

## Catalog

|  |    |
|--|----|
| <b>1. Introduce</b>                            | 4  |
| 1.1. <i>Purpose of the document</i>            | 4  |
| 1.2. <i>Document scope</i>                     | 4  |
| 1.3. <i>Definitions</i>                        | 4  |
| 1.4. <i>References</i>                         | 4  |
| 1.5. <i>Document description</i>               | 5  |
| <b>2. Software overview</b>                    | 6  |
| 2.1 <i>Software general requirements</i>       | 6  |
| 2.2 <i>Software objectives</i>                 | 6  |
| 2.3. <i>User</i>                               | 6  |
| 2.4. <i>Software overall model</i>             | 7  |
| <b>3. Analysis business process</b>            | 7  |
| 3.1. <i>Use case “Manage Room”</i>             | 9  |
| 3.1.1. <i>Use case “Add room”</i>              | 9  |
| 3.1.2. <i>Use case “Delete Room”</i>           | 10 |
| 3.1.2. <i>Use case “Update Room”</i>           | 11 |
| 3.2. <i>Use case “Manage Service”</i>          | 12 |
| 3.2.1. <i>Use case “Add Service”</i>           | 12 |
| 3.2.2. <i>Use case “Delete Service”</i>        | 13 |
| 3.2.3. <i>Use case “Update Service”</i>        | 14 |
| 3.3. <i>Use case “Change rules”</i>            | 15 |
| 3.4. <i>Use case “Turnover statistics”</i>     | 16 |
| 3.5. <i>Use case “Manage accounts”</i>         | 17 |
| 3.5.1. <i>Use case “Add account”</i>           | 17 |
| 3.5.2. <i>Use case “Delete account”</i>        | 18 |
| 3.5.3. <i>Use case “Update account”</i>        | 19 |
| 3.5.4. <i>Use case “Change security level”</i> | 20 |

|  |           |
|--|-----------|
| 3.6. Use case “Book room” .....  | 21        |
| 3.7 Use case “Order services” .....                                    | 22        |
| 3.8. Use case “Checkout” .....   | 23        |
| 3.9. Use case “Search customer information” .....                      | 25        |
| 3.10. Use case “Search Room Information” .....                         | 26        |
| <b>4. Specific requirements for functional software .....</b>          | <b>27</b> |
| 4.1. Specifications "Login" function.....                              | 27        |
| 4.2. Specifications “Create new account” function .....                | 27        |
| 4.3. Specifications “Change Password” function.....                    | 28        |
| 4.4. Specifications “Reservation” function.....                        | 29        |
| 4.5. Specifications “Check out” function.....                          | 29        |
| 4.6. Specifications “Add room” function.....                           | 30        |
| 4.7. Specifications “Delete room” function .....                       | 31        |
| 4.8. Specifications “Update room” function .....                       | 32        |
| 4.9. Specifications “Add service” function.....                        | 32        |
| 4.10. Specifications “Delete service” function .....                   | 33        |
| 4.11. Specifications “Update service” function .....                   | 34        |
| 4.12. Specifications “Order services” function.....                    | 34        |
| 4.13. Specifications “Room search” function.....                       | 35        |
| 4.14. Specifications “Search customer” function .....                  | 36        |
| 4.15. Specifications “Change regulation” function .....                | 37        |
| 4.16. Specifications "Turnover statistics by room type" function ..... | 37        |
| 4.17. Specifications “Show used service list” function .....           | 38        |
| <b>5. Other requirements.....</b>                                      | <b>40</b> |
| 5.1. Ease of Use Requirements .....                                    | 40        |
| 5.2. Stability requirements .....                                      | 40        |
| 5.3. Security requirements .....                                       | 40        |
| 5.4. Backup and restore requirements .....                             | 41        |
| 5.5. Support eligibility requirements .....                            | 41        |

5.6. *Technology requirements*..... 41

5.7. *Communication requirements*.....42

5.8. *Requires user documentation and online support* .....42

5.9. *Legal requirements* .....42

5.10. *Requirements for applicable standards* .....42

## 1. Introduce

### *1.1. Purpose of the document*

This document was developed for the purpose of analyzing, specifying the requirements for the hotel management software development project. This document describes software requirements for the entire system or for each of the subsystems used as input to the design process, the programming usecase construction, the system test of the system build.

### *1.2. Document scope*

Documentation is the result of a survey of the requirements, learning the documents related to the necessary functions of the hotel management system.

### *1.3. Definitions.*

| Configuration | Definition       | Explaining                                   |
|---------------|------------------|--|
| HTM           | Hotel Management | Name of project                              |
| DB            | Database         | Where to store information and allow access. |
| PK            | Primary Key      | Main key                                     |
| UK            | Unique Key       | Unique key                                   |
| ...           | ...              | ...  |

### *1.4. References*

The references are:

- Document about action of hotel.
- Projects related to management in the Internet.

- Experiment when hide hotel room,

### *1.5. Document description*

The structure of the specification document includes the following sections:

- **Part 1 - Introduce:** General introduction of the document, help the reader visualize the content, purpose and general layout of the document.
- **Part 2 - Overview of software:** Identify common requirements, objectives, user objects, and build the overall model of the software
- **Part 3 - Usecase process analysis:** Presentation and analysis of usecase processes of the bookstore management system module.
- **Part 4 - Functional requirements of the software:** Demonstrates the functional requirements that the software should provide.
- **Part 5 - Other requirements:** Presentation of non-functional requirements: ease of use, stability, performance, security, backup and restore capabilities, supportability, and more.

## **2. Software overview**

### *2.1 Software general requirements*

General requirements of the hotel management system software consists of two parts: room management; information management in the database.

a) Room management:

- Processing guest's booking process: check, find room for guests, changing room request.
- Enter customer information for booking.
- Payment and invoice printing for guests

b) Management of information in the database

- Management, updating information staff, services, rooms in the hotel and discount promotions (if any).
- Manage turnover statistics by day, month, quarter.
- Manage turnover statistics by room.
- Administering user access permissions by functional group: includes "Management" and "Reception". User can change password to access the system.
- Store the list of customers who have reservations in the hotel.
- Store the list of services used by customer to check invoices
- Store invoices that have been paid out over time for future reporting.

### *2.2 Software objectives*

Small and low professional level hotel and motel. Help the management to be more effectively, accurately and quickly.

### *2.3. User*

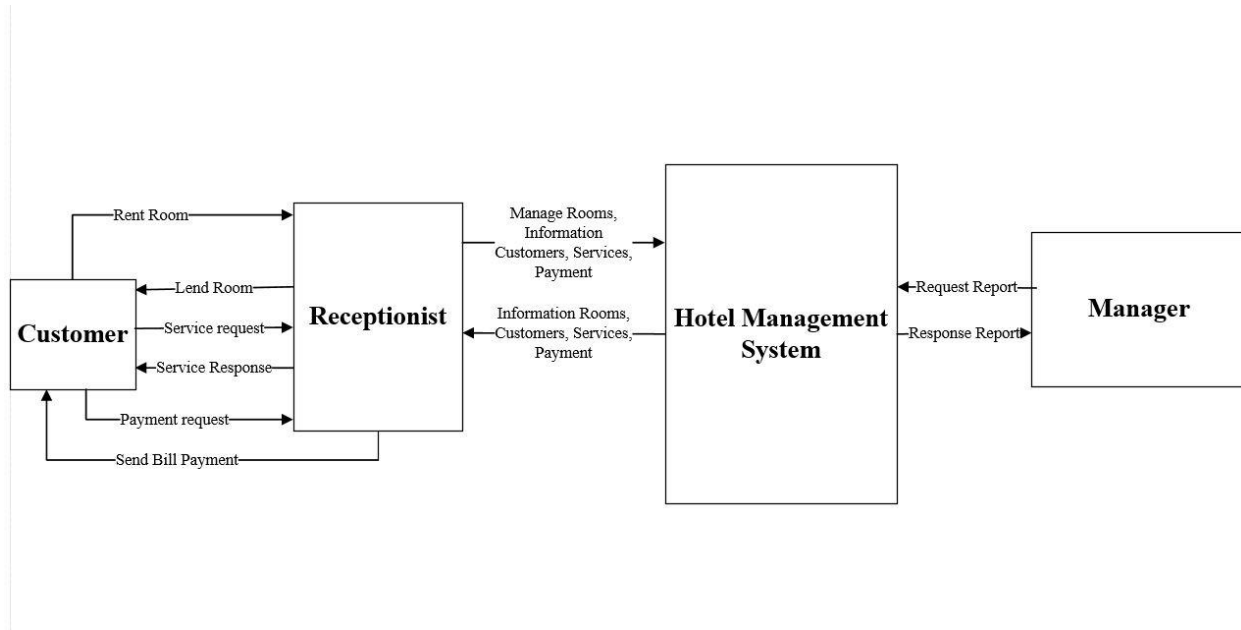
System has these user group:

- Management group: control at the whole system.
- Professional operation group:

+ Reception: check availability, enter customer information, book rooms, book services, check out and billing for guests.

+ Customer: Direct reservations or by telephone, call to reception to request service.

## 2.4. Software overall model

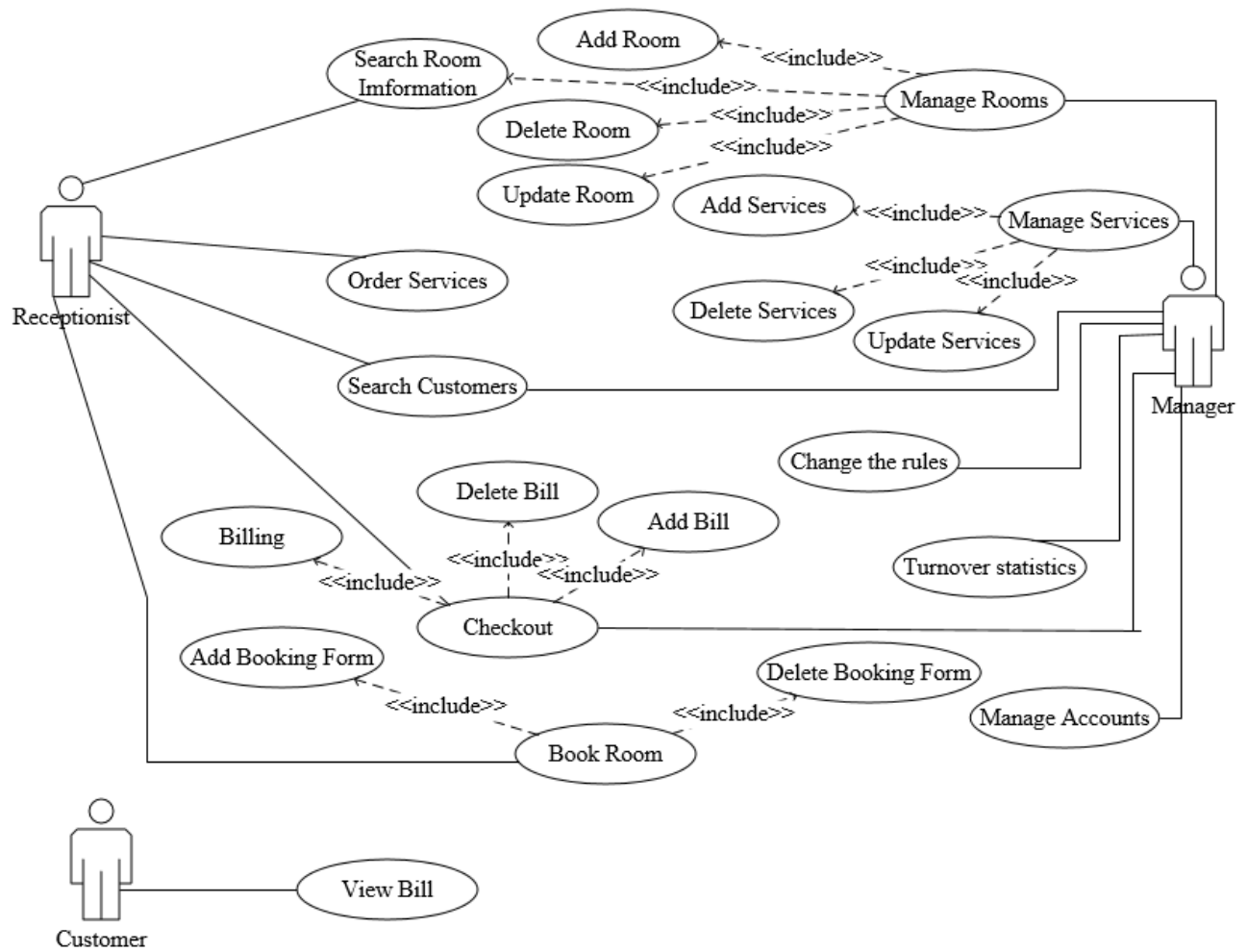


H1. Software overall model

## 3. Analysis business process

Describe in detail each specific business, allowing members of the project to build the best quality software. With the UseCase (UC) model, software requirements are described by UCs and in additional specifications.

## Use case diagram



H2. Use Case Diagram



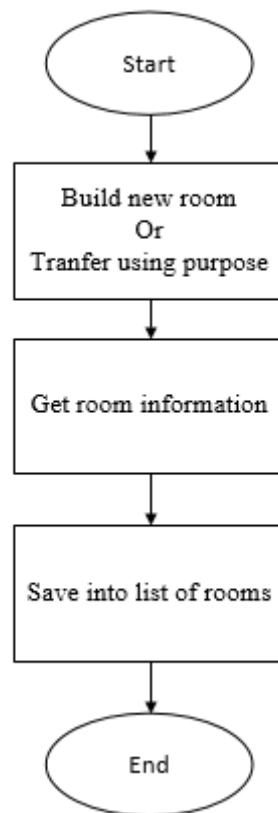
### 3.1. Use case “Manage Room”

#### 3.1.1. Use case “Add room”

##### a. Event triggered

When hotel build more rooms or tranfer using purpose of other rooms to rentor room, information of rooms must be updated.

##### b. Business process model



##### c. Describe the steps

- Step 1: Get information of room

+ Input: Room note.

+ Output: Room information is entered.

- Step 2: Update to list of rooms

+ Input: Information of room need to store.

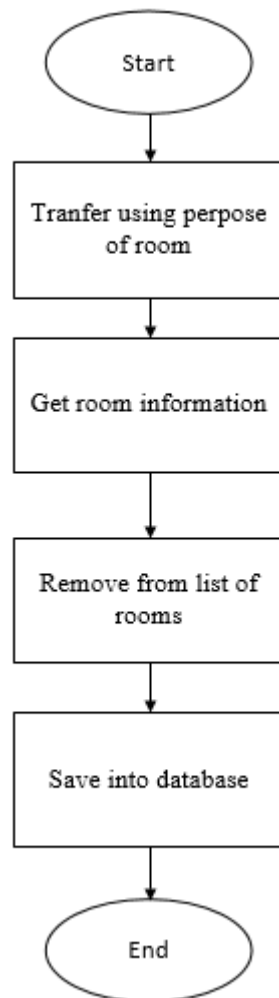
+ Output: Database store room information

### 3.1.2. Use case “Delete Room”

#### a. Event triggered

When hotel transfer using purpose of renter rooms to other purpose, database should be update.

#### b. Business process model



#### c. Describe the steps

- Step 1: Get information of room

+ Input: Choise room.

+ Output: Room information.

- Step 2: Remove room record from database:

+ Input: Room record.

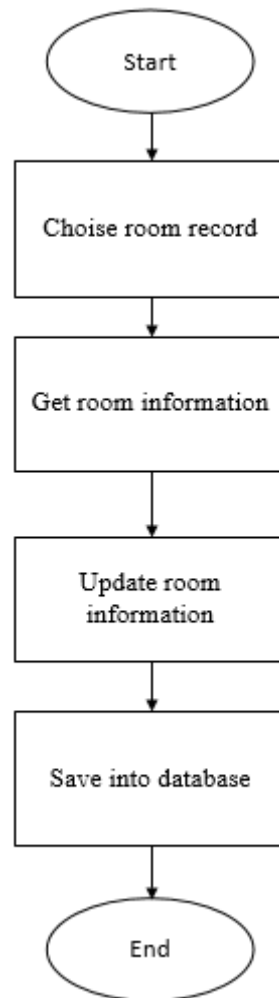
+ Output: Room record is removed from database.

### 3.1.2. Use case “Update Room”

#### a. Event triggered

When manager want change information of room, it should be updated.

#### b. Business process model



#### c. Describe the steps

- Step 1: Get information of room
  - + Input: Information from manager
  - + Output: Information of room is entered
- Step 2: Update to database
  - + Input: Room information
  - + Output: Database updated new information of room.

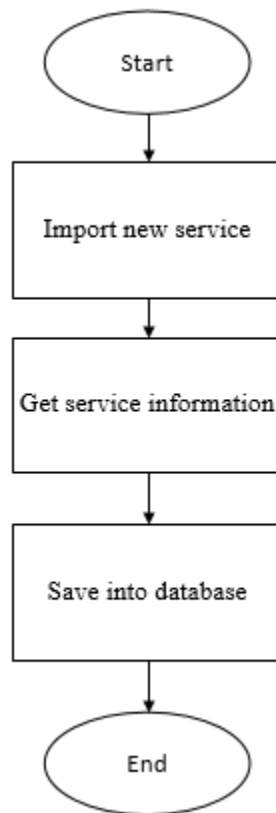
### 3.2. Use case “Manage Service”

#### 3.2.1. Use case “Add Service”

##### a. Event triggered

When manager decide to provide new service in hotel.

##### b. Business process model



##### c. Describe the steps

- Step 1: Import new service information

+ Input: Service information

+ Output: Service information is entered

- Step 2: Get new service information

+ Input: Service information

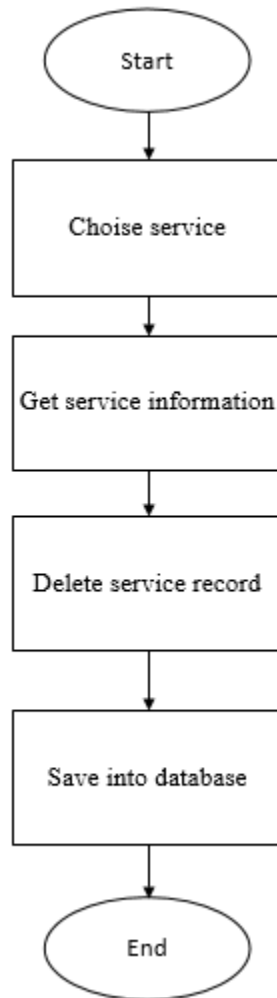
+ Output: Save service information to list of service.

### 3.2.2. Use case “Delete Service”

#### a. Event triggered

When manager decide stop providing service.

#### b. Business process model



#### c. Describe the steps

- Step 1: Get service information

+ Input: Manager choise the service need to delete

+ Output: Service information

- Step 2: Delete service

+ Input: Service code

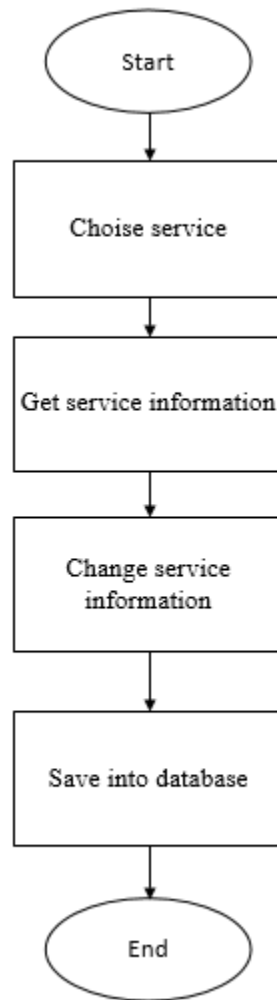
+ Output: Service records is removed from database.

### 3.2.3. Use case “Update Service”

#### a. Event triggered

When manager change service information.

#### b. Business process model



#### c. Describe the steps

- Step 1: Get service information

+ Input: Choise service

+ Outut: Service information

- Step 2: Change service information

+ Input: New information of service

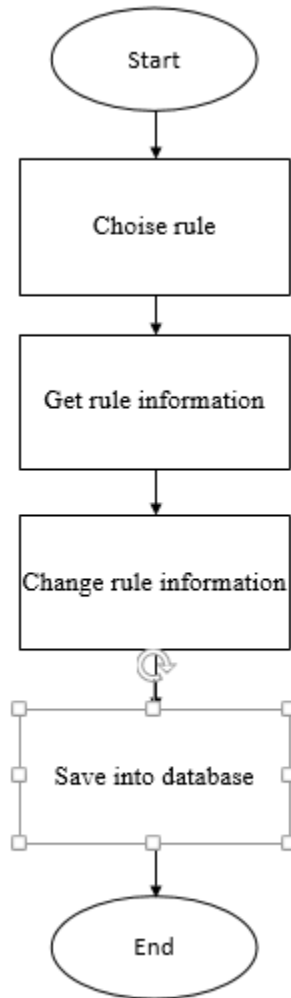
+ Output: Database updated new information of the service.

### 3.3. Use case “Change rules”

#### a. Event triggered

When manager change rule of hotel.

#### b. Business process model



#### c. Describe the steps

- Step 1: Get rule information

+ Input: Choise rule

+ Outut: rule information

- Step 2: Change rule information

+ Input: New information of rule

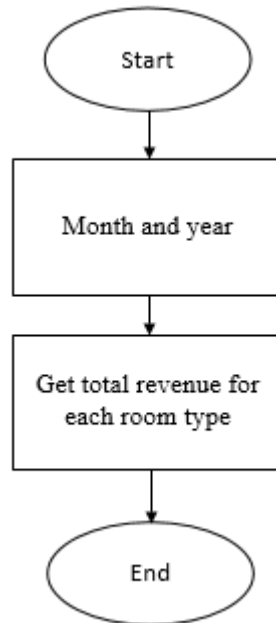
+ Output: Database updated new information of the rule.

### 3.4. Use case “Turnover statistics”

#### a. Event triggered

When manager need audit Turnover.

#### b. Business process model



#### c. Describe the steps

- Step 1: Enter month and year

+ Input: Month and year need to statistic

+ Output: Corresponding Turnovers of room types



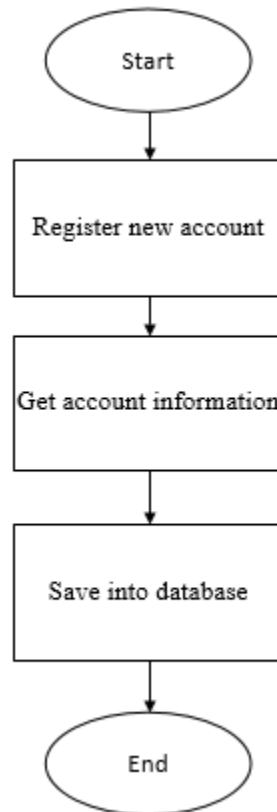
### 3.5. Use case “Manage accounts”

#### 3.5.1. Use case “Add account”

##### a. Event triggered

When manager decide to provide new account in hotel.

##### b. Business process model



##### c. Describe the steps

- Step 1: Import new account information

+ Input: Account information

+ Output: Account information is entered

- Step 2: Get new account information

+ Input: Account information

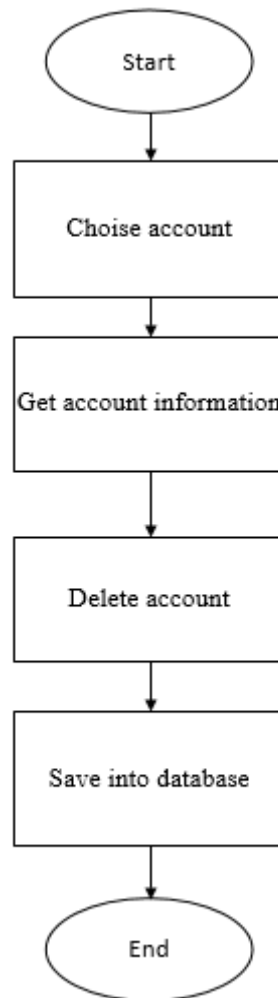
+ Output: Save account information to list of account.

### 3.5.2. Use case “Delete account”

#### a. Event triggered

When manager decide stop providing account.

#### b. Business process model



#### c. Describe the steps

- Step 1: Get account information

+ Input: Manager choise the account need to delete

+ Output: Account information

- Step 2: Delete service

+ Input: Account ID

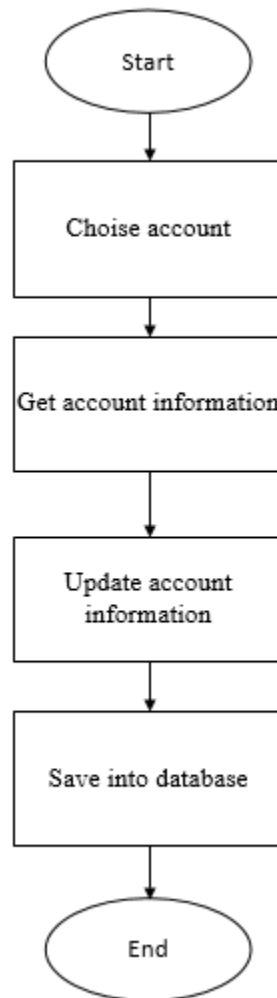
+ Output: Account records is removed from database.

### 3.5.3. Use case “Update account”

#### a. Event triggered

When manager change account information.

#### b. Business process model



#### c. Describe the steps

- Step 1: Get account information

+ Input: Choise account

+ Outut: Account information

- Step 2: Change account information

+ Input: New information of account

+ Output: Database updated new information of the account.

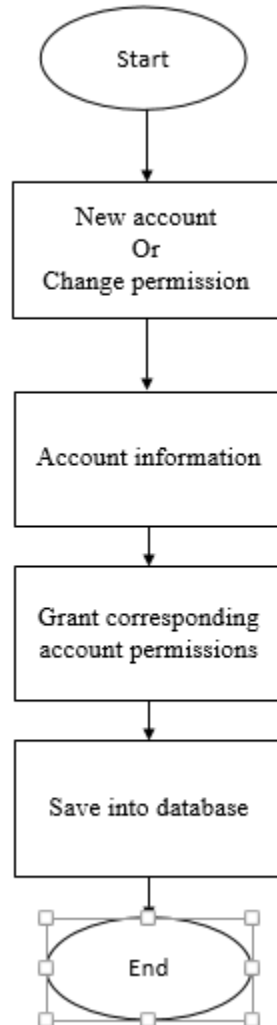
### 3.5.4. Use case “Change security level”

#### a. Event triggered

Each account has different security level to access the system.

When account is added new or changed security level.

#### b. Business process model



#### c. Describe the steps

- Step 1: Get account information

+ Input: Account ID

+ Output: Account information

- Step 2: Grant account permission

+ Input: Grant new permission or change permission

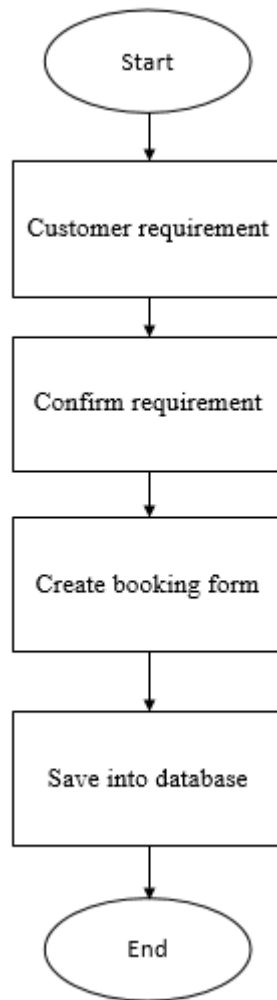
+ Output: Database updated account information

### 3.6. Use case “Book room”

#### a. Event triggered

When customer require booking hotel room.

#### b. Business process model



#### c. Describe the steps

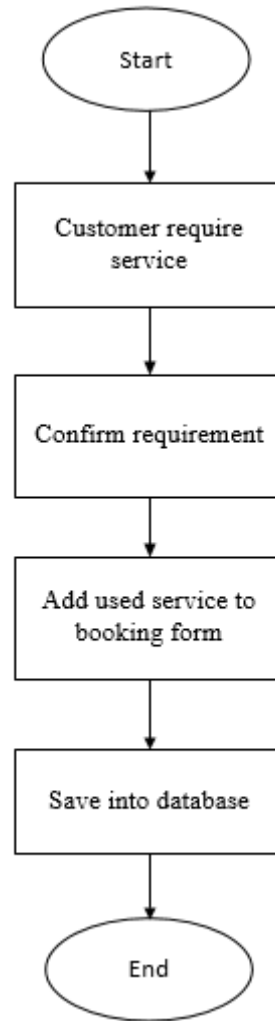
- Step 1: Get customer requirement
  - + Input: Customer requirement
  - + Output: Information of corresponding rooms
- Step 2: Create booking form
  - + Input: Customer and room information
  - + Output: New booking form is saved to database

### 3.7 Use case “Order services”

#### a. Event triggered

When customer call the receptionist to order services.

#### b. Business process model



#### c. Describe the steps

- Step 1: Get order service

+ Input: Customer requirement

+ Output: Get ordered services information

- Step 2: Order service

+ Input: Ordered services information

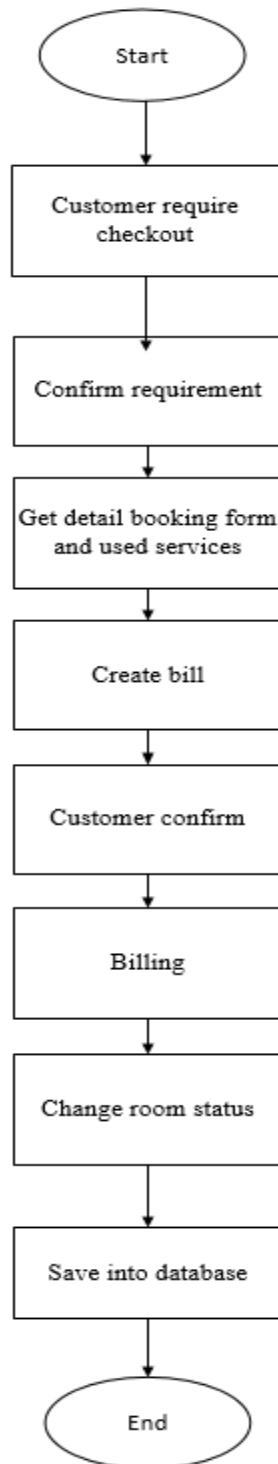
+ Output: Corresponding services were ordered and add used service to booking form.

### 3.8. Use case “Checkout”

#### a. Event triggered

When customer checkout.

#### b. Business process model



c. Describe the steps

- Step 1: Get customer requirement checkout

- + Input: Customer information

- + Output: Rentor room, booking form information and used service

- Step 2: Create bill

- + Input: Booking form information and used service

- + Output: Payment

- Step 3: Checkout

- + Input: Customer confirm

- + Output: Billing, change room status to empty and save changes to database.

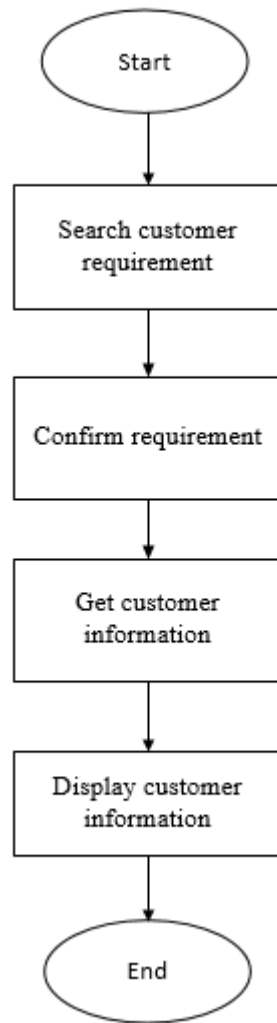


### 3.9. Use case “Search customer information”

#### a. Event triggered

When manager or receptionist search customer

#### b. Business process model



#### c. Describe the steps

- Step 1: Get searching customer requirement

+ Input: Customer information

+ Output: Customer record

- Step 2: Display customer record

+ Input: Customer record

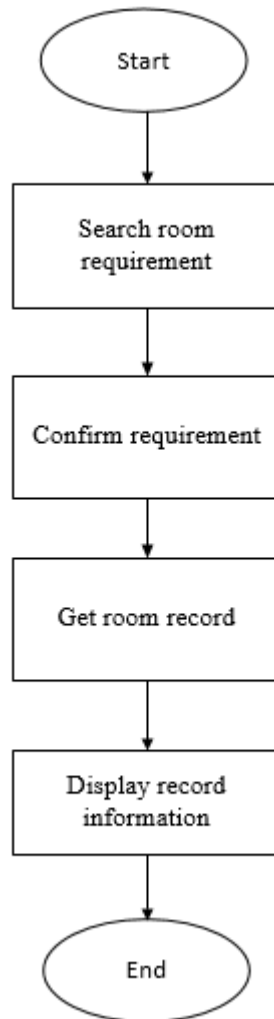
+ Output: Display full customer record

### 3.10. Use case “Search Room Information”

#### a. Event triggered

When manager or receptionist search customer

#### b. Business process model



#### c. Describe the steps

- Step 1: Get searching customer requirement

+ Input: Room information

+ Output: Room record

- Step 2: Display room record

+ Input: Room record

+ Output: Display full room information

#### 4. Specific requirements for functional software

*To perform the above mentioned operations, the software must have the following functions:*

##### 4.1. Specifications "Login" function

| Use Case: {UC_01_Login}  |  |
|--------------------------|--|
| <b>Purpose</b>           | User log in the system based on permission   |
| <b>Description</b>       | Allow user to log in to use the functions that supported for these kinds of user   |
| <b>Actor</b>             | Reception and manager  |
| <b>Pre-condition</b>     | Access the hotel management software   |
| <b>Basic flows</b>       | <ul style="list-style-type: none"><li>- Enter username</li><li>- Enter password</li></ul>  |
| <b>Alternative Flows</b> | <ul style="list-style-type: none"><li>- Fail to log in because of wrong username -&gt; Re-enter</li><li>- Fail to log in because password does not match-&gt; Re-enter</li></ul> |
| <b>Post-condition</b>    | Log in successfully and start to use other functions of the system   |

##### 4.2. Specifications "Create new account" function

| Use Case: {UC_02_Create new account} |                          |
|--------------------------------------|--------------------------|
| <b>Purpose</b>                       | Provide account for user |
| <b>Description</b>                   | Enter user information   |

|                          |  |
|--------------------------|--|
| <b>Actor</b>             | Manager  |
| <b>Pre-condition</b>     | <ul style="list-style-type: none"> <li>- Log in successfully with administrator permission</li> <li>- Click "Staff" on "Admin"</li> <li>- Click button "Provide account" in this form</li> </ul> |
| <b>Basic flows</b>       | Enter user name, password for user account   |
| <b>Alternative Flows</b> | User account information is stored in Database   |
| <b>Post-condition</b>    | Enter username, password for user's account  |

#### 4.3. Specifications “Change Password” function

| Use Case: {UC_03_Change Passwrod} |   |
|-----------------------------------|---|
| <b>Purpose</b>                    | Change user password  |
| <b>Description</b>                | Change user password  |
| <b>Actor</b>                      | Manager + Attendants  |
| <b>Pre-condition</b>              | <ul style="list-style-type: none"> <li>- Log in successfully</li> <li>- Click "Account" in Main menu</li> <li>- Choose "Change password"</li> </ul> |
| <b>Basic flows</b>                | Change user account password  |
| <b>Alternative Flows</b>          | Informations are saved in database  |

|                       |                      |
|-----------------------|----------------------|
| <b>Post-condition</b> | Change user password |
|-----------------------|----------------------|

#### 4.4. *Specifications* “Reservation” function

| Use Case: {UC_04_Reservation} |   |
|-------------------------------|---|
| <b>Purpose</b>                | Reservation   |
| <b>Description</b>            | Check room list and show customer the room that not currently in use, then connect to reservation data  |
| <b>Actor</b>                  | Reception   |
| <b>Pre-condition</b>          | <ul style="list-style-type: none"> <li>- Log in successfully with reception permission or higher</li> <li>- Click “Room”, then “Reservation” This time room list displays</li> <li>- Choose room by images or room number.</li> </ul> |
| <b>Basic flows</b>            | Reservation for customers   |
| <b>Alternative Flows</b>      | Out of room or don’t meet the customer's requirements → Reservation ends  |
| <b>Post-condition</b>         | Reservation complete, save reservation and customer data. Update room status from "Free" to "Currently in use"  |

#### 4.5. *Specifications* “Check out” function

| Use Case: {UC_05_Check out} |  |
|-----------------------------|--|
| <b>Purpose</b>              | Check out by demand of customer or out of time |

|                          |  |
|--------------------------|--|
| <b>Description</b>       | Process of payment, printing invoice and updating status of room,<br>Delete guest list.  |
| <b>Actor</b>             | Reception  |
| <b>Pre-condition</b>     | <ul style="list-style-type: none"> <li>- Log in successfully with reception permission or higher</li> <li>-Click “Room”, then “Check out”. This time room list displays</li> <li>-Click onto room in list.</li> <li>-Click “Add invoice” to invoicing.</li> <li>-Click “Pay” to print invoice for our customer.</li> </ul> |
| <b>Basic flows</b>       | Add and print invoices.  |
| <b>Alternative Flows</b> | Customer have response about invoice(bill) or want to cancel reservation ➔ Delete invoice to re-check invoice information.   |
| <b>Post-condition</b>    | List of rooms, customers is updated by data and status   |

#### 4.6. Specifications “Add room” function

| Use Case: {UC_06_Add room} |   |
|----------------------------|---|
| <b>Purpose</b>             | Add room for rent                             |
| <b>Description</b>         | Add room for rent into database.              |
| <b>Actor</b>               | Manager                                       |
| <b>Pre-condition</b>       | -Log in successfully with manager permission. |

|                          |  |
|--------------------------|--|
|                          | <ul style="list-style-type: none"> <li>- Click “Manager”, click “Room management”. This time room list displays</li> <li>- Click “Add room” to enter information of the new room.</li> </ul> |
| <b>Basic flows</b>       | Type Room Code, Room Status, Room Type, Price and Note   |
| <b>Alternative Flows</b> | Newly added room's code is duplicated → Re-enter   |
| <b>Post-condition</b>    | Room list is updated.  |

#### 4.7. Specifications “Delete room” function

| Use Case: {UC_07_Delete room} |  |
|-------------------------------|--|
| <b>Purpose</b>                | Delete room that will be not in use out of the system  |
| <b>Description</b>            | Delete room that will be not in use out of the system  |
| <b>Actor</b>                  | Manager  |
| <b>Pre-condition</b>          | <ul style="list-style-type: none"> <li>- Log in successfully with manager permission.</li> <li>- Click “Manager”, click “Room management”. This time room list displays</li> <li>- Click onto the room you want to delete. Click “Delete room” to delete data of that room.</li> </ul> |
| <b>Basic flows</b>            | Delete room from list  |
| <b>Alternative Flows</b>      | Chosen room is at status "Full" or "Repairing" → Delete room is not permitted  |

|                       |                       |
|-----------------------|-----------------------|
| <b>Post-condition</b> | Room list is updated. |
|-----------------------|-----------------------|

#### 4.8. *Specifications* “Update room” function

| Use Case: {UC_08_Update room} |  |
|-------------------------------|--|
| <b>Purpose</b>                | Edit room information  |
| <b>Description</b>            | Edit room information  |
| <b>Actor</b>                  | Manager  |
| <b>Pre-condition</b>          | <ul style="list-style-type: none"> <li>-Log in successfully with manager permission.</li> <li>- Click click “Manager”, click “Room management”. This time room list displays</li> <li>- Click the room that you want to edit. Click click “Update room”</li> </ul> |
| <b>Basic flows</b>            | Enter room status, type and notes  |
| <b>Alternative Flows</b>      | Click "Stop" -> Cancel update room   |
| <b>Post-condition</b>         | Room list is updated   |

#### 4.9. *Specifications* “Add service” function

| Use Case: {UC_09_Add service} |                                  |
|-------------------------------|----------------------------------|
| <b>Purpose</b>                | Add new service                  |
| <b>Description</b>            | Add new service when add service |



|                          |   |
|--------------------------|---|
| <b>Actor</b>             | Manager   |
| <b>Pre-condition</b>     | <ul style="list-style-type: none"> <li>-Log in successfully with manager permission.</li> <li>- Click “Manager”, click “Manager service”. This time service list displays</li> <li>- Click “Add service” then enter information about new service.</li> </ul> |
| <b>Basic flows</b>       | Service được add mới  |
| <b>Alternative Flows</b> | New service code is duplicated → Re-enter   |
| <b>Post-condition</b>    | Service list is updated.  |

#### 4.10. Specifications “Delete service” function

| Use Case: {UC_10_Delete service} |   |
|----------------------------------|---|
| <b>Purpose</b>                   | Delete services that are no longer supported  |
| <b>Description</b>               | Delete services that are no longer supported  |
| <b>Actor</b>                     | Manager   |
| <b>Pre-condition</b>             | <ul style="list-style-type: none"> <li>-Log in successfully with manager permission</li> <li>- Click click “Manager”, click “Manager service”. This time service list displays</li> <li>- Click “Delete service” to delete that service.</li> </ul> |
| <b>Basic flows</b>               | Delete service out of database.   |

|                          |                          |
|--------------------------|--------------------------|
| <b>Alternative Flows</b> | None                     |
| <b>Post-condition</b>    | Service list is updated. |

#### 4.11. *Specifications* “Update service” function

|   |  |
|---|--|
| <b>Use Case:</b> {UC_11_Update service} |  |
| <b>Purpose</b>                          | Update service information   |
| <b>Description</b>                      | Create list of the thing that customer require   |
| <b>Actor</b>                            | Manager  |
| <b>Pre-condition</b>                    | <ul style="list-style-type: none"> <li>-Log in successfully with manager permission</li> <li>- Click click “Manager”, click “Manager service”. This time service list displays</li> <li>- Click “Update service” to change service information.</li> </ul> |
| <b>Basic flows</b>                      | Enter service name, unit, price.   |
| <b>Alternative Flows</b>                | Service data is stored in database   |
| <b>Post-condition</b>                   | Service list is updated.   |

#### 4.12. *Specifications* “Order services” function

|   |
|---|
| <b>Use Case:</b> {UC_12_Order services} |
|---|

|                          |  |
|--------------------------|--|
| <b>Purpose</b>           | Order services for customers to select   |
| <b>Description</b>       | Customer call the Reception to request for a service. Reception will execute the request   |
| <b>Actor</b>             | Reception and Manager  |
| <b>Pre-condition</b>     | <ul style="list-style-type: none"> <li>- Log in successfully with reception permission or higher</li> <li>- Click “Manager”, then “Service management”, choose “Order service”.</li> <li>- Or click “Room”, then “Order service”. This time order service table display</li> </ul> |
| <b>Basic flows</b>       | <ul style="list-style-type: none"> <li>- Enter room code</li> <li>- Click service name</li> <li>- Enter quantity</li> </ul>  |
| <b>Alternative Flows</b> | Close ➔ Cancel Order   |
| <b>Post-condition</b>    | Information about "Order services" is saved to attach into invoice   |

#### 4.13. *Specifications* “Room search” function

| Use Case: {UC_13_Room search} |  |
|-------------------------------|--|
| <b>Purpose</b>                | Check information of rooms in hotel              |
| <b>Description</b>            | Provide information by room, price, availability |

|                          |   |
|--------------------------|---|
| <b>Actor</b>             | Reception + Manager   |
| <b>Pre-condition</b>     | <ul style="list-style-type: none"> <li>- Click search button</li> <li>- Click "Room search" button</li> </ul>   |
| <b>Basic flows</b>       | <ul style="list-style-type: none"> <li>- Click type or price range.. to show proper room information</li> <li>- Click empty ,Click button "Reservation".</li> </ul> |
| <b>Alternative Flows</b> | Room doesn't found=> Search again   |
| <b>Post-condition</b>    | Information list of rooms   |

#### 4.14. *Specifications* “Search customer” function

| Use Case: {UC_14_ Search customer} |   |
|------------------------------------|---|
| <b>Purpose</b>                     | Search for detail customer information  |
| <b>Description</b>                 | See in detail customer information  |
| <b>Actor</b>                       | Reception và manager  |
| <b>Pre-condition</b>               | Log in successfully with reception permission or higher <ul style="list-style-type: none"> <li>- Click search button</li> <li>- Click “Search customer” in "Admin"</li> </ul>                   |
| <b>Basic flows</b>                 | <ul style="list-style-type: none"> <li>- Choose customer by writing their name(customer's name,address, customer type, room code) in search box.</li> <li>-Show customer information</li> </ul> |

|                          |  |
|--------------------------|--|
| <b>Alternative Flows</b> | -Search is failed because of wrong information -> Re-enter |
| <b>Post-condition</b>    | List of customers with detail informations                 |

#### 4.15. *Specifications* “Change regulation” function

| Use Case: {UC_15_Change regulation} |   |
|-------------------------------------|---|
| <b>Purpose</b>                      | Reinstall important value of the system   |
| <b>Description</b>                  | Update: price, surcharge, coefficient, number of customers  |
| <b>Actor</b>                        | Manager   |
| <b>Pre-condition</b>                | <ul style="list-style-type: none"> <li>- Log in successfully with manager permission</li> <li>- Click "System", choose "Change regulation"</li> </ul> |
| <b>Basic flows</b>                  | <ul style="list-style-type: none"> <li>- Choose parameters</li> <li>- Enter new values</li> <li>- Click "Update" to save</li> </ul>                   |
| <b>Alternative Flows</b>            | Parameters are not entered → No update occurs<br><br>Close → Section ends   |
| <b>Post-condition</b>               | Change regulation of the system   |

#### 4.16. *Specifications* "Turnover statistics by room type" function

| Use Case: {UC_16_Turnover statistics by room type} |
|--|
|--|

|                          |  |
|--------------------------|--|
| <b>Purpose</b>           | Check turnover statistic by room type  |
| <b>Description</b>       | Calculate turnover statistics in each room in specific period of time  |
| <b>Actor</b>             | Manager  |
| <b>Pre-condition</b>     | <ul style="list-style-type: none"> <li>- Log in successfully with manager permission</li> <li>- Click "Turnover" in "Admin"</li> <li>- Click "Turnover by room type".</li> </ul> |
| <b>Basic flows</b>       | <ul style="list-style-type: none"> <li>- Click reservation time.</li> <li>- Click button "Show statistic"</li> </ul>   |
| <b>Alternative Flows</b> | <ul style="list-style-type: none"> <li>- Enter wrong format data type</li> </ul>   |
| <b>Post-condition</b>    | Turnover statistics in each room in specific period of time  |

#### 4.17. *Specifications* “Show used service list” function

| Use Case: {UC_17_Show used service list} |  |
|--|--|
| <b>Purpose</b>                           | To compare with the invoice (bill)   |
| <b>Description</b>                       | When customer requests service, the history will be stored   |
| <b>Actor</b>                             | Manager  |
| <b>Pre-condition</b>                     | <ul style="list-style-type: none"> <li>- Log in successfully with reception or manager permission</li> <li>- Click “Invoice list”</li> </ul> |
| <b>Basic flows</b>                       | Enter room code and reservation time   |

|                              |   |
|------------------------------|---|
| <b>Alternative<br/>Flows</b> | Wrong room code ➔ Re-enter<br>Reservation time doesn't not match ➔ Re-enter |
| <b>Post-condition</b>        | Corresponding service list is shown   |

## 5. Other requirements

### *5.1. Ease of Use Requirements*

| <b>Request code</b> | <b>Describe</b>   |
|---------------------|---|
| RQ 5.1.1            | Interface language easy to understand                                       |
| RQ 5.1.2            | Symbols should have a consistent meaning                                    |
| RQ 5.1.2            | Support for using fast shortcuts  |
| RQ 5.1.2            | Building the notification system  |
| RQ 5.1.3            | The functions are consistent with the user's habits                         |
| RQ 5.1.4            | Icons, interface close to the working environment                           |
| RQ 5.1.5            | Manipulation directly on icons to represent a function                      |
| RQ 5.1.6            | Functions need to enter data, must create a form for the user to enter data |

### *5.2. Stability requirements*

| <b>Request code</b> | <b>Describe</b> |
|---------------------|-----------------|
| RQ 5.2.1            | Operates 24/24  |

### *5.3. Security requirements*

| <b>Request code</b> | <b>Describe</b>                                   |
|---------------------|---|
| RQ 5.3.1            | The user is granted a unique account and password |
| RQ 5.3.2            | Authorize the user to each function               |



| <b>Request code</b> | <b>Describe</b>  |
|---------------------|--|
| RQ 5.3.3            | The user only has the right to change the login password, can not change the account name. |

#### *5.4. Backup and restore requirements*

| <b>Request code</b> | <b>Describe</b>  |
|---------------------|--|
| RQ 5.4.1            | Ask before performing a function such as delete, save.     |
| RQ 5.4.2            | Temporary deletions can be resumed by using checkpointing. |

#### *5.5. Support eligibility requirements*

| <b>Request code</b> | <b>Describe</b>                                   |
|---------------------|---|
| RQ 5.5.1            | Have help categories to guide the user            |
| RQ 5.5.2            | The information must be short, easy to understand |
| RQ 5.5.2            | Support for installation and operation            |

#### *5.6. Technology requirements*

| <b>Request code</b> | <b>Describe</b>   |
|---------------------|---|
| RQ 5.6.1            | Applying new and modern technologies.   |
| RQ 5.6.2            | Buy software support to ensure the software runs securely in all environments.  |
| RQ 5.6.3            | Minimum configuration: The software must be installed on the windows operating system, from the windows xp version upwards. |

### *5.7. Communication requirements*

| <b>Request code</b> | <b>Describe</b>  |
|---------------------|--|
| RQ 5.7.1            | Use natural language.  |
| RQ 5.7.2            | Use sequence diagrams to represent the operations of the system. |

### *5.8. Requires user documentation and online support*

| <b>Request code</b> | <b>Describe</b>   |
|---------------------|---|
| RQ 5.8.1            | Maintain software to ensure that the software does not give the wrong results, perform the correct functions. |

### *5.9. Legal requirements*

| <b>Request code</b> | <b>Describe</b>   |
|---------------------|---|
| RQ 5.9.1            | Do not violate copyright laws when developing software. |
| RQ 5.9.2            | Ensures software privacy and security                   |

### *5.10. Requirements for applicable standards*

| <b>Request code</b> | <b>Describe</b>                                 |
|---------------------|---|
| RQ 5.10.1           | Meets IEEE standards for software documentation |