# Team Roles

Project Manager -Bob

Documentation -Patrick

Business Analysist  
Requirements Engineer -Jonathan

Architect -Jonathan

Systems Analysist -Patrick

Designer -Bob

Technical Lead -Oliver

Programmers -All

Tester -All

Dev Ops -Oliver

# Hotel Management System

# Use Cases

|  |  |
| --- | --- |
| Use Case 1 | Log In |
| Actor Action | System Response |
| 1 - Staff enters username and password | 2 - Login details are checked. Main screen is displayed. |
| Alternative Route | |
| 1(a) - Staff enters incorrect login details | 2(a) - Login check fails, appropriate error displayed. |

**Non functional requirement: Security**  
User passwords must be encrypted i.e. hashed passwords.  
If manager login is detected the two tier authentication is used.

|  |  |
| --- | --- |
| Use Case 2 | Log Out |
| Actor Action | System Response |
| 1 - Staff selects log out option | 2 - Systems displays logged out screen |

|  |  |
| --- | --- |
| Use Case 3 | View All Rooms |
| Actor Action | System Response |
| 1 - Staff select view rooms option | 2 - System displays all rooms and room details e.g. booked/occupied/available |

|  |  |
| --- | --- |
| Use Case 4 | Reserve Room |
| Actor Action | System Response |
| 1 - Staff selects reserve room option | 2 - System displays reserve room form |
| 3 - Staff completes reservation form | 4 - System displays payment form |
| 5 - Staff completes payment | 6 - Room booked |
| Alternative Route | |
| 3(a) - Staff cancels reservation | 4(a) - System displays main menu |
| 5(a) - Staff cancels payment | 6(a) - System returns to reservation form |
| 5(b) - Payment rejected | 6(b) - System returns to reservation form |

**Non functional requirement: Security**  
Payment processes €1 transaction to confirm card is valid. Remainder of bill calculated and processed at checkout.

|  |  |
| --- | --- |
| Use Case 5 | Check Out |
| Actor Action | System Response |
| 1 – Staff selects check out customer | 2 – System displays check out screen with final bill. |
| 3- Staff completes payment | 4 – System prints receipt. Room set to cleaning required. |
| Alternative Route | |
| 3(a) - Payment fails | 4(a) – System returns to check out screen |
| 3(b) – Staff cancels payment | 4(b) – System prints non payment form |

|  |  |
| --- | --- |
| Use Case 6 | Clean Room Request |
| Actor Action | System Response |
| 1 – Staff selects clean room | 2 – System generates request for cleaning staff |

|  |  |
| --- | --- |
| Use Case 7 | Clean room complete |
| Actor Action | System Response |
| 1 – Cleaning staff set Clean Room Request to complete | 2 – System records staff member and time, sets room to clean. |

|  |  |
| --- | --- |
| Use Case 8 | View Room Service Orders |
| Actor Action | System Response |
| 1 – Staff selects view room service orders | 2 – System Shows all current room service orders |

|  |  |
| --- | --- |
| Use Case 9 | View room details |
| Actor Action | System Response |
| 1 – Staff selects a room | 2 – System displays room details e.g. Current bill, room service details, guest details |

|  |  |
| --- | --- |
| Use Case 10 | Cancel Reservation |
| Actor Action | System Response |
| 1 – Staff selects cancel reservation option | 2 – System displays current reservations |
| 3 – Staff selects relevant reservation | 4 – System displays reservation details and warning |
| 5 – Staff confirms cancellation | 6 – System records staff id and time, removes reservation |
| Alternative Route | |
| 5(a) – Staff does not confirm cancellation | 6(a) – System takes no action, returns to main screen |

|  |  |
| --- | --- |
| Use Case 11 | Check in |
| Actor Action | System Response |
| 1 – Staff selects check in | 2 – System displays list of pending check in’s |
| 3 – Staff selects guest | 4 – System displays details to confirm |
| 5 – Staff confirms details | 6 – System records check in. |
| Alternative Route | |
| 5(a) Staff edits details | 6(a) – System records changes and check in |

|  |  |
| --- | --- |
| Use Case 12 | Program Key |
| Actor Action | System Response |
| 1 – Staff selects program key | 2 – System displays program key menu |
| 3 – Staff selects room number to write to key | 4 – System writes data to magnetic key card |
| Alternative Route | |
| 3(a) – No key card inserted | 4(a) – System displays appropriate error |

|  |  |
| --- | --- |
| Use Case 13 | Add Staff |
| Actor Action | System Response |
| 1 – Manager selects add staff | 2 – System displays add staff form |
| 3 – Manager inputs staff details | 4 – System creates staff account |
| Alternative Route | |
| 3(a) – Manager inputs invalid/missing details | 4(a) – System highlights issues on form |

|  |  |
| --- | --- |
| Use Case 14 | Remove Staff |
| Actor Action | System Response |
| 1 – Manager selects remove staff option | 2 – System displays remove staff menu |
| 3 – Manager selects staff member and confirms | 4 – System removes staff members login facility |
| Alternative Route | |
| 3(a) – Manager does not confirm | 4(a) – System takes no action, returns to remove staff menu |

|  |  |
| --- | --- |
| Use Case 15 | View Sales |
| Actor Action | System Response |
| 1 – Manager selects view sales | 2 – System displays sales options e.g. dates |
| 3 – Manager inputs dates and other details | 4 – System displays sales report |

|  |  |
| --- | --- |
| Use Case 16 | Edit Pricing |
| Actor Action | System Response |
| 1 – Manager selects edit pricing | 2 – System displays pricing menu |
| 3 – Manager manually edits pricing options e.g. discounts, room base price | 4 – System saves changes, does not affect current bookings |

|  |  |
| --- | --- |
| Use Case 17 | Forecast report |
| Actor Action | System Response |
| 1 – Manager selects forecast report | 2 – System runs forecasting script and displays results |
| 3 – Manager views results and accepts pricing | 4 – System saves new pricing levels. Does not affect current bookings |
| Alternative Route | |
| 3(a) – Manager rejects pricing | 4(a) – System makes no changes returns to main menu. |
| 3(b) – Manager manually edits pricing | 4(b) – System saves new pricing levels. Does not affect current bookings |

|  |  |
| --- | --- |
| Use Case 18 | Online booking |
| Actor Action | System Response |
| 1 – User view hotel website | 2 – System displays availability in viewed hotel |
| 3 – User selects book | 4 – System displays booking form |
| 5 – User completes form | 6 – System displays payment screen |
| 7 – User completes payment | 8 – System processes payment, displays booking information |
| Alternative Route | |
| 5(a) Booking form not complete/incorrect | 6(a) – System highlights errors |
| 7(a) Payment fails | 8(a) – System displays appropriate error, returns to booking form |

**Non functional requirement: Security**  
Payment processes €1 transaction to confirm card is valid. Remainder of bill calculated and processed at checkout.

**Non functional requirement: Atomic Transaction**When customer selects book initially the room is immediately set to booked before form is completed. If booking is cancelled or form/payment fails the room is returned to available. Otherwise the details are saved. This ensures the same room cannot be booked by two different web clients at the same time.

|  |  |
| --- | --- |
| Use Case 19 | Online Cancellation |
| Actor Action | System Response |
| 1 – User cancels own booking | 2 – System displays booking information |
| 3 – User confirms cancellation | 4 – System removes booking information, refunds card test |

|  |  |
| --- | --- |
| Use Case 20 | Self check in |
| Actor Action | System Response |
| 1 – User selects check in at hotel kiosk | 2 – System requests name and credit card |
| 3 – User enters name and card | 4 – Displays booking information |
| 5 – User confirms booking | 6 – System writes room number to key card and provides card to customer. Room set to occupied |
| 3(a) – User name/card details incorrect | 4(a) – System refers customer to reception |
| 5(a) – User does not confirm booking | 6(a) - System refers customer to reception |
| 5(b) – Room not ready | 6(b) - System refers customer to reception, generates priority room clean request |

|  |  |
| --- | --- |
| Use Case 21 | Self check out |
| Actor Action | System Response |
| 1 – User selects check out at hotel kiosk | 2 – System requests room card and credit card |
| 3 – User enters both cards | 4 – Displays billing information |
| 5 – User confirms billing | 6 – System proceeds to payment |
| 7 – User completes payment | 8 – System prints receipt, generates clean rom request for room, sets room to available. |
| 3(a) – room/credit card incorrect | 4(a) – System refers customer to reception |
| 5(a) – User does not confirm billing | 6(a) - System refers customer to reception |
| 7(a) – Payment fails | 8(a) - System refers customer to reception |

|  |  |
| --- | --- |
| Use Case 22 | View Gym Status |
| Actor Action | System Response |
| 1 – User selects view gym status | 2 – System displays gym status |

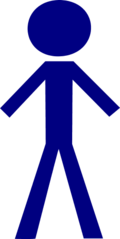
|  |  |
| --- | --- |
| Use Case 23 | View room details |
| Actor Action | System Response |
| 1 – User selects view room status | 2 – System displays room & bill details |

# Use Case Detailed Description

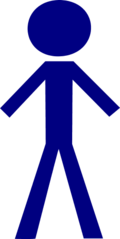
|  |  |  |
| --- | --- | --- |
| Use Case 4 | | Reserve Room |
| Goal in Context | | Room booked, customer information stored & credit card validated |
| Preconditions | | Customer books in person or on phone |
| Success End Condition | | Customer has room booked |
| Failed End Condition | | Room is not booked, customer details not kept |
| Primary, Secondary, Actors | | Customer, receptionist on behalf of customer, credit card company, bank |
| Trigger | | Customer requests specific room for specific dates |
| **DESCRIPTION** | **Step** | **Action** |
|  | 1 | Customer contacts reception |
|  | 2 | Staff selects reserve room option |
|  | 3 | Staff checks dates availability |
|  | 4 | Staff captures customer details |
|  | 5 | Staff validates credit card |
|  | 6 | Room booked |
| **EXTENSIONS** | **Step** | **Branching Action** |
|  | 2a | Room not available, staff offers other room type or other dates |
|  | 3a | Dates unavailable, staff offers other dates |
|  | 5a | Credit card fails, cannot book without valid credit card |
| **VARIATIONS** |  | **Branching Action** |
|  | 1 | Customer may phone in or contact in person |
|  | 5 | Full payment can be in advance or at checkout |

|  |  |
| --- | --- |
| **RELATED INFORMATION** | 4. Reserve Room |
| **Priority** | Top |
| **Performance** | 5 minutes for customer details/interaction  Less than 1 minute for credit card validation |
| **Frequency** | 20/day per hotel |
| **Channel to Actors** | Terminal at front desk |
| **OPEN ISSUES** | What if credit card is stolen  Power outage during transaction |
| **Due Date** | Release 1.0 |
| **…any other management information** |  |
| **Superordinates** | View All Rooms (use case 3) |
| **Subordinates** | Cancel reservation (use case 10) |
|  |  |

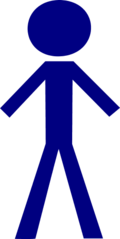
Manager



Staff



Customer



# Quality Attributes

**Security**:

Some examples of security attributes are password hashing for all stored passwords. There will be an intermediate layer between any external data i.e. online bookings and the data base. Staff and Manager login will be local network access only. Also the managers login will use 2 phase authentication, where after the initial login the manager will be prompted for additional information i.e. several digits from a six digit PIN.

**Maintainability & Extensibility:**

High encapsulation and iterative releases will help provide both maintainability and extensibility. We applied the agile methodology to this project with twice weekly meetings and using Slack messaging to aid quick communication. Good use of interfaces and coding practices will ensure that the system is easily extended.

**Performance:**

Using savepoint with rollback function for atomic transactions on the database to provide performance. This ensures that the same room cannot be booked twice and also this will prevent a customer from entering the entire booking form only for the transaction to be rejected as someone else booked the room first.

# GUI Prototypes

