|  |  |  |  |
| --- | --- | --- | --- |
| Assignment 1 | | Project Summary | |
| Course | | **Web Application & Service Development with Spring Framework v5** **- 2018** | |
|  | | | |
| Project author | | | |
| № | Pseudonym | | Face-to-face/ online |
| 1 | Teodora Bancheva | | face-to-face |
| 2 | Hristo Iliev | |  |
| 3 | Kaloyan Simitchiyski | |  |
| 4 | Simeon Gerginov | |  |

|  |  |
| --- | --- |
| Project name | ClubHub |

|  |
| --- |
| 1. Short project description (Business needs and system features) |
| The ***Online Club Reservation System (ClubHub)*** provides the ability for students to book a table at a particular club or event, view their clubbing history, get discounts and rate clubs. In addition to that it allows users to register, and club managers to add events and handle bookings. The system will be developed using ***Spring 5 Application Development Framework***.It will implement web-based front-end client using ***Thymeleaf*** templates and ***jQuery*** or ***Angular*** JavaScript/TypeScript libraries for asynchronous page data updates. Each page will have a distinct URL, and the routing between pages will be done server side using ***SpringMVC***. The backend will be implemented as a ***REST/JSON API*** using JSON data serialization.  The main user roles (actors in UML) are:  • *Anonymous User* – can only view the information pages about *clubs* and *events*.  • *Client* – can make bookings for club events, rate clubs, get discounts.  • *Club* – can manage (create, edit and delete) club information, as well as events. |

|  |  |  |
| --- | --- | --- |
| 1. Main Use Cases / Scenarios | | |
| **Use case name** | **Brief Descriptions** | **Actors Involved** |
| * 1. **Browse information** | The *User* can browse information about clubs and upcoming events. | All users |
| * 1. **Register** | *Anonymous User* can register in the system by providing a valid e-mail address, first and last name, and choosing password. By default, all new registered users have *Client* role.  *Anonymous User* can also register in the system as a *Club* by providing information about the club. | *Anonymous User* |
| * 1. **Make booking** | *Clients* can make bookings for upcoming club events specifying a particular number of people, choosing upfront the drinks for the table. | *Client* |
| * 1. **Rate club** | *Clients* can give clubs *ratings* and leave comments. | *Client* |
| * 1. **Create event** | *Clubs* can create various *events*, featuring *Singers* and *DJ-s*. | *Club* |
| * 1. **View history** | *Clients* can view the history of their bookings. | *Client* |
| * 1. **Get Discount** | *Clients* get points for every booking they make and when they have enough points they get a certain discount.  *Clubs* can determine for how many points and what percentage the discounts will be. | *Client, Club* |
| * 1. **Leaderboards** | Clients can view the current or past leaderboards.  Clubs can create new leaderboards. | Client. Club |
| * 1. **Menu** | Clients can view the menu of each club.  Clubs can add/remove/change the menu items. | Clients, Club |
| * 1. **Subscription** | Clients can create subscription for a particular club and receive notifications about incoming events. They can also delete/update their existing subscription or view all subscription that they’ve ceated. | Client |
| * + 1. **Notifications** | Client can receive notifications about incoming events to a notification channel of their choice. Notifications are defined by subscriptions. | Clients, Clubs |
| * + 1. **Book a parking slot** | Clients can book a parking slot for a given club in advance. They can view their current bookings, as well as their past ones.  Clubs can define the amount of parkings slots that can be booked. | Clients, Clubs |
| * + 1. **Chat with the club** | Clients can ask their question about a club directly to them via chat. | Clients, Clubs |

|  |  |  |
| --- | --- | --- |
| 1. Main Views (Frontend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Home** | Presents the introductory information for the purpose of the system as well as detailed instructions how to start using it. Prominently offers ability to register. | / |
| * 1. **Login** | Presents a view allowing the users to login. | */login* |
| * 1. **User history** | Presents a view with the user’s clubbing history. | */history* |
| * 1. **Events** | Presentsinformation about events. | */clubs/event* |
| * 1. **Booking** | Presents a view allowing the users to make a booking. | */clubs/booking* |
| * 1. **Rating** | Presents a view allowing the users to rate a club. | */clubs/rating* |
| * 1. **Leaderboard** | Presents a view allowing the users to view the leaderboard | */clubs*/leaderboards |
| * 1. **Menu** | Presents information about the menu of a club | */clubs/menu* |
| * 1. **Subscription** | Presents information about the subscription for a club | */clubs*/subscriptions |
| * 1. **Notifications** | Presents information about all received notifications | */clubs/notifications* |
| * 1. **Booking parking slot** | Presents a view allowing the user to make a booking for a parking slot | */clubs/parking-slot-bookings* |
| * 1. **Chat with the club** | Presents a view allowing the user to chat with a club or check their chat history | */clubs/chat* |

|  |  |  |
| --- | --- | --- |
| 1. API Resources (Backend) | | |
| **View name** | **Brief Descriptions** | **URI** |
| * 1. **Users** | GET *User Data* for a specific users, and POST new *User Data*. Available only for *Administrators*. | */api/users/{userID}* |
| * 1. **Login** | GET *User Credentials* (password). | */api/login* |
| * 1. **Sign in** | POST *User Credentials* (password) if the user is a normal client.  POST *Club Credentials* if the user is a club manager. | */api/signin* |
| * 1. **Logout** | POST a logout request. | */api/logout* |
| * 1. **Client History** | GET *User* history for a specific user according to the data stored in Bookings. | */api/users/{userID}/history* |
| * 1. **View Club Information** | GET, PUT, DELETE *ClubInfo* with specified *clubID*. | */api/clubs/{clubID}* |
| * 1. **Add Event** | POST *Event* to a club with specified *clubID*. | */api/clubs/{clubID}/event* |
| * 1. **View Event Information** | GET *Event* information if you are a client.  PUT or DELETE *Event* with specified *eventID* if you are a club manager. | */api/events/{eventID}* |
| * 1. **Make Booking** | POST *Booking* for a particular club with *clubID* or event with *eventID*. | */api/clubs/{clubID}/booking*  */api/events/{eventID}/booking* |
| * 1. **View Booking Information** | GET information about a *Booking* with a specified *bookingID.* | */api/clubs/{clubID}/bookings/{bookingID}* |
| * 1. **Rate a club** | POST *Rating* for a club with specified *clubID*. | */api/clubs/{clubID}/ratings* |
| * 1. **Modify club’s menu** | POST, PUT, DELETE information about menu for a club with specified *clubID*. | */api/clubs/}{clubID}/drinks* |
| * 1. **View club menu** | GET information about menu for a club with specified *clubID*. | */api/clubs/}{clubID}/drinks* |
| * 1. **View leaderboards** | GET information about the current leaderboard for a club with specified clubID  GET information about specific leaderboard of a club  GET information about past leaderboards for a club with specified clubID | /api/clubs/{clubID}/leaderboards/current  /api/clubs/{clubID}/leaderboards/{leaderboardID}  */api/clubs/{clubID}/leaderboards?start=<START\_DATE>&end=<END\_DATE>* |
| * 1. **Modify leaderboards** | POST – create new leaderboard for a club with specified clubID  PUT – modfy existing leaderboard for a club with specified clubID  DELETE – delete existing leaderboard of a club with specified clubID | */api/clubs/{clubID}/leaderboards*  */api/clubs/{clubID}/leaderboards/{id}*  */api/clubs/{clubID}/leaderboards/{id}* |
| * + 1. **View menu** | GET information about the menu of a club with specified clubID | */api/clubs/{clubID}/menu* |
| * + 1. **Modify menu** | POST – create new menu for a club with specified club id  PUT – modify existing menu for a club with specified clubID  DELETE – remove menu for a club with specified clubID | */api/clubs/{clubID}/menu*  */api/clubs/{clubID}/menu/{menuID}*  */api/clubs/{clubID}/menu/{menuID}* |
| * + 1. **View subscription** | GET information about subscriptions created for a club with specified clubID | */api/clubs/{clubID}/subscriptions* |
| * + 1. **Modify subscriptions** | POST – create new subscription for a club with specified clubID  PUT – modfy existing subscription for a club with specified clubID  DELETE – remove existing subscription created for a club with specified clubID | */api/clubs/{clubID}/subscriptions*  */api/clubs/{clubID}/subscriptions/{subID}*  */api/clubs/{clubID}/subscriptions/{subID]* |
| * + 1. **View received notifications** | GET information about received notifications from a club with specified clubID | */api/clubs/{clubID}/notifications* |
| * + 1. **View parking slot bookings** | GET information about parking slot bookings for a club with specified clubID  GET information about a specific parking slot booking for a club with specified clubID | */api/clubs/{clubID}/parkings-slot-bookings*  */api/clubs/{clubID}/parkings-slot-bookings/{bookingID}* |
| * + 1. **Modify parking slot bookings** | POST – create new parking slot bookings for a club with specified clubID  DELETE – remove a given parking slot booking for a club with specified clubID  PUT – modify existing parking slot booking for a club with specified clubID | */api/clubs/{clubID}/parkings-slot-bookings*  */api/clubs/{clubID}/parkings-slot-bookings/{bookingID}*  */api/clubs/{clubID}/parkings-slot-bookings/{bookingID}* |