



**Table of Contents**

[**Goal 3**](#_gjdgxs)

[**Vision statement: 3**](#_30j0zll)

[**User Stories 3**](#_1fob9te)

[**Requirements 4**](#_3znysh7)

[**Tasks 4**](#_tyjcwt)

## Goal

## Vision statement:

Our reservation system aims to provide the fastest and most efficient service to customers and hotel staff with the help of a user-friendly environment.

Overall Goal:

The system we are developing aims to aid customers to do simple tasks in a hotel reservation environment. The simple tasks include booking a room with desired qualities (quality of room, number of persons, location, date etc.), updating reservations (editing and cancelling), using coupons, making payment for the selected choice and searching rooms with desired qualities.

Strategy:

We used agile model when doing this project so working implementation and constant updated results were more important for our team than documentation. We met regularly and divided the work equally depending on each team member’s strength which provided us with a time efficient result.

## User Stories









## Requirements

|  |  |
| --- | --- |
| Functional Requirements | Non-functional Requirements |
| The users should be able to make a reservation. | User guide should be found on the bottom of the website. |
| The users should be able to cancel a reservation. | The hotel reservation should be able to work on web. |
| The system should be able to apply coupons if the customer used the hotel over 5 times or booked more than 10 rooms. | The system should be able to prevent advertisements. |
| The users should be able to make accurate payment through the system. | Our system prototype uses mock-up data but the product uses MySQL as database management system. |
| The users should be able to search hotel rooms with desired qualities, location and date. | The database backups the information every day. |
| The users should be able to edit their reservations according to hotel regulations. | Java Script is the language used to implement the hotel reservation system. Angular 2 is used as web framework it is written by type script and type script is the wrapper of java script. |
| The users should be able to register with their e-mail, user id, password, telephone number information. | No limitation of code length is present; any efficient working code is acceptable. |
| The users should be able to view and check their reservations. | The budget for the product is estimated as 10000$ |
| The users should be able to successfully login the system with user id and password. | If the system stops working due to a disconnection etc. the data entered is not saved. |
| The users should be able to logout the system. | The product will be licensed every year. |
| The hotel staff should be able to update check in and check out status of the room. | The system will be active 24 hours/ 7 days. |
| The hotel staff should be able to update room information for desired room. | The system will work on Internet Explorer ≥9, Firefox ≥16, Chrome ≥22, Safari ≥5.1 |
| The users should receive SNS messages after booking confirmation, cancellation or any update in their booking. | The password contains at least 8 characters and at most 15 characters. |

## 

## Tasks

Tasks are connected to user stories.

User Story 1: (Related requirement ID: User Story 3)

1. Design database
2. Designing User Interface
3. Implement Log in function or register function
4. Implement coupon function
5. Negotiate and connect with the bank
6. Implement the method for sending result to the customer.
7. Test the function.

User Story 2: (Related requirement ID: User Story 1)

1. Implement search method for existing reservations.
2. Implement the cancel and edit page.
3. Implement edit reservation and cancel reservation functionality.
4. Testing

User Story 3:

1. Design database
2. Designing User Interface
3. Implement Search room function.
4. Test the function.

User Story 4:

1. Design database
2. Design User Interface
3. Implement login and register function.
4. Implement modifying method for hotel staff.
5. Testing

User Story 5: (Related requirement ID: User Story 1)

1. Design database
2. Design User Interface
3. Implement check in/out update function
4. Testing

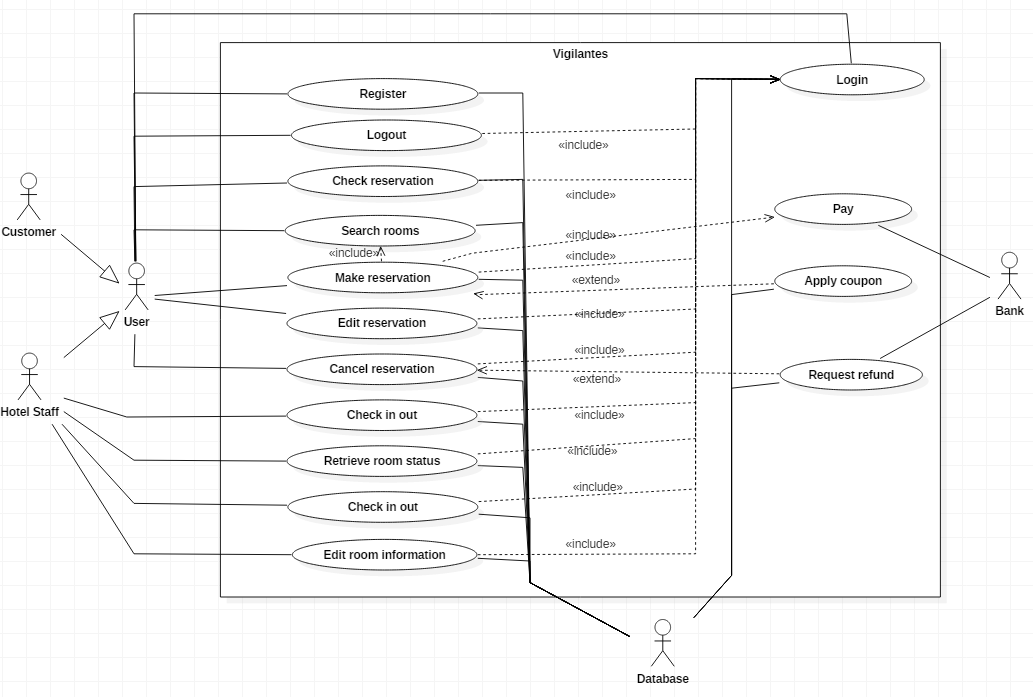
User Story 6: (Related requirement ID: User Story 1 and 3)

1. Implement multiple room booking
2. Testing

## System Design

### System Architecture

### Use case diagram and description



|  |  |  |
| --- | --- | --- |
| **Use case name** | Cancel Reservation | |
| Related requirements | The system is expected to cancel a reservation for the selected customer by the user or hotel staff. | |
| Goal in context | The use-case aims to cancel reservation at a given time. | |
| Preconditions | An existing reservation. | |
| Success End Condition | The reservation is cancelled. | |
| Failed End Condition | The reservation cancellation request is rejected. | |
| Primary Actors | User | |
| Secondary Actors | Database | |
| Trigger | The hotel staff or customer asks the system to make a reservation. | |
| Included Cases | Login, Check reservation | |
| Extended Cases | Apply Refund | |
| Main Flow | **Step** | **Action** |
| 1  2  3  4  5  6 | The user logins  Include:: Login  The user searches for existing reservations.  Include:: Search reservation  The system query the reservation from the database  The user selects reservation.  The user confirm the cancellation  The system checks the database if users need to make payment or cancel coupon  The system remove reservation from database |
| Extensions | **Step** | **Branching Action** |
| 2.1  5.1  5.2 | User does not have any existing reservation  Cancelling applied coupon if exists and update in database  The user receive refund if necessary  Include:: Refund |

|  |  |  |
| --- | --- | --- |
| **Use case name** | Make reservation | |
| Related requirements | The system is expected to make a reservation for the selected rooms by the customer or hotel staff. | |
| Goal in context | The use-case aims to make reservation in a given time within a specified room type. It is one of the vital use-cases. | |
| Preconditions | Room is available. | |
| Success End Condition | The room is booked. | |
| Failed End Condition | The reservation request is rejected. | |
| Primary Actors | User | |
| Secondary Actors | Database | |
| Trigger | The hotel staff or customer asks the system to make a reservation. | |
| Included Cases | Login, Pay, Search rooms | |
| Extended Cases | Apply coupon | |
| Main Flow | **Step** | **Action** |
| 1  2  3  4  5 | User logins  Include:: Login  User searches rooms  Include:: Search rooms  User select room to reserve  The system checks for available coupons for this user from the database  The user makes payment  Included:: Pay |
| Extensions | **Step** | **Branching Action** |
| 4.1 | There are some coupons and user want to apply one of the coupon  Extend:: Apply coupon |

|  |  |  |
| --- | --- | --- |
| **Use case name** | Apply coupon | |
| Related requirements | The system is expected to apply coupon discount for selected reservation. | |
| Goal in context | The use-case aims to apply coupon discount for selected reservation. | |
| Preconditions | User previously had 10 rooms or 5 reservations booked. | |
| Success End Condition | Coupon is successfully applied and discount 20% of the cost. | |
| Failed End Condition | Applying coupon failed | |
| Primary Actors | Customer | |
| Secondary Actors | Database | |
| Trigger | The hotel staff or customer asks the system to apply coupon. | |
| Main Flow | **Step** | **Action** |
| 1  2 | User select coupon to apply  The system check the validity of the coupon from the database |
| Extensions | **Step** | **Branching Action** |
| 2.1 | If the coupon is valid, reduce 20% from the total price if the coupon is valid. |

|  |  |  |
| --- | --- | --- |
| **Use case name** | Pay | |
| Related requirements | The system is expected to receive payment for selected reservation. | |
| Goal in context | The use-case aims to make receive payment from customer. | |
| Preconditions | Existent account, existent current reservation, accurate card information. | |
| Success End Condition | Payment is successful. | |
| Failed End Condition | Payment is rejected. | |
| Primary Actors | Customer | |
| Secondary Actors | Bank | |
| Trigger | The customer asks the system to receive payment. | |
| Main Flow | **Step** | **Action** |
| 1  2 | The customer inputs his/her card information  The system forward the inputted card information with the payment amount to the Bank |
| Extensions | **Step** | **Branching Action** |
| - |  |

|  |  |  |
| --- | --- | --- |
| **Use case name** | **Login** | |
| Related requirements | The system shall allow the users to log in to it. | |
| Goal in context | The user can log in to the service. | |
| Preconditions | The user should register their own account. | |
| Success End Condition | The user is logged in to the service successfully. | |
| Failed End Condition | The login request is rejected. | |
| Primary Actors | User | |
| Secondary Actors | Database | |
| Trigger | The user asks to log in to the service. | |
| Main Flow | **Step** | **Action** |
| 1  2 | The user enters his/her username and password.  The system verify username and password with the database |
| Extensions | **Step** | **Branching Action** |
| - | - |

|  |  |  |
| --- | --- | --- |
| **Use case name** | **Logout** | |
| Related requirements | The system shall make the user logout from the service. | |
| Goal in context | The user logged out from the service. | |
| Preconditions | The user needs to be logged in. | |
| Success End Condition | The user is logged out from the service successfully. | |
| Failed End Condition | - | |
| Primary Actors | User | |
| Secondary Actors | - | |
| Trigger | The user asks the system to logout from it. | |
| Included Cases | Login | |
| Main Flow | **Step** | **Action** |
| 1  2 | The user logins  Include:: Login  The user asks to log out. |
| Extensions | **Step** | **Branching Action** |
| - | - |

|  |  |  |
| --- | --- | --- |
| **Use case name** | **Check reservation** | |
| Related requirements | The system shall allow the user to check his/her reservation information. | |
| Goal in context | The user checks his/her reservation information. | |
| Preconditions | The user needs to have reserved some rooms. | |
| Success End Condition | - | |
| Failed End Condition | Checking request is rejected. | |
| Primary Actors | User | |
| Secondary Actors | Database | |
| Trigger | The customer asks the system to show his/her reservation information. | |
| Included Cases | Login | |
| Main Flow | **Step** | **Action** |
| 1  2  3 | The customer logins.  Included:: Login  The customer asks the system to check his/her reservation.  The system query the user reservation from Database |
| Extensions | **Step** | **Branching Action** |
| - |  |

|  |  |  |
| --- | --- | --- |
| **Use case name** | **Request refund** | |
| Related requirements | The system shall refund the customer when the customer cancels their reservation | |
| Goal in context | Refund to customer bank account | |
| Preconditions | The refunding is necessary according to the system | |
| Success End Condition | Refund success | |
| Failed End Condition | Refunding is rejected | |
| Primary Actors | Customer | |
| Secondary Actors | Bank, Database | |
| Trigger | The customer request the system for canceling their reservation | |
| Main Flow | **Step** | **Action** |
| 1  2  3 | The customer request the system for refunding  The system checks the transaction ID of the reservation from the Database  The system request the Bank for refunding |
| Extensions | **Step** | **Branching Action** |
| - | - |

|  |  |  |
| --- | --- | --- |
| **Use case name** | Search rooms | |
| Related requirements | It is supposed to find the specific room that the user requires. | |
| Goal in context | It’s main goal is to search for the desired available room in Database, and give back to the user | |
| Preconditions | Customer should know the check in/out dates | |
| Success End Condition | Display what is matched | |
| Failed End Condition | Display failure message | |
| Primary Actors | User | |
| Secondary Actors | Database | |
| Trigger | A user searches | |
| Main Flow | **Step** | **Action** |
| 1  2 | Customer inputs number of guest, desired room type, check-in and check-out date to the system  The system checks the database for any rooms satisfying the inputted condition |
| Extensions | **Step** | **Branching Action** |
| - | - |

|  |  |  |
| --- | --- | --- |
| **Use case name** | Edit Reservation | |
| Related requirements | It is supposed to change any reservation information | |
| Goal in context | The use-case aims to edit reservation before a week before its date(group decision) | |
| Preconditions | You should have been made a reservation | |
| Success End Condition | Change the information in the reservation | |
| Failed End Condition | The reservation was not changed | |
| Primary Actors | User | |
| Secondary Actors | Database | |
| Trigger | A user asks the hotel stuff to change the reservation details | |
| Included Cases | Login, Check reservation | |
| Extended Cases | Pay, Request refund | |
| Main Flow | **Step** | **Action** |
| 1  2  3  4  5  6 | The user logins  Include:: Login  The user checks for his/her reservation  Include:: Check reservation  The user select the reservation he wants to edit  The user input the information he wants to edit  The system checks with the Database if payment or refund is necessary  The system update the Database |
| Extensions | **Step** | **Branching Action** |
| 5.1  5.2 | If payment is necessary, goto Pay  Extend:: Pay  If refund is necessary, goto Refund  Extend:: Refund |

|  |  |  |
| --- | --- | --- |
| **Use case name** | Register | |
| Related requirements | It is supposed to create an account for a user | |
| Goal in context | The use-case aims to register any time | |
| Preconditions | User should have an email (Emails shouldn’t have been used before in registration) | |
| Success End Condition | The user got registered. | |
| Failed End Condition | Passwords do not match, Username was used, this email has been used in another account | |
| Primary Actors | User | |
| Secondary Actors | - | |
| Trigger | A user is asked to register | |
| Main Flow | **Step** | **Action** |
| 1  2 | User input his desired username, password, name, email address, and phone number.  The system checks with the Database if the information is correct |
| Extensions | **Step** | **Branching Action** |
| 2.1 | If the information is correct, create an account in the Database |

|  |  |  |
| --- | --- | --- |
| **Use case name** | **Edit room info** | |
| Related requirements | The system shall allow the hotel staff to make changes on room information. | |
| Goal in context | The hotel staff requests changes in room information. | |
| Preconditions | The hotel staff needs to have an appropriate proof of identity. | |
| Success End Condition | The room information is changed. | |
| Failed End Condition | The changing request is rejected | |
| Primary Actors | Hotel staff | |
| Secondary Actors | Database | |
| Included Cases | Login | |
| Trigger | The hotel staff asks the system to modify room type or price. | |
| Main Flow | **Step** | **Action** |
| 1  2  3  4  5  6  7 | The hotel staff logins  Include:: Login  The hotel staff requests the list of rooms  The system query all rooms from the Database  The hotel staff select the room he wants to edit  The system query the previous information about that room  The hotel staff enter new room information  The system saves new information to the Database |
| Extensions | **Step** | **Branching Action** |
| - |  |

|  |  |  |
| --- | --- | --- |
| **Use case name** | **Retrieve room status** | |
| Related requirements | The system shall allow hotel staff to retrieve room booking status. | |
| Goal in context | The hotel staff requests room booking status. | |
| Preconditions | The hotel staff needs to have an appropriate proof of identity. | |
| Success End Condition | Room booking status are shown | |
| Failed End Condition | The retrieve request is rejected | |
| Primary Actors | Hotel staff | |
| Secondary Actors | Database | |
| Trigger | The hotel staff asks the system to show room booking status | |
| Included Cases | Login | |
| Main Flow | **Step** | **Action** |
| 1  2 | The hotel staff logins  Include:: Login  The hotel staff requests the system to retrieve the room status |
| Extensions | **Step** | **Branching Action** |
| - | - |

|  |  |  |
| --- | --- | --- |
| **Use case name** | **Check in / out** | |
| Related requirements | The system shall allow the hotel staff to enter customer check in / out information. | |
| Goal in context | The hotel staff request to enter customer check in / out information | |
| Preconditions | The hotel staff needs to have an appropriate proof of identity. | |
| Success End Condition | Check in / out information is saved. | |
| Failed End Condition | Check-in / out is being rejected | |
| Primary Actors | Hotel staff | |
| Secondary Actors | Database | |
| Trigger | The hotel staff asks the system to show room booking status | |
| Included Cases | Login | |
| Main Flow | **Step** | **Action** |
| 1  2  3 | The hotel staff logins  Include:: Login  The hotel staff input customer phone number  The system checks the reservation and customer information in the Database |
| Extensions | **Step** | **Branching Action** |
| - | - |