**Booking Service**

1. **Task statement of objectives and basic requirements**

In the course of this project, it was necessary to develop a website for booking services. The website should allow the user to register, browse the pages of the website, filter the list of services, and also leave a booking for the selected service. The development of the external presentation of the website is done on the markup language HTML and CSS, using the template engine Thymeleaf. The server side is developed on the object-oriented programming language Java, using the universal framework for Java - Spring Framework. We used Amazon RDS cloud database to store the website's information. Relational database management system (DBMS) MySQL was used to connect to the database and for the work. The development environment was IntelliJ IDEA Community Edition. UML diagrams were modeled using Visual Paradigm tool.

1. **Architecture**

This project uses the Model-View-Controller (MVC) software architectural pattern

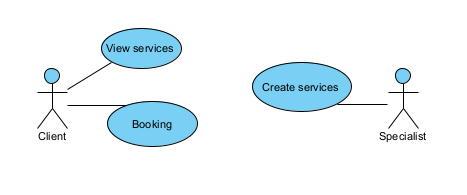
The developed website will need to display and work correctly in the latest versions of browsers such as Google Chrome, Opera and Microsoft Edge.

The MVC template is used to implement the website. An MVC is an architectural design pattern for a web application consisting of three separate components, each with its own function in the application being developed. It also allows the project to be divided into logical parts. This structure allows you to work on each object separately from the others without affecting their work, making it easy to change elements.

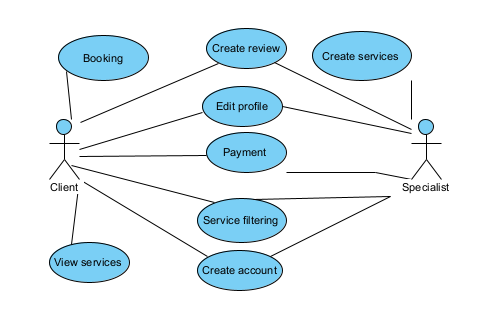
1. **Designing**

Agile methodology was used in the process of work on the project

**3.1 Use case**



Pic.3.1. Initial Use Case diagram



Pic.3.2. Extended use case diagram

**3.2 Actor Identification**

The actors of the booking website are:

- Client - a user who goes to the website in search of services and to book them

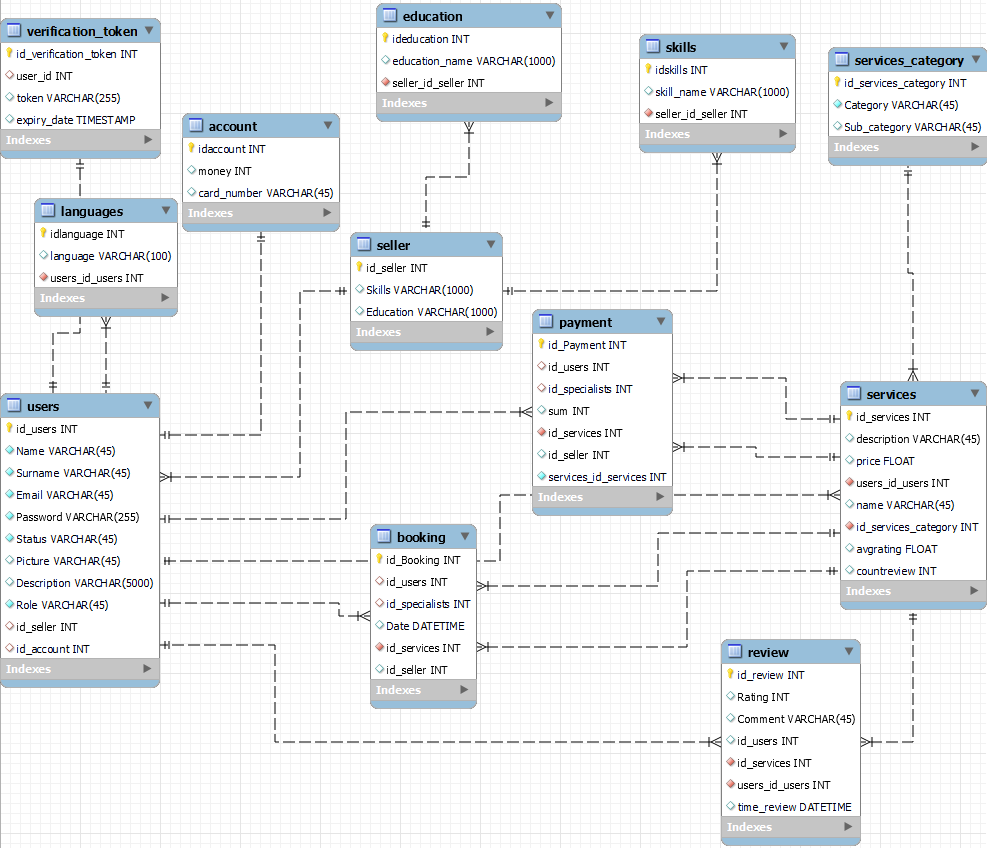
- Specialist - a user who provides services to others

**3.3 Identification and description of Use Case diagram elements**

|  |  |
| --- | --- |
| **Element name** | **Description** |
| Create review | Contains all the actions required to create a new review. |
| Edit profile | Contains all actions that are required to change user information. |
| Payment | Contains all actions that are required for payment. |
| Service filtering | Contains all actions that are required to control data filtering. |
| Create account | Contains all the steps required to create an account. |
| Booking | Contains all actions required to create a new booking |
| View services | Contains all actions that are required to view the list of services. |
| Create services | Contains all actions required to create a new service. |

**3.3 Database Design**

Entity-Relationship (ER)-the database model is represented as Martin's Notation, also known as Crow's Foot Notation, in which entities are represented as rectangles within which attributes are written, and relationships are displayed as lines at the ends of which the relationship type is specified.

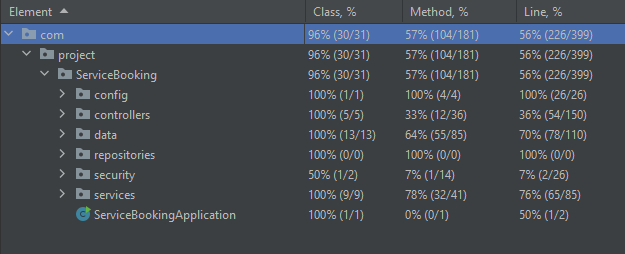


Pic.3.3. Database Design

1. **TESTING**

To verify the correctness of the application's operation, white-box and black-box methodology tests were performed. For testing using the white-box test approach, unit tests were performed to check the individual modules of the program. The Mockito framework was used for testing. Functional testing was performed for testing using the black-box approach.

* 1. **Unit Testing**



Pic.4.1. Test coverage rate

* 1. **Functional testing**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Test Case Description** | **Test Steps** | **Expected results** | **Results obtained** | **Success/Error** |
| 1 | Editing Profile | 1. Login to the site  2. Open the profile page  3. Click on edit  4. Change info  5. Save сhanges | Changed information on the profile page. | Changed information on the profile page. | **Success** |
| 2 | Opening the registration form | 1. Go to home page  2. Goes to the Login page  3. Go to the registration page | The page is displayed correctly. | The page is displayed correctly. | **Success** |
| 3 | Checking Registration. | 1. Go to the registration page  2. Enter data  3. Press registration  4. Activate your account by the link in the email | The user will enter in the login data and it will successfully log in. | The user will enter in the login data and it will successfully log in. | **Success** |
| 4 | Creating reviews | 1. Login to the site  2. Go to the service page  3. Enter information in the field and select a rating.  4. Save review | A new review appeared on the service page | A new review appeared on the service page | **Success** |
| 5 | Filtering on the services page | 1. Go to services page  2. Enter data for filtering.  3. Confirm. | The page will display services according to the filtering conditions. | The page will display services according to the filtering conditions. | **Success** |
| 6 | Creating a booking | 1. Login to the site  2. Opens the desired service  3. Clicks to create a booking  4. On a new page, enter the desired information.  5. Save | A new booking will appear. | A new booking will appear. | **Success** |