Projet 6:

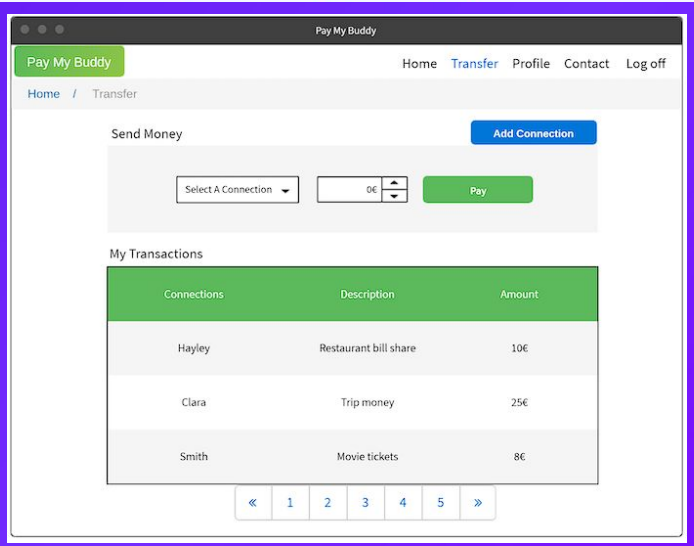
Concevez une application web Java de A à Z

Présentation de l'application

Pay my Buddy :

Transfert d'argent entre amis.

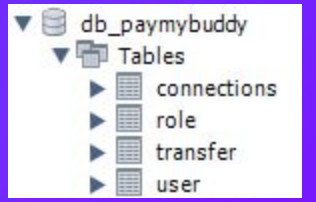
Une interface permettant d'envoyer une somme d'argent à un contact que l'on à enregistrer au préalable.



Développement de l'application

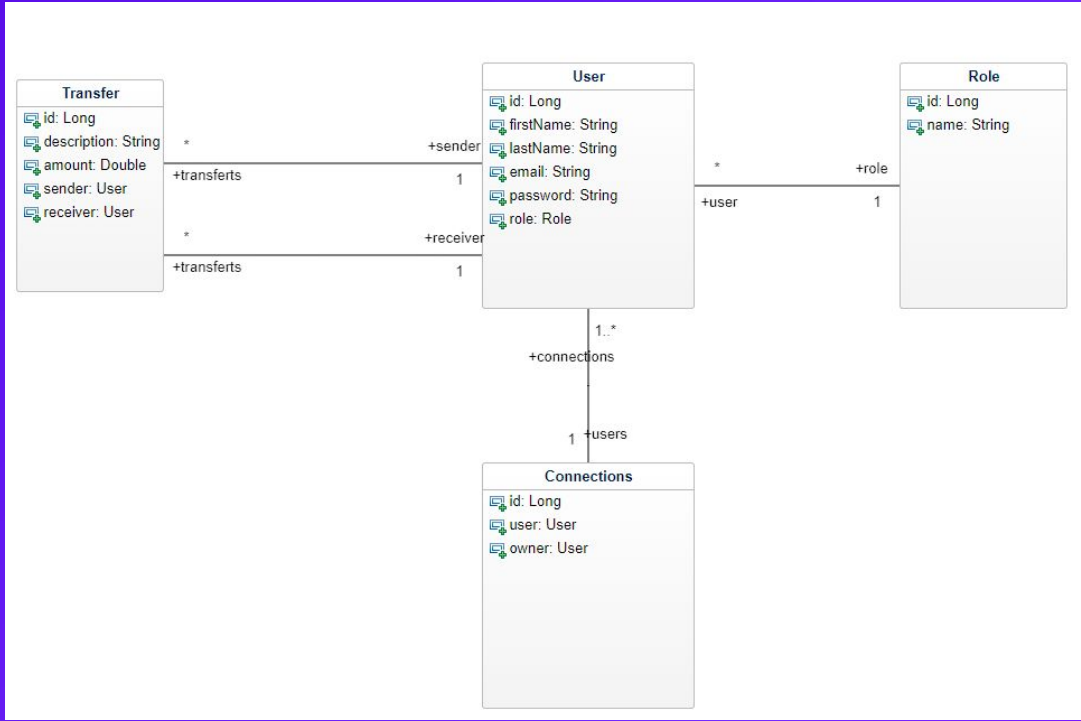
Base de donnée MySQL :

```
server.port=8090

spring.jpa.hibernate.ddl-auto=update
spring.datasource.url=jdbc:mysql://localhost:3306/db_paymybuddy
spring.datasource.username=root
spring.datasource.password=@zertyuiop123
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect

spring.security.oauth2.client.registration.github.client-id=f50e4bab765269d9eb65
spring.security.oauth2.client.registration.github.client-secret=9661c6ee979eec455f524a007a3833bba75ab05c
```



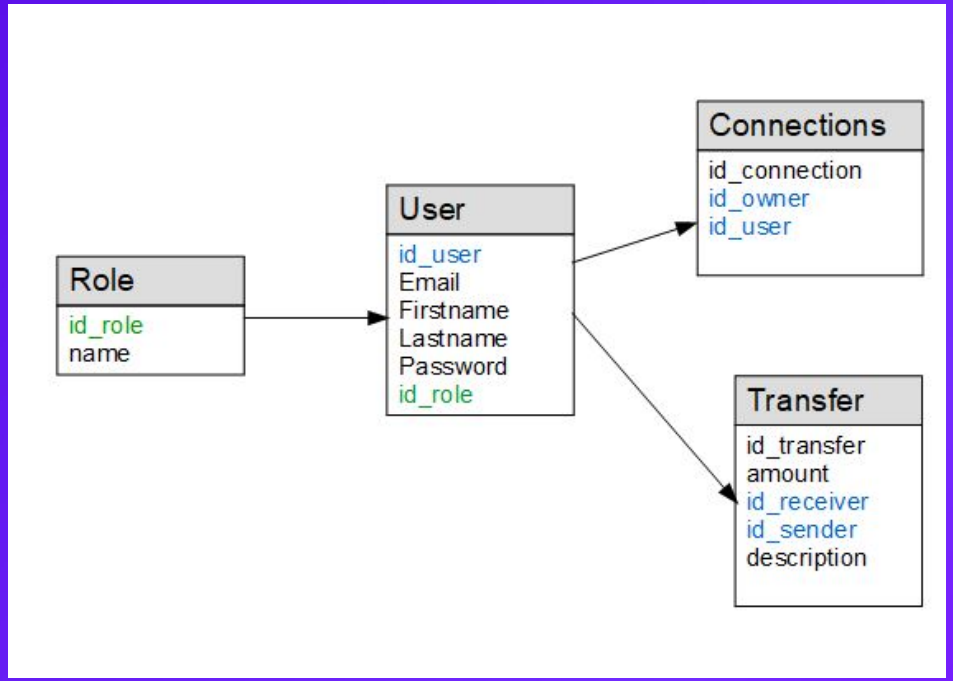
Développement de l'application

Modélisation du domaine métier :



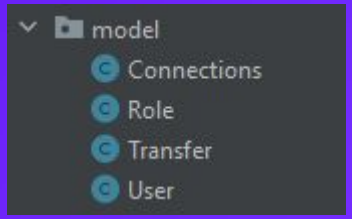
Développement de l'application

Modélisation physique des données :



Développement de l'application

Solution Modele-View-Controller :




```java
@Entity
@Data
@Table(name = "connections")
public class Connections {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @ManyToOne
    User user;

    @ManyToOne
    User owner;

    public Connections(User user, User owner) {
        this.user = user;
        this.owner = owner;
    }

    public Connections() {

    }
}
```

```java
@Entity
@Data
@Table (name = "role")
@AllArgsConstructor
@NoArgsConstructor
public class Role {

    @Id
    @GeneratedValue (strategy = GenerationType.IDENTITY)
    private Long id;

    private String name;


    public Role(String roleName) {
        this.name = roleName;
    }
}
```

```java
@Entity
@Data
@Table(name = "transfer")
public class Transfer {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    private Double amount;

    private String description;

    @ManyToOne
    User sender;

    @ManyToOne
    User receiver;
}
```

```java
@Entity
@Data
@Table (name = "user", uniqueConstraints = @UniqueConstraint(columnNames = "email"))
public class User implements UserDetails {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @Column (name = "first_name")
    private String firstName;

    @Column (name = "last_name")
    private String lastName;

    private String email;

    private String password;

    @ManyToOne(fetch = FetchType.EAGER, cascade = CascadeType.ALL)
    private Role role;

    @Override
    public Collection<? extends GrantedAuthority> getAuthorities() {
        SimpleGrantedAuthority authority = new SimpleGrantedAuthority(role.getName());
        return Collections.singleton(authority);
    }

    @Override
    public String getUsername() { return email; }

    @Override
    public boolean isAccountNonExpired() { return true; }

    @Override
    public boolean isAccountNonLocked() { return true; }

    @Override
    public boolean isCredentialsNonExpired() { return true; }

    @Override
    public boolean isEnabled() { return true; }
}
```

Développement de l'application

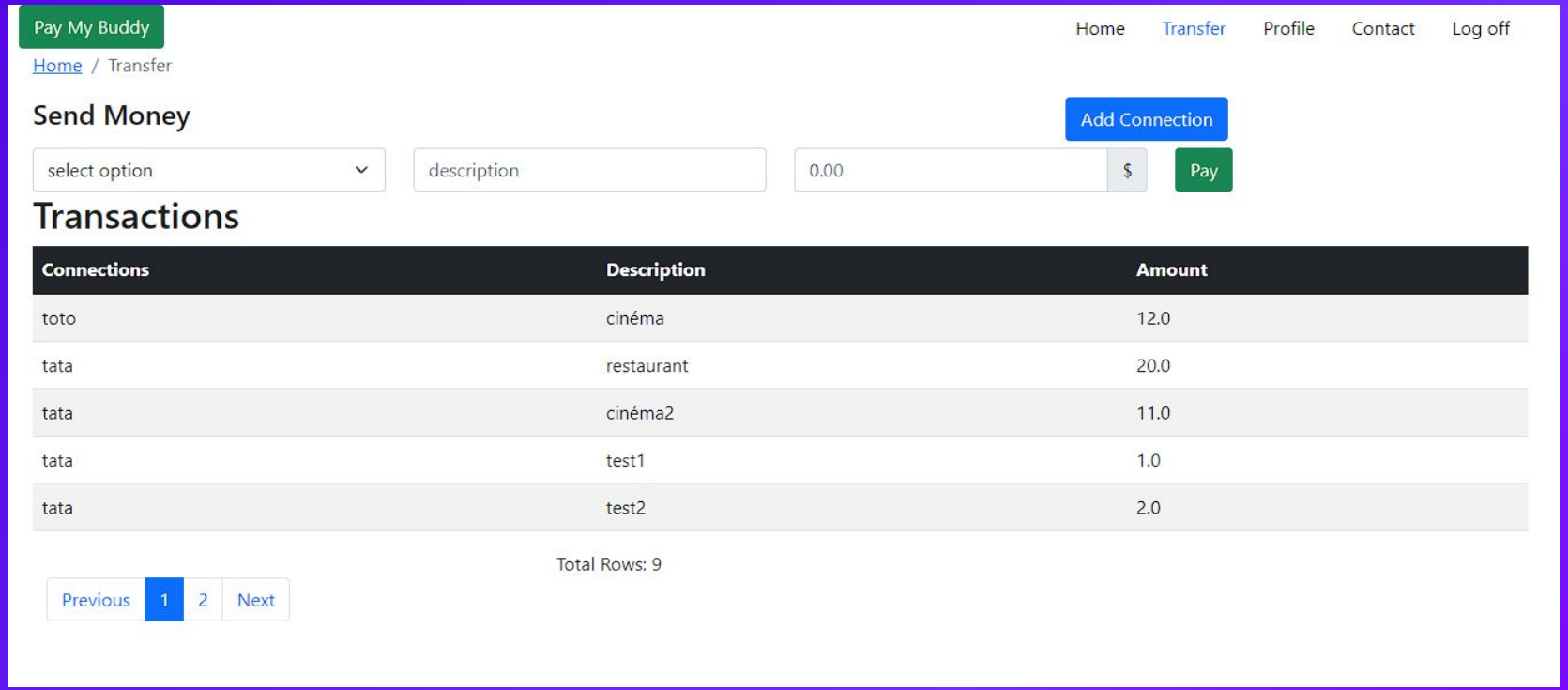
Solution Modele-View-Controller :

```
@GetMapping("/signup")
public String showSignUpForm(User user) {return "signup";}

@PostMapping("/signup")
public String addUser(@Valid User user, BindingResult result) {...}

@GetMapping("/transfer")
public String showUserList(Transfer transfer, Model model,
                           @RequestParam("page") Optional<Integer> page,
                           @RequestParam("size") Optional<Integer> size) {...}

@PostMapping("/update/{id}")
public String updateUser(@PathVariable("id") long id, @Valid User user,
                         BindingResult result, Model model) {...}

@GetMapping("/delete/{id}")
public String deleteUser(@PathVariable("id") long id, Model model) {...}

@PostMapping("/user/transfer")
public String doTransfer(Transfer transfer, BindingResult result, Model model) {...}

@GetMapping("/user/addconnection")
public String showAddConnection() { return "/addconnection"; }

@GetMapping("/user/profile")
public String showProfile(Model model) {...}

@PostMapping("/user/addconnection")
public String addConnection(@RequestParam String email, Model model){...}
```

Développement de l’application

Solution Modele-View-Controller :

/transfer :



Développement de l'application

Solution Modele-View-Controller :

/transfer :

```
@GetMapping("/transfer")
public String showUserList(Transfer transfer, Model model,
                           @RequestParam("page") Optional<Integer> page,
                           @RequestParam("size") Optional<Integer> size) {
    UserDetails userDetails = (UserDetails) SecurityContextHolder.getContext().getAuthentication().getPrincipal();
    User connectedUser = userRepo.findByEmail(userDetails.getUsername()).get();
    model.addAttribute( attributeName: "connections", connectionsRepo.findUserConnections(connectedUser.getId()));


    int currentPage = page.orElse( other: 1);
    int pageSize = size.orElse( other: 5);

    Pageable pageable = PageRequest.of( page: currentPage-1, pageSize);

    Page<Transfer> transferPage = transferRepo.findBySender(connectedUser,pageable);

    List < Transfer > listTransferts = transferPage.getContent();

    model.addAttribute( attributeName: "currentPage", currentPage);
    model.addAttribute( attributeName: "totalPages", transferPage.getTotalPages());
    model.addAttribute( attributeName: "totalItems", transferPage.getTotalElements());
    model.addAttribute( attributeName: "transfers", listTransferts);
    return "transfer";
}
```

Développement de l'application

Solution Modele-View-Controller :

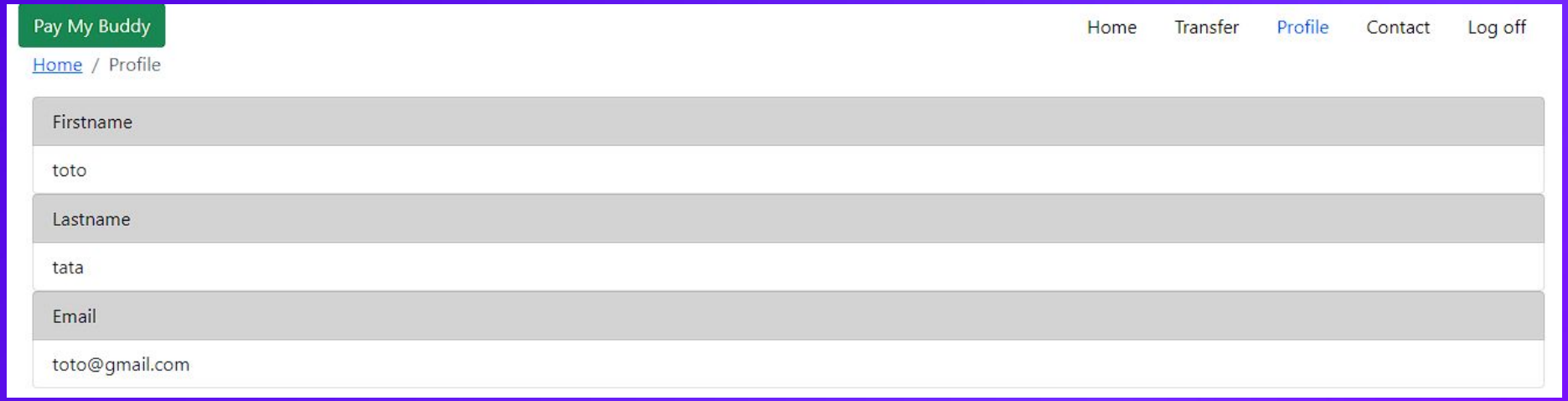
/transfer :

```
@PostMapping("/user/transfer")
public String doTransfer(Transfer transfer, BindingResult result, Model model) {
    if (result.hasErrors()) {
        return "index";
    }

    UserDetails userDetails = (UserDetails) SecurityContextHolder.getContext().getAuthentication().getPrincipal();
    User sender = userRepo.findByEmail(userDetails.getUsername()).get();
    transfer.setSender(sender);
    transfer.setAmount(transfer.getAmount() - 0.005 * transfer.getAmount());
    transferRepo.save(transfer);
    return "redirect:/transfer?success";
}
```

Développement de l'application

Solution Modele-View-Controller :

/profile :



Développement de l'application

Solution Modele-View-Controller :

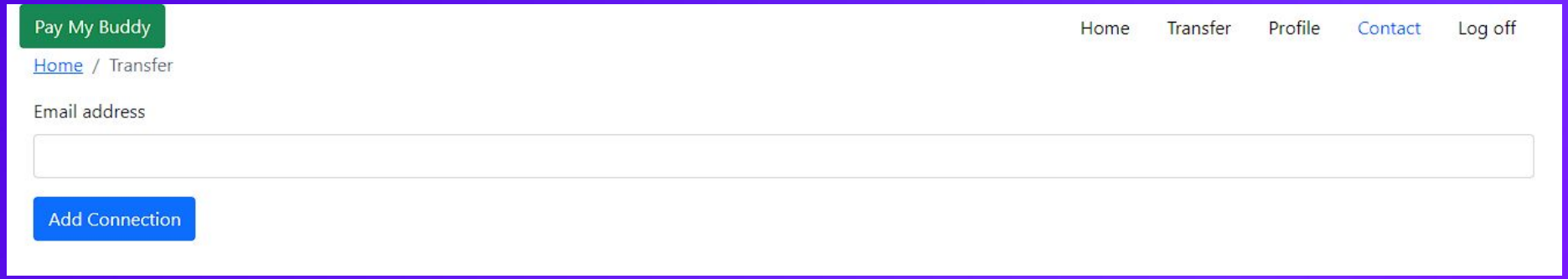
/profile :

```
@GetMapping("/user/profile")
public String showProfile(Model model) {

    UserDetails userDetails = (UserDetails) SecurityContextHolder.getContext().getAuthentication().getPrincipal();

    User owner = userRepo.findByEmail(userDetails.getUsername()).get();
    model.addAttribute( attributeName: "currentUser",owner);

    return "/profile";
}
```

Développement de l'application

Solution Modele-View-Controller :

/user/addconnection :



Développement de l'application

Solution Modele-View-Controller :

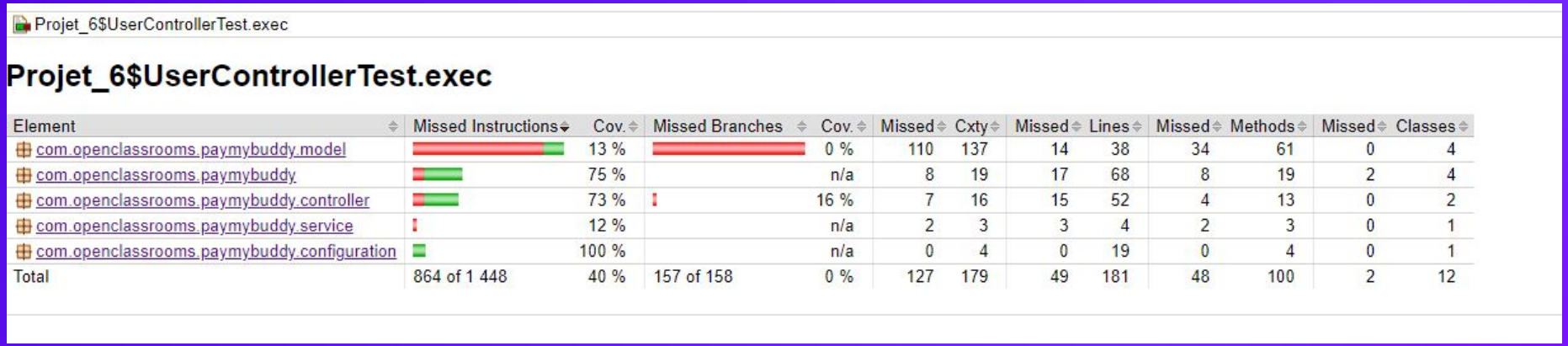
/user/addconnection :

```
@GetMapping("/user/addconnection")
public String showAddConnection() { return "/addconnection"; }
```

```
@PostMapping("/user/addconnection")
public String addConnection(@RequestParam String email, Model model){

    User user = userRepo.findByEmail(email).get();

    UserDetails userDetails = (UserDetails) SecurityContextHolder.getContext().getAuthentication().getPrincipal();

    User owner = userRepo.findByEmail(userDetails.getUsername()).get();

    Connections c = new Connections(user, owner);

    connectionsRepo.save(c);

    return "redirect:/transfer?success";

}
```

Développement de l'application

Solution Modele-View-Controller :



Projet_6$UserControllerTest.exec

Projet_6$UserControllerTest.exec

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
com.openclassrooms.paymybuddy.model		13 %		0 %	110	137	14	38	34	61	0	4
com.openclassrooms.paymybuddy		75 %		n/a	8	19	17	68	8	19	2	4
com.openclassrooms.paymybuddy.controller		73 %		16 %	7	16	15	52	4	13	0	2
com.openclassrooms.paymybuddy.service		12 %		n/a	2	3	3	4	2	3	0	1
com.openclassrooms.paymybuddy.configuration		100 %		n/a	0	4	0	19	0	4	0	1
Total	864 of 1 448	40 %	157 of 158	0 %	127	179	49	181	48	100	2	12