**REQUIREMENTS ENGINEERING**

**Use Cases**

**BBQReserver**

Members of Group 8:

**Arman Kialbekov**

**Kirill Kuyan**

**Batyrzhan Zhetpisbaev**

Glossary

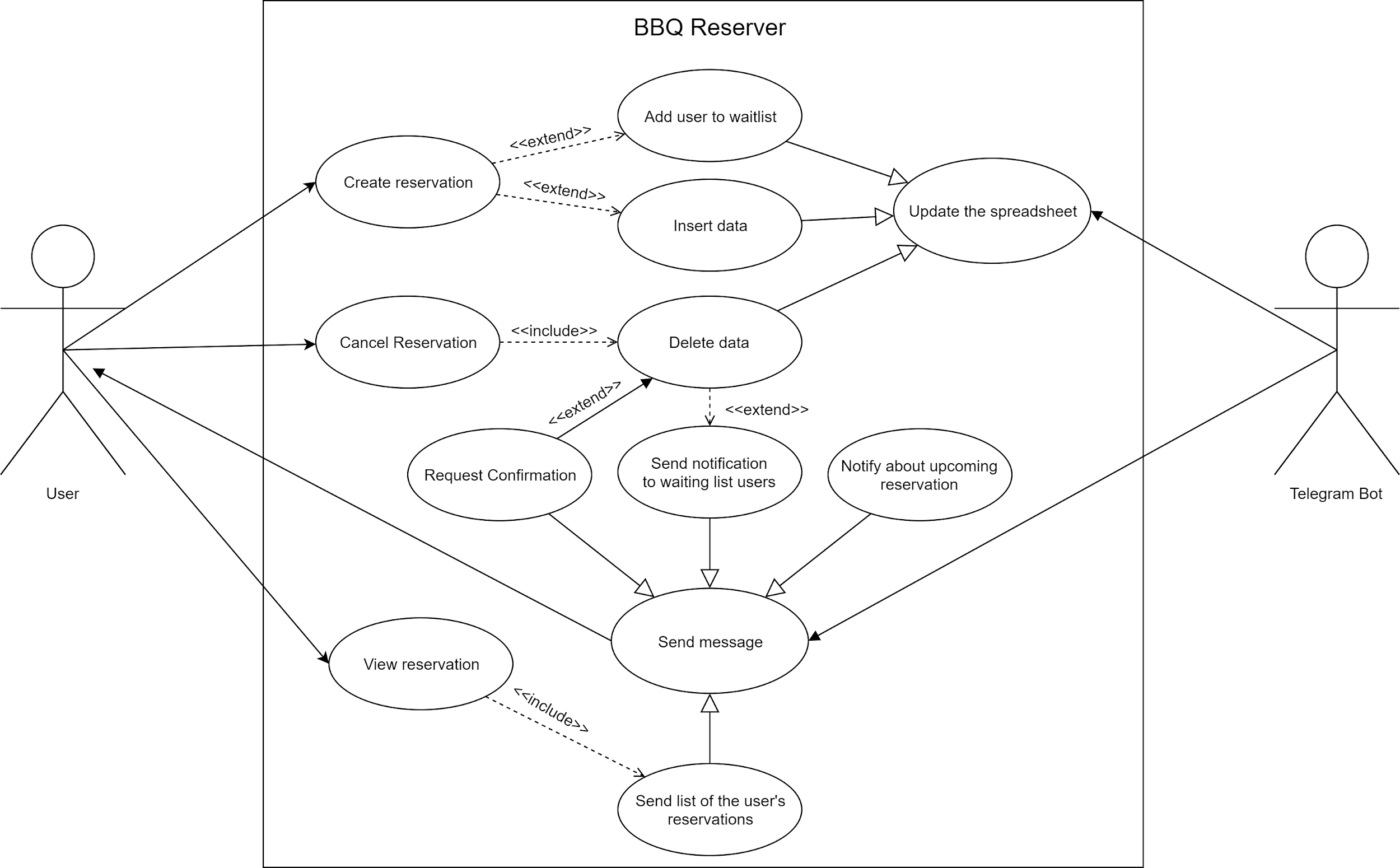
**Time slot** - the time unit for reservation a BBQ zone. In the period 8 AM - 6 PM equals *two hours*. In the period 6 PM - 12 AM equals *three hours*.

**Database** - SQL database

**Telegram ID** - unique identifier of a user in Telegram messenger used by the Bot

**Waiting list** - list of users who wants to reserve time slots that already booked but, can be available if owners of the reservation will cancel the reservation.

BBQ Reserver Use Diagram



User use cases

|  |  |
| --- | --- |
| **Use Case Name:** | Create reservation |
| **Use Case ID:** | NR-10 |
| **Primary Actors:** | Telegram user |
| **Secondary Actors:** | Telegram bot |
| **Brief description:** | The user creates reservation by selecting an available date and time. |
| **Preconditions:** | The user already has:   * less than 1 reservation for the selected day * less than 3 reservation for the selected week * less than 12 reservation for the selected month |
| **Flow of events:** | 1. The user selects the option to create a reservation. 2. The bot returns a list of available dates:  * today * tomorrow * the day after tomorrow.  1. The user selects the date. 2. The bot returns the list of time slots for the selected date. |
| **Postconditions:** | The bot provides the list of time slots for the selected date and option to be added to the waitlist. |
| **Priority:** | High |
| **Alternative flows**  **and exceptions:** | The user cannot create a reservation if he has reached the limit for the maximal number of reservations. |
| **Assumptions:** | A user started the chat with the Telegram bot. |

|  |  |
| --- | --- |
| **Use Case Name:** | Cancel reservation |
| **Use Case ID:** | CR-10 |
| **Primary Actors:** | Telegram user |
| **Secondary Actors:** | Telegram bot |
| **Brief description:** | A user delete own existing reservation |
| **Preconditions:** | User’s reservation must be existed |
| **Flow of events:** | 1. User selects option cancel reservation 2. User receives list of his reservations 3. User selects reservation for cancelation 4. Include DD-10 |
| **Postconditions:** | Selected reservation time slot |
| **Priority:** | High |
| **Alternative flows**  **and exceptions:** | * If user has not any active reservations, user will be provided with message about absence of reservations, and cancelation of chosen user’s action * If user select option by accident, user can leave option with back button |

|  |  |
| --- | --- |
| **Use case name:** | View reservation |
| **ID:** | VR-10 |
| **Primary Actors:** | Telegram user |
| **Secondary Actors:** | Telegram bot |
| **Brief description:** | A user requests the bot to provide his/her list of existing reservations |
| **Precondition:** | At least one existing reservation for a user who initiate the command in upcoming dates |
| **Flow of events:** | 1. A user presses button “View my reservations”  2. The bot reads data of the user by telegram id in the database  3. Include SL-10 |
| **Postcondition:** | List of user’s reservations |
| **Priority:** | High |
| **Alternative flows**  **and exceptions:** | If the user does not have any reservations the bot must send message: “You have not made any reservations yet” |

Bot use cases

Update Database Specification

|  |  |
| --- | --- |
| **Use case name:** | Update the database (abstract use case) |
| **Use case ID:** | NR-20 |
| **Brief description:** | The bot makes updates in the database. |
| **Flow of events:** | 1. The bot identifies user id. 2. The bot makes changes in the corresponding row for the user. |
| **Postconditions:** | The data is updated in the database for the user. |
| **Priority:** | High |
| **Nonbehavioral**  **requirements:** | The bot should have constant connection to the Internet |
| **Assumptions:** | The bot has read and write access to the database |
| **Issues:** | The bot can have problems with accessing the database |

|  |  |
| --- | --- |
| **Extending use case name:** | Add the user to the waitlist |
| **Extending use case ID:** | NR-21 |
| **Description:** | The user is added to the waitlist to be notified if the time slot becomes free. |
| **Parent use case name:** | Update the database (abstract use case) |
| **Parent use case ID:** | NR-20 |
| **Extended use case name:** | Create reservation |
| **Extension point:** | Add user to the waitlist. |
| **Guard condition (precondition):** | The user selects an option to be added to the waitlist. |
| **Flow of events:** | 1. The bot adds the user to the waitlist table in the database (S - Activity 2, NR-20) |
| **Postconditions:** | The user is added to the waitlist. |
| **Priority:** | Medium |

|  |  |
| --- | --- |
| **Extending use case name:** | Insert data |
| **Extending ID Number:** | NR-22 |
| **Brief description:** | The selected time slot is reserved by the user. The bot inserts data about the new reservation to the database. |
| **Parent use case name:** | Update the database (abstract use case) |
| **Parent use case ID:** | NR-20 |
| **Extended use case name:** | Create reservation |
| **Extension point:** | Create a reservation for a free time slot. |
| **Guard condition (precondition):** | The user selects a free time slot. |
| **Flow of events:** | 1. The bot inserts the selected date, time and the user’s telegram ID to the reservations table in the database. (S - Activity 2, NR-20) |
| **Postconditions:** | The data about reservations is inserted in the database. |
| **Priority:** | High |
| **Alternative flows**  **and exceptions:** | The user may not respond in the appropriate time and the chosen time slot can be reserved, in this case user will receive message “This reservation has already been booked” |
| **Issues:** | Two users can select in the same time slots at the same time. This causes the problem with concurrent data. |

|  |  |
| --- | --- |
| **Use Case Name:** | Delete Data |
| **Use Case ID:** | DD-10 |
| **Primary Actors:** | Telegram Bot |
| **Brief description:** | Telegram bot deletes a row of reservation from the database if one exists |
| **Parent use case name:** | Update the database (abstract use case) |
| **Parent use case ID:** | NR-20 |
| **Preconditions:** | Selected reservation |
| **Flow of events:** | 1. Search reservation in database (S - Activity 2, NR-20) 2. Delete the reservation (S - Activity 2, NR-20) |
| **Postconditions:** | Deleted reservation row |
| **Priority:** | High |
| **Alternative flows**  **and exceptions:** | * If user has not any active reservations, user will be provided with message about absence of reservations, and cancelation of chosen user’s action |

Send Message Specification

|  |  |
| --- | --- |
| **Use case name:** | Send message (abstract use case) |
| **Use case ID:** | SM-10 |
| **Brief description:** | The bot makes updates in the database. |
| **Flow of events:** | 1. The bot finds the user. 2. The bot sends a message. |
| **Postconditions:** | The message is sent to the user. |
| **Priority:** | High |
| **Issues:** | The bot cannot send message to the user if he closes the chat. |

|  |  |
| --- | --- |
| **Extending use case name:** | Send Notification to waiting list users |
| **Extending ID Number:** | SN-20 |
| **Brief description:** | The bot checks if the waiting list has any user for sending notifications after the reservation was cancelled. If there is at least one user in the list - the bot sends notification |
| **Parent use case name:** | Send message (abstract use case) |
| **Parent use case ID:** | SM-10 |
| **Extension point:** | Deleted reservation row |
| **Guard condition (precondition):** | There is at least one user in waiting list on the reservation date |
| **Flow of events:** | 1. Bot extracts telegram ids from waiting list (S - Activity 1, SM-10) 2. Bot sends notifications for extracted ids (S - Activity 2, SM-10) |
| **Postconditions:** | Notifications are sent to the users |
| **Priority:** | High |

|  |  |
| --- | --- |
| **Use case name:** | Send list of the user’s reservations |
| **Use Case ID:** | SL-10 |
| **Primary Actors:** | Telegram bot |
| **Secondary Actors:** | Telegram user |
| **Brief description:** | The bot sends message with information about user’s reservations |
| **Parent use case name:** | Send message (abstract use case) |
| **Parent use case ID:** | SM-10 |
| **Flow of events:** | 1. Bot transforms data of the list into a message according to a template (S - Activity 2, SM-10)  2. Bot sends the message to the user (S - Activity 2, SM-10) |
| **Postcondition:** | Message with the user’s reservations |
| **Priority:** | Medium |

|  |  |
| --- | --- |
| **Use case name:** | Request confirmation |
| **Use Case ID:** | RC-10 |
| **Primary Actors:** | Telegram bot |
| **Secondary Actors:** | Telegram user |
| **Brief description:** | The bot sends requests to a user to confirm upcoming reservation |
| **Parent use case name:** | Send message (abstract use case) |
| **Parent use case ID:** | SM-10 |
| **Flow of events:** | 1. Bot finds user’s telegram id by a reservation (S - Activity 1, SM-10)  2. Bot sends a message to the user with the reservation description and two options to choose: “Confirm” and “Cancel” (S - Activity 2, SM-10) |
| **Postcondition:** | The user’s choice: “Confirm” or “Cancel” |
| **Priority:** | High |
| **Alternative flows**  **and exceptions:** | - If the user chooses “Confirm”, the bot will send a message to user with the text “Your reservation was confirmed” with no data modification  - If the user chooses “Cancel” - go to the extending use case DD-10  - The user may respond not on time, for example when the reservation has already expired, in this case the bot sends message “This reservation has already expired” |
| **Assumptions:** | - The user may not be able to respond to the bot request, in this the bot will do nothing |