|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description of change |
| 01 | 29/10/2022 | Nuray Seker | Create Use Case |
|  |  |  |  |
|  |  |  |  |

|  |
| --- |
| Header |

## 

|  |  |
| --- | --- |
| Use Case: | Create Rooms (id, size, available doors, windows, lights and fans) |
| ID: | UC-002 |
| Description: | The user can create a room by entering imputs such as id, room size, the number of available doors, the number of available windows, lights, and fans. |
| Actors: | User, Admin |
| Stakeholders and Interests: | User, Admin |
| Trigger | Clicking on the context menu field “new room”, providing the provided input that is required and clicking on “OK”. |

|  |
| --- |
| **Pre-Conditions** |

The database doesn’t include the room that is about to be added.

The user knows the information that is required to create a new room.

|  |
| --- |
| **Post-Conditions** |

The database has one more room.

|  |
| --- |
| **Success end condition** |

Room is added to the database and can be accessed through the application.

|  |
| --- |
| **Failure end condition:** |

No room added.

|  |
| --- |
| **Main Success Scenario:** |

1. User opens application
2. User selects context menu field “New User”
3. User is required to give numeric input.
   1. ID
   2. Room Size
   3. Number of doors
   4. Number of windows
   5. Number of lights
   6. Number of fans
4. User clicks on “OK” button.
5. User is added.

|  |
| --- |
| **Alternative Flow and Exceptions:** |

No alternative flow.

Exceptions

User tries to enter wrong input such as size 0 and 0 doors.

User clicks on “Cancel” button.

|  |
| --- |
| **Non-Functional Requirements:** |

Performance

Adding rooms should not take longer than 5 seconds.

Security

Provided input must not leak outside of the database.

User Interface

Adding users must be easily done without using a user manual.

Consistency

The application must make sure to commit transaction correctly, not allowing data inconsistencies.