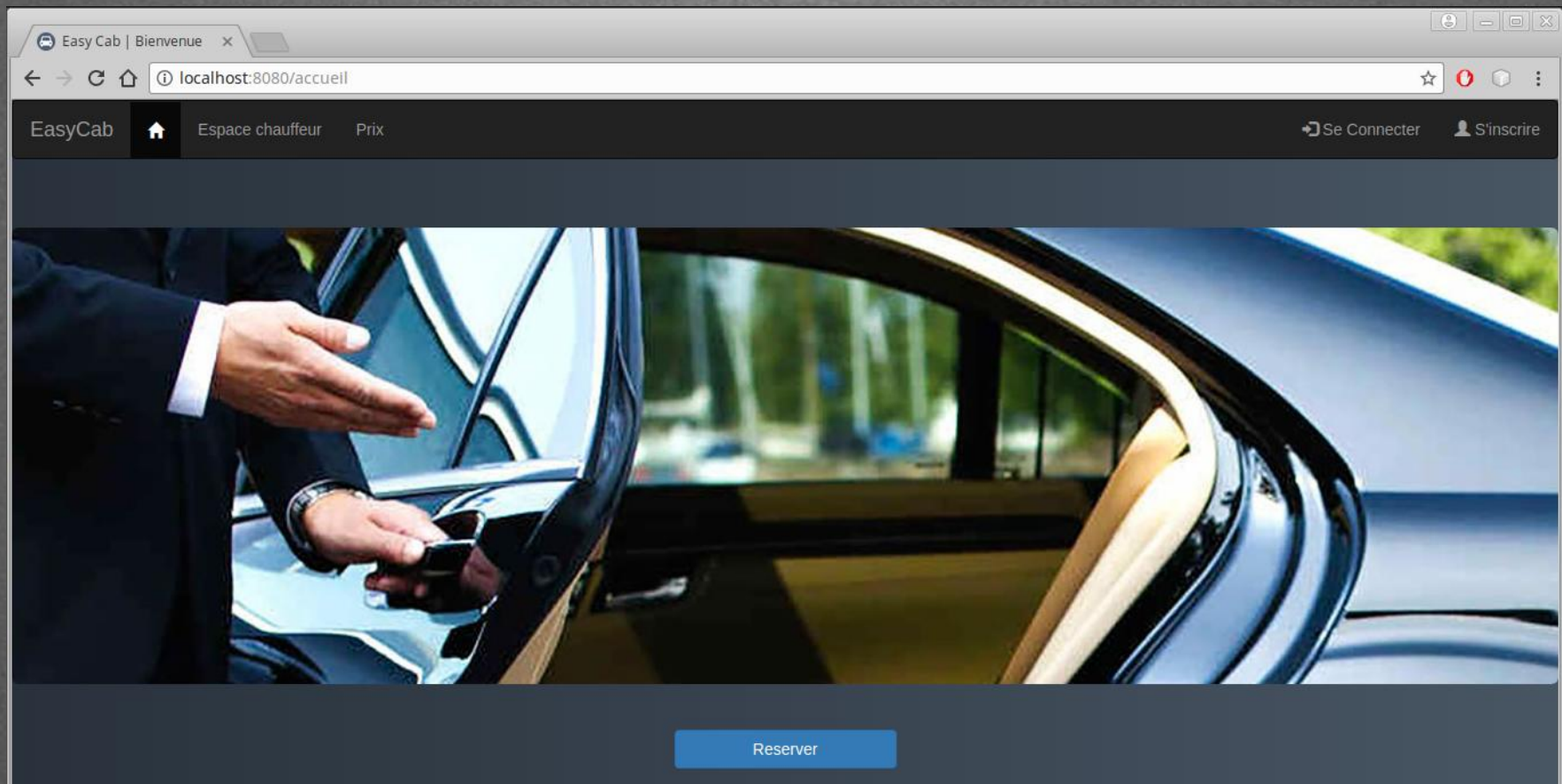
```
1 /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6
7 package com.dao.repository;
8
9 import com.dao.entities.Client;
10 import org.springframework.data.jpa.repository.JpaRepository;
11 import org.springframework.data.jpa.repository.Query;
12 import org.springframework.data.repository.query.Param;
13
14
15 public interface IClientRepository extends JpaRepository<Client, Long> {
16
17     @Query("select c from Client c WHERE c.email = :x")
18     public Client findByEmail(@Param("x") String email);
19
20 }
21 }
```

Interface Homme-Machine



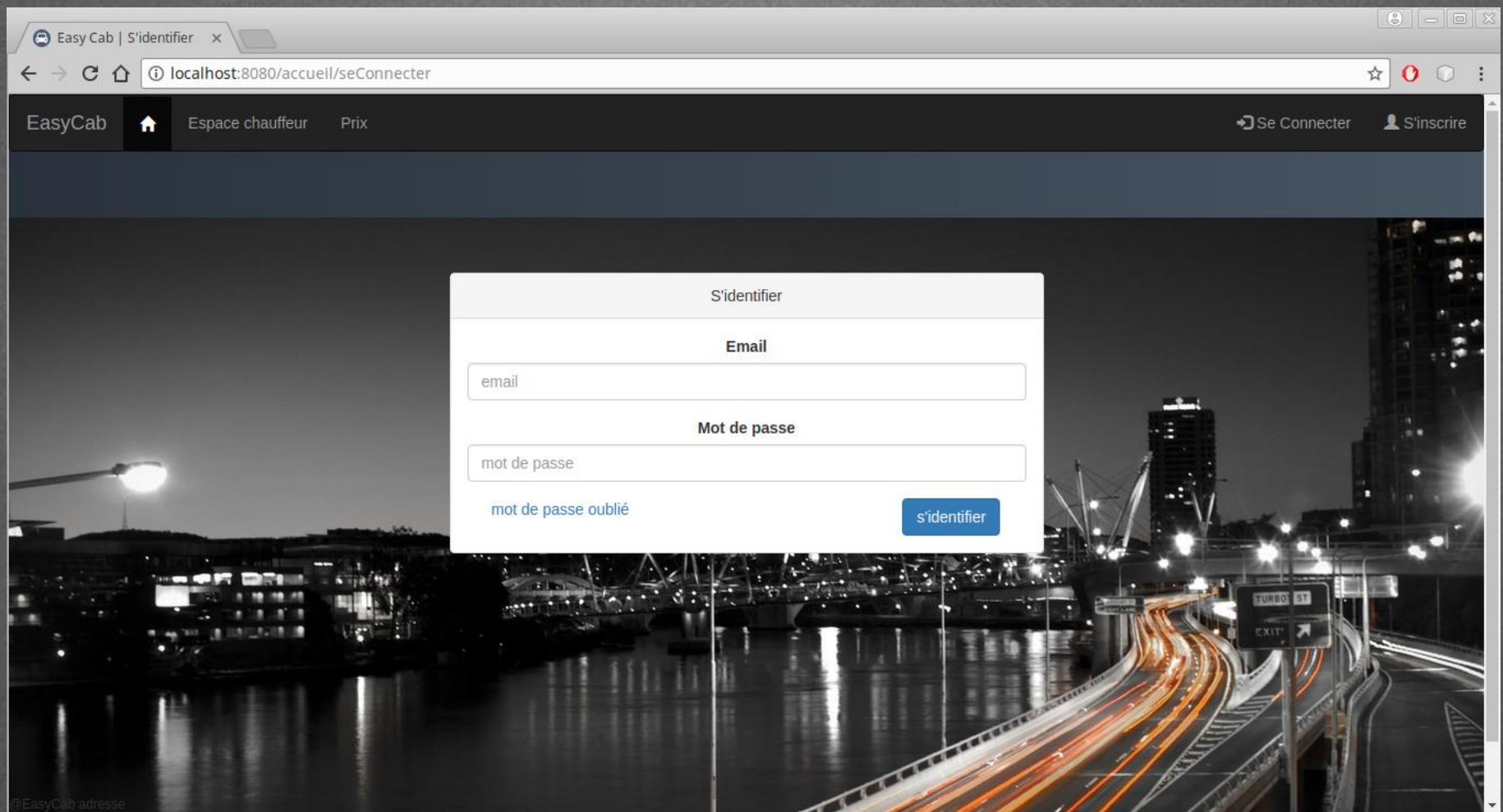
IHM

Page Accueil



IHM

Page Connexion



The screenshot displays a web browser window with the URL `localhost:8080/accueil/seConnecter`. The page features a dark navigation bar with the 'EasyCab' logo, a home icon, and links for 'Espace chauffeur' and 'Prix'. On the right side of the navigation bar are links for 'Se Connecter' and 'S'inscrire'. The main content area has a background image of a city at night with light trails from traffic. A white login form is centered on the page, titled 'S'identifier'. It contains two input fields: 'Email' (with the placeholder text 'email') and 'Mot de passe' (with the placeholder text 'mot de passe'). Below the password field is a link for 'mot de passe oublié' and a blue 's'identifier' button.

EasyCab | S'identifier

localhost:8080/accueil/seConnecter

EasyCab

Espace chauffeur Prix

Se Connecter S'inscrire

S'identifier

Email

email

Mot de passe

mot de passe

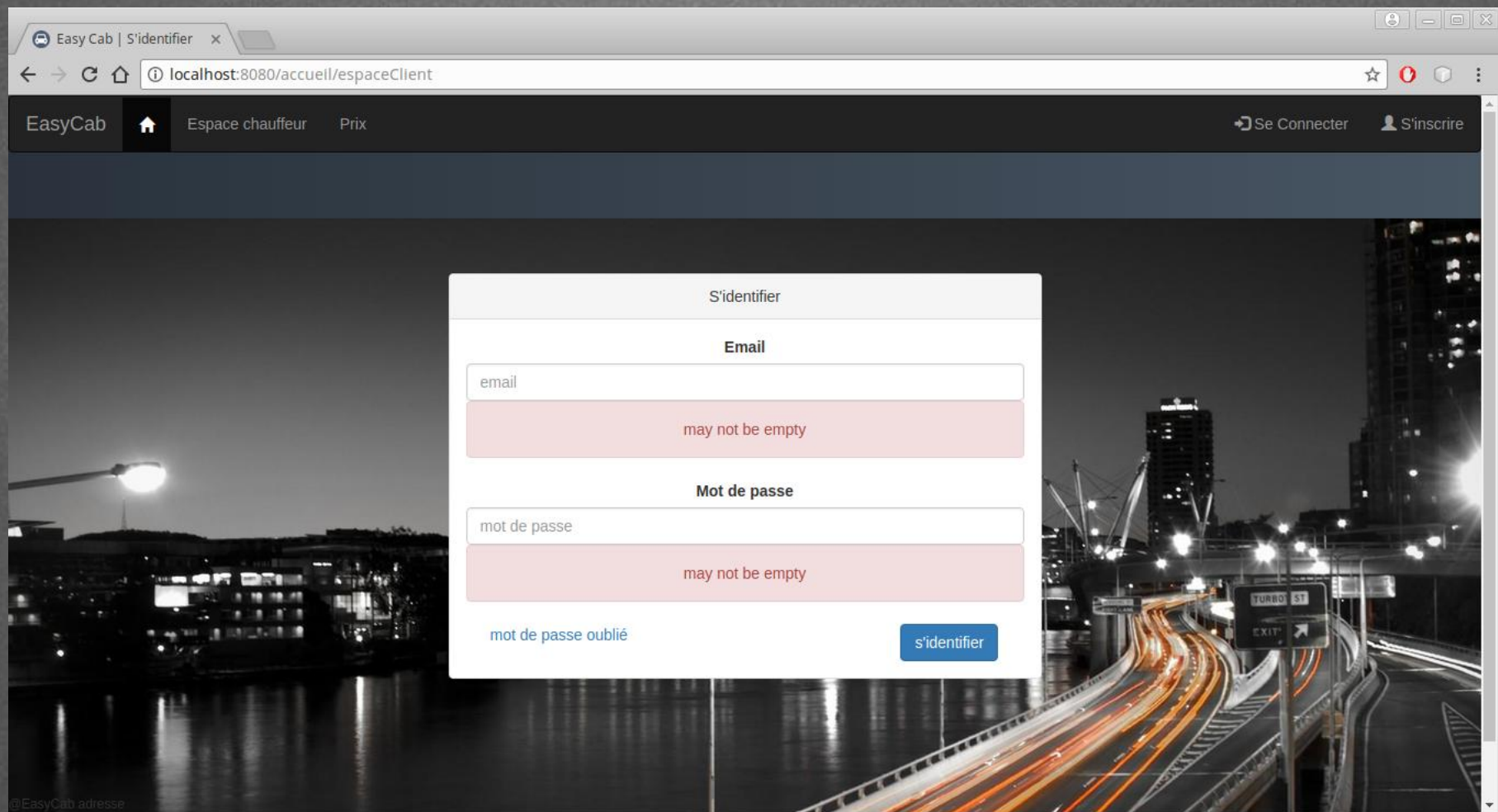
[mot de passe oublié](#)

[s'identifier](#)

EasyCab adresse

IHM

Essai avec saisie vide



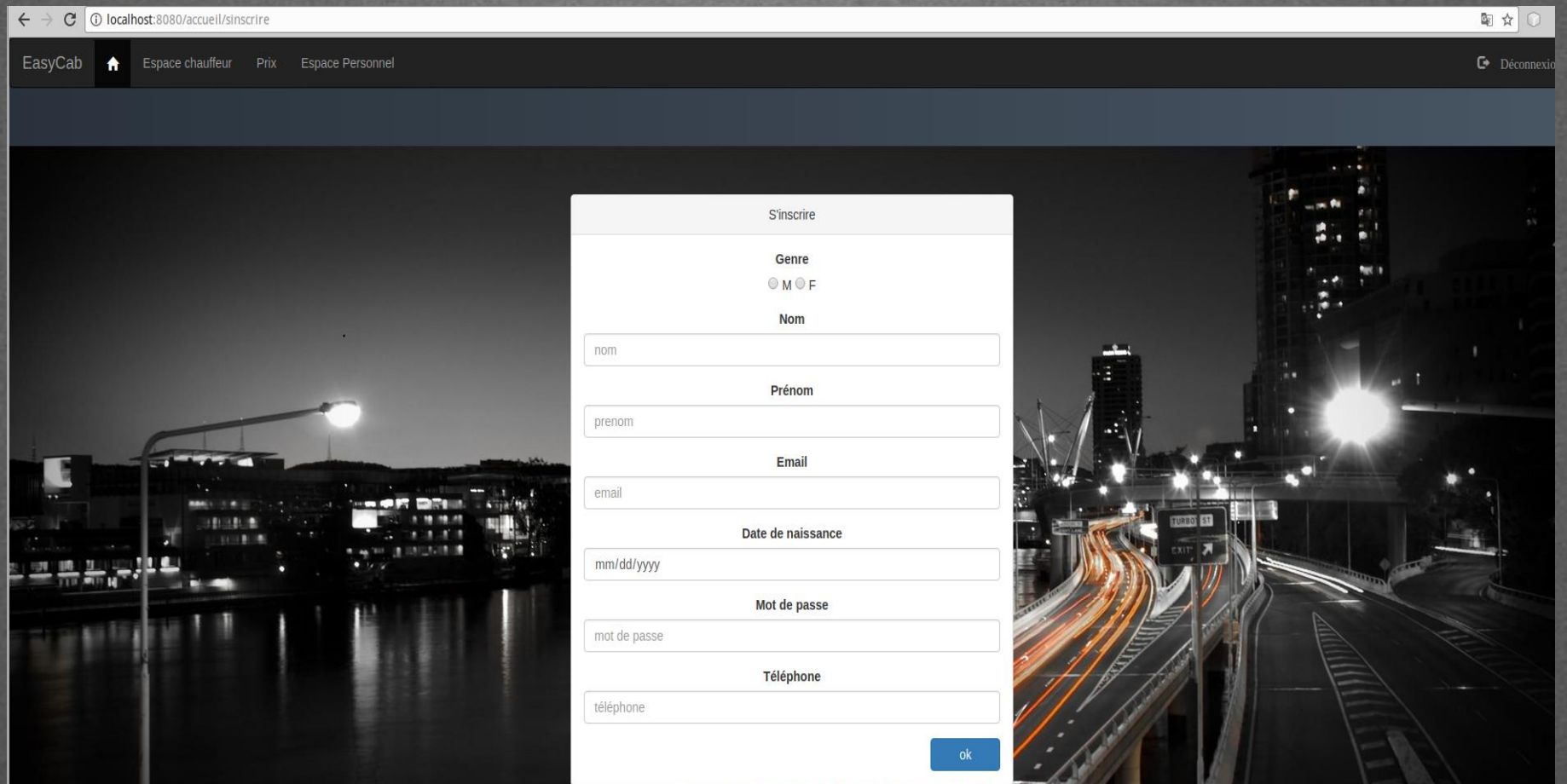
The screenshot shows a web browser window with the title "Easy Cab | S'identifier". The address bar displays "localhost:8080/accueil/espaceClient". The navigation bar includes links for "EasyCab", "Espace chauffeur", and "Prix", along with "Se Connecter" and "S'inscrire". The main content area features a login form titled "S'identifier" with the following elements:

- Email:** A text input field containing "email". Below it, a red error message states "may not be empty".
- Mot de passe:** A text input field containing "mot de passe". Below it, a red error message states "may not be empty".
- A link labeled "mot de passe oublié" (forgot password).
- A blue button labeled "s'identifier" (login).

The background of the page is a night-time photograph of a city street with light trails from cars.

IHM

Inscription



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/accueil/sinscrire'. The page features a dark navigation bar with the 'EasyCab' logo and links to 'Espace chauffeur', 'Prix', 'Espace Personnel', and 'Déconnexion'. The main content area has a background image of a city at night with a river and a highway. A white registration form is centered on the page, titled 'S'inscrire'. The form includes fields for 'Genre' (M/F), 'Nom', 'Prénom', 'Email', 'Date de naissance' (mm/dd/yyyy), 'Mot de passe', and 'Téléphone', each with a corresponding input box. An 'ok' button is located at the bottom right of the form.

localhost:8080/accueil/sinscrire

EasyCab

Espace chauffeur Prix Espace Personnel

Déconnexion

S'inscrire

Genre

☐ M ☐ F

Nom

nom

Prénom

prenom

Email

email

Date de naissance

mm/dd/yyyy

Mot de passe

mot de passe

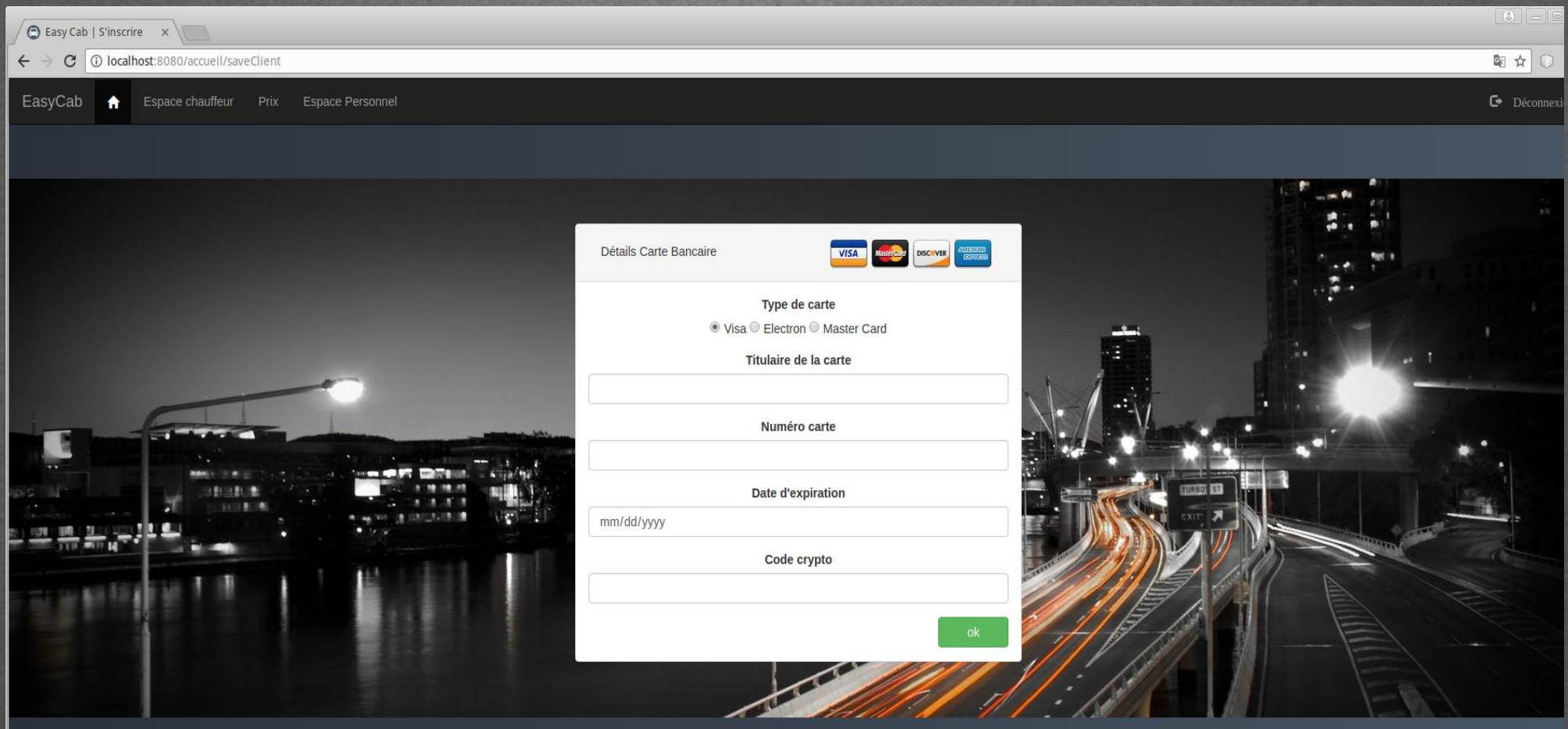
Téléphone

téléphone

ok

IHM

Saisie données Carte Bancaire



The screenshot shows a web browser window with the URL `localhost:8080/accueil/saveClient`. The browser's address bar and tabs are visible. The website has a dark header with the 'EasyCab' logo and navigation links: 'Espace chauffeur', 'Prix', and 'Espace Personnel'. A 'Déconnexion' link is in the top right. The main content area features a large background image of a city at night with light trails from traffic. A white modal form titled 'Détails Carte Bancaire' is centered on the screen. The form includes radio buttons for card type (Visa is selected), input fields for cardholder name, card number, expiration date (with a placeholder 'mm/dd/yyyy'), and a crypto code, followed by an 'ok' button.

EasyCab | S'inscrire

localhost:8080/accueil/saveClient

EasyCab

Espace chauffeur Prix Espace Personnel

Déconnexion

Détails Carte Bancaire

VISA MasterCard DISCOVER AMERICAN EXPRESS

Type de carte

☒ Visa ☐ Electron ☐ Master Card

Titulaire de la carte

Numéro carte

Date d'expiration

mm/dd/yyyy

Code crypto

ok

Conclusion

Questions ?