

*Subject:* **INTRODUCTION TO SOFTWARE ENGINEERING**

***REQUIREMENTS SPECIFICATION AND ANALYSIS***

**HOTEL MANAGEMENT SOFTWARE**

***Instructor: Mas.Nguyen Thi Thanh Truc***

*Team* **27**

*Name of all member:*

Truong Van Khai – 21520274

Le Ngo Minh Đuc – 21520195

Le Yen Nhi – 21522427

Ngo Phuc Danh – 21521924

*Ho Chi Minh, 1st July 2023*

****

HO CHI MINH CITY NATIONAL UNIVERSITY

**UNIVERSITY OF INFORMATION TECHNOLOGY**

**SOFTWARE ENGINEERING DEPARTMENT**

**CONTENTS**

[REQUIREMENTS ANALYSIS DOCUMENT 3](#_Toc138784298)

[1. Introduction 3](#_Toc138784299)

[*1.1.* *Purpose of the System* 3](#_Toc138784300)

[*1.2.* *Scope of the System* 3](#_Toc138784301)

[*1.3.* *Objectives and Success Criteria of the Project* 3](#_Toc138784302)

[*1.4.* *Definitions and Abbreviations* 3](#_Toc138784303)

[2. Users and Systems requirements 5](#_Toc138784304)

[*2.1.* *Functional Requirements* 5](#_Toc138784305)

[*2.2.* *Nonfunctional Requirements* 5](#_Toc138784306)

[3. Use case 7](#_Toc138784307)

[*3.1. Scope of the System* 7](#_Toc138784308)

[*3.2. Use case diagram* 8](#_Toc138784309)

[4. Data - flow diagram 19](#_Toc138784310)

[*4.1.* *Booking room* 19](#_Toc138784311)

[*4.2.* *See Guest Information* 20](#_Toc138784312)

[*4.3.* *Check Dairy Task* 21](#_Toc138784313)

[*4.4.* *Check The Events* 22](#_Toc138784314)

[*4.5.* *Change Salary* 23](#_Toc138784315)

[*4.6.* *Edit Room Detail* 24](#_Toc138784316)

[*4.7.* *Cancel An Event* 25](#_Toc138784317)

[*4.8.* *Edit Employee’s Task* 26](#_Toc138784318)

[*4.9.* *Log Into System* 27](#_Toc138784319)

[*4.10.* *Get Report Of Employee’s Task* 28](#_Toc138784320)

[*4.11.* *Request Day Off* 29](#_Toc138784321)

[*4.12.* *Create Employee Shift* 30](#_Toc138784322)

[*4.13.* *Check Employee Shift* 31](#_Toc138784323)

[*4.14.* *Check Weekly Schedule* 32](#_Toc138784324)

[*4.15.* *Create Guest Account* 33](#_Toc138784325)

[*4.16.* *Edit Employee’s Information* 34](#_Toc138784326)

[*4.17. Refund Payment* 35](#_Toc138784327)

[*4.18.* *Request Room Service* 36](#_Toc138784328)

[*4.19.* *Request Bill* 37](#_Toc138784329)

# **REQUIREMENTS ANALYSIS DOCUMENT**

## Introduction

### *Purpose of the System*

The main purpose of the Hotel Management System is to make Hotel Staff comfortable to manage their employees and perform tasks, allow them to ensure their guests are pleased by their accommodation and experience. The system also creates a safe and fast booking and satisfying interface for the guests. By using our system to book a room, the user no longer needs to call the hotel and spend too much time on it. As well as being able to book a room in the hotel online safely and in a fast manner. Hotels can easily manage what their employees must do daily, get reports weekly about employee’s work, feedbacks, and workloads, reach every customers and rooms details. Managers can access their employee’s payrolls, and all financial expenses of their hotel.

### *Scope of the System*

This system allows the guests to book rooms and contact the receptionist online, allows the hotel managers to keep track of their employee’s progress and the changes in their hotel. The system also allows for easy access to information and contact between all its users, hotel staff can use the system to organize and schedule their tasks and reach them easily, the system allows for handling of related financial matters.

### *Objectives and Success Criteria of the Project*

The objective of this project is to build a functioning online hotel management system, that provides secure and fast interactions between its users.

The project would be considered a success if:

* + - Project is completed in time.
    - Project meets the appropriate quality targets.
    - Project delivered all items within the agreed scope.
    - Project meets the functional requirements.
    - Project meets the nonfunctional requirements.
    - Project used the approved technology.

### *Definitions and Abbreviations*

**User**: Any person who uses the system.

**Guest:** Any person with the intention to spend time in the hotel and use it’s services.

**Employee**: Any person working at the hotel in any capacity.

**Admin**: Administrator of the hotel also known as General manager.

**Manager**: Manages staff and receptionists and deals with events hosted by the hotel, deals with guests in cases where he is needed.

**Receptionist:** Deals with guest related matters and helps guests when necessary.

**Staff:** Can be cleaners, bellboys, waiters, cooks, and chefs.

**RAD:** Requirement Analysis Document.

**HTML:** Hyper Text Markup Language. It is the standard markup language for documents designed to be displayed in a web browser.

**CSS:** Cascading Style Sheets. CSS is a style sheet language used for describing the presentation of a document written in HTML.

**UX:** User Experience. UX abbreviation is used to define the design process to create products that provide meaningful and proper experiences to users.

**SQL:** Structured Query Language. The SQL language is used in programming for managing data in a relational database.

**JS:** JavaScript. JavaScript is the main programming language of the Web.

**IDE:** Integrated Development Environment. And IDE is a software application that provides facilities to programmers for software development.

## 2. Users and Systems requirements

### *2.1.* *Functional Requirements*

* + - The receptionist can access the management system, reserves rooms and cancel reservations, see information about rooms, guests, and events.
    - The guest should be able to request room service and cleaning, list the events such as conferences.
    - The admin should be able to access all the information related to guests, rooms, employees, and managers, as well as editing said information and salaries.
    - The manager should be able to create/cancel events, access and edit employees’ information as well as rooms information and guest details.
    - The employees and staff should be able to access and list their tasks and information, they can also give feedback on completed tasks.
    - The system allows guests to create their own accounts.
    - System only allows admins to change employee salaries.
    - System authenticates the user if appropriate credentials were given.
    - System show the user only the appropriate pages when valid authentications are given.
    - System only give permission to the manager to create an event.
    - System only allows managers and admins to make an announcement through the System.
    - First time the user logs into the system with his/her account, System necessitate the user to change her/his password.

### *2.2.* *Nonfunctional Requirements*

***Usability***

The system has user friendly and designed with all type of users in mind.

Documentation required to operate the system is also provided.

***Reliability***

The system provides reliable methods of payment and reservations, the functionality of the system also allows for uninterrupted usage and transactions using robust and secure measur

***Performance***

The system loads the pages within 2 seconds and fetches the data from the database in acceptable time interval. It also can handle a large amount of concurrent users and a high number of transactions per second.

***Supportability***

The system uses technologies that are widely supported by the open source community and we will be providing support after the release to fix any possible bugs.

***Implementation***

The system is implemented on a windows platform but designed to be accessible from all platforms since it is a web application.

***Interface***

The interface is designed and implemented in a way that allows ease of use and practicality, without sacrificing usability and performance, it also provides visual hierarchy to help the user perform their task uninterrupted. The system also uses HTTP and TCP/IP protocols.

***Packaging***

The system doesn’t need any installation whatsoever from the user, since it’s web based.

***Legal***

The software is developed in the scope of SOFT3101, no commercial use is expected, thus no license is needed.

## 3. Use case

### *. Scope of the System*

|  |  |  |
| --- | --- | --- |
| **UC Code** | **UC Name** | **Meaning / Function** |
| **UC1** | Booking Room | Booking for a room in our system. |
| **UC2** | see Guest information | Access the guest information in a timely manner. |
| **UC3** | Check daily task | Access information and task in a timely manner. |
| **UC4** | Check the events | Listing the events in a timely manner. |
| **UC5** | Change salary | Changing Hotel Staff slary. |
| **UC6** | Edit room details | Editing the room’s information in details. |
| **UC7** | Cancel an event | Canceling an in event in hotel |
| **UC8** | Edit employee’s tasks. | Editing the employee’s tasks for some cases. |
| **UC9** | Log in to the system | Allow users logging into the system. |
| **UC10** | Get report of employee’s tasks | Allow Manager/Admin to get report of employee’s tasks |
| **UC11** | Request off day | Allow hotel staff to request a day off. |
| **UC12** | Create employee shift | Creating a emplyee shift. |
| **UC13** | Check employee’s shift | Checking employee’s shift for all. |
| **UC14** | Check weekly schedule. | Checking a weekly schedule. |
| **UC15** | Create Guest account | Creating a Guest account. |
| **UC16** | Edit employee’s information | Editing employee’s information. |
| **UC17** | Refund payment | Refunding a payment for guest. |
| **UC18** | Request room service | Requesting room service. |
| **UC19** | Request Bill | Requesting for a bill. |

### *. Use case diagram*

|  |
| --- |
| **Use case name**: Booking Room |
| **Participating actors**: Receptionist, Guest |
| **Flow of Events**:   1. Guest logs into the system with the required credentials. 2. Guest creates a request for creating/canceling a reservation. 3. The system sends the request to Receptionist. 4. Receptionist accesses the system with his user ID and password. 5. Receptionist accepts/denies the request from Guest. 6. The system informs the guest with an announcement. 7. The guest gives their feedback. 8. The system updates according to the new changes and lets the user know. |
| **Entry condition**: Receptionist and Guest logged into system. |
| **Exit conditions**: Guest payment failed, or Guest payment succeeded, or Receptionist completed the booking procedure. |
| **Quality requirement:**   * Searches done by the receptionist should return results withing an acceptable delay. * The guest should be able to cancel a reservation withing a fixed period. |

|  |
| --- |
| **Use case name**: see Guest information |
| **Participating actors**: Receptionist |
| **Flow of events**:   1. Receptionist accesses the system with user ID and password. 2. Receptionist clicks on show all guest information button on the main page. 3. The system shows a page with the list of guests and their information. |
| **Entry condition**: Receptionist logged into system |
| **Exit condition**: The receptionist successfully reached the guest using the information found on the system, or the guest could not be reached/ wrong number. |
| **Quality requirement:**   * The receptionist should be able to access the guest information in a timely manner. * The customer’s sensitive information should be protected without compromising his safety while looking for relevant information. |

|  |
| --- |
| **Use case name**: Check daily task |
| **Participating actors**: Employee |
| **Flow of events**:   1. The employee logs into the system. 2. The employee clicks the Daily Task Button. 3. System displays the list to employee. |
| **Entry condition**: Employee clicks the Daily Task Button |
| **Exit condition**: System displays the list to employee. |
| **Quality requirement:**   * Employee should be able to access their information and task in a timely manner. * The employee should only be able to see his/her own task. |

|  |
| --- |
| **Use case name**: Check the events |
| **Participating actors**: Guest |
| **Flow of events**:   1. The guest logs into the system. 2. The guest clicks on the Check events button. 3. The system shows the guest a list of the events they can attend. |
| **Entry condition**: The guest clicks on the Check events button. |
| **Exit condition**: The system shows the events available. |
| **Quality requirement:**   * The guest listed the events in a timely manner. * The guest did not have to compromise any personal information to perform this task. |

|  |
| --- |
| **Use case name**: Change salary |
| **Participating actors**: Admin |
| **Flow of events**:   1. The admin logs into the system. 2. The admin clicks on list of employees. 3. The system shows the list of the employees. 4. The admin selects an employee or group of employees. 5. The admin clicks Change salary button and changes the salary/salaries. 6. The system updates the employees’ information accordingly. |
| **Entry condition**: The admin clicks on list of employees. |
| **Exit condition**: The system updates the employees’ information successfully. |
| **Quality requirement:**   * The admin should be able to access the staff list in a timely manner * System interface gives meaningful feedbacks. |

|  |
| --- |
| **Use case name**: Edit room details |
| **Participating actors**: Admin |
| **Flow of events**:   1. The admin logs into the system. 2. The admin clicks on list of rooms button. 3. The system shows the list of the rooms. 4. The admin selects a room from the list and clicks edit button. 5. The admin inputs the changes and clicks save button. 6. The system updates the room information. |
| **Entry condition**: The admin clicks on list of rooms. |
| **Exit condition**: The system updates the room information |
| **Quality requirement:**   * The admin accessed the room details in a timely manner * The admin changed the information in question without compromising other sensitive data. |

|  |
| --- |
| **Use case name**: Cancel an event |
| **Participating actors**: Manager |
| **Flow of events**:   1. The Manager logs into the system. 2. The Manager clicks on list of events. 3. The system shows the list of the events available. 4. The Manager selects an event. 5. The Manager clicks cancel event button. 6. The system updates the list of events accordingly. 7. The system sends a notification to all attendees and staff related. |
| **Entry condition**: The manager logged into the system. |
| **Exit condition**: The system updates the list of events successfully, or the cancelation is not permitted for legal reasons. |
| **Quality requirement:**   * The manager should be able to cancel the event withing the period agreed on with the attendees and guests. * System interface gives meaningful feedbacks |

|  |
| --- |
| **Use case name**: Edit employee’s tasks. |
| **Participating actors**: Manager |
| **Flow of events**:   1. Manager logs into the system. 2. Manager clicks on Edit employee’s task button 3. System shows the manager a list of all the employees. 4. The manager selects an employee to edit a task for. 5. The system shows the selected employee’s current tasks. 6. The manager chooses a task to edit. 7. The system updates the employee’s list of tasks. |
| **Entry condition**: The manager logged into the system. |
| **Exit condition**: The manager successfully edited the task for that employee. |
| **Quality requirement:**   * The manager accessed the employee’s list of tasks in a timely manner. * The manager should be able to edit the employee’s task without interfering with other employees’ tasks. |

|  |
| --- |
| **Use case name**: Log into the system |
| **Participating actors**: System User |
| **Flow of Events**:   1. The user opens the website. 2. The user clicks the log in button. 3. System shows a form with username and password 4. User enters username and password and submit the form 5. System validates the inputs. 6. System authenticate the user and redirect him to the dashboard. |
| **Entry condition**: Click the log in button |
| **Exit conditions**: Submitting the wrong information to system or logging in successfully. |
| **Quality requirement:**   * System should respond in few seconds * System interface gives meaningful feedbacks. |

|  |
| --- |
| **Use case name**: See all Guests in a specific time interval |
| **Participating actors**: Manager/Admin |
| **Flow of Events**:   1. Manager/Admin logs into the system. 2. Manager/Admin clicks on List Guests. 3. The system shows Manager/Admin the list of all guests and input for time interval. 4. The Manager/Admin fill in the time interval. 5. The system updates the list of the guests that meet that requirement. |
| **Entry condition**: Clicking the list guest button |
| **Exit conditions**: Updating the list of the guests successfully. |
| **Quality requirement:**   * System should respond in few seconds. * System interface gives meaningful feedbacks. |

|  |
| --- |
| **Use case name**: Get report of employee’s tasks |
| **Participating actors**: Manager/Admin |
| **Flow of Events**:   1. Manager/Admin logs into the system 2. Manager/Admin clicks on get report of employee’s task button. 3. The system shows the Manager/Admin a list of employees to choose from. 4. Manager/Admin chooses an employee from the list and clicks get report button. 5. The system generates the employee’s report and displays the report. |
| **Entry condition**: Manager/Admin clicks on get report of employee’s task button. |
| **Exit conditions**: Successfully generating the report. |
| **Quality requirement:**   * System should respond in a timely manner. * System interface gives meaningful feedbacks. |

|  |
| --- |
| **Use case name**: Request off day |
| **Participating actors**: Employee, Admin/Manager. |
| **Flow of Events**:   1. Employee logs into the system. 2. Employee clicks on request off day. 3. The system asks about the date of the off day. 4. The Employee selects a date interval. 5. The system creates a request to be sent to the Admin/Manager. 6. The Admin/Manager gets the request and gives his/her feedback. 7. The system updates the Employee request. |
| **Entry condition**: Employee/Receptionist clicks on request off day |
| **Exit conditions**: The system updates the Employee/Receptionist request. |
| **Quality requirement:**   * System interface gives meaningful feedbacks. * Employee/Receptionist cannot request more than two weeks of break. * Employee/Receptionist cannot request an off day in the current week of the request. |

|  |
| --- |
| **Use case name**: Create employee shift |
| **Participating actors**: Manager |
| **Flow of Events**:   1. Manager logs into the system. 2. Manager clicks on Create employee shift button. 3. System shows the manager a list of all the employees. 4. The manager selects an employee to create a shift for. 5. The system shows the selected employee’s working hours and days. 6. The manager chooses working hours for the shift. 7. The system updates the employee’s schedule. |
| **Entry condition**: Manager clicks on Create employee shift button |
| **Exit conditions**: The system updates the employee’s schedule, or if the employee does not have suitable hour for the shift. |
| **Quality requirement:**   * System interface gives meaningful feedbacks. * The system returns the list of employees in seconds. * The manager should be able to create the employee’s shift without interfering with other employees’ shifts. |

|  |
| --- |
| **Use case name**: Check employee’s shift |
| **Participating actors**: Manager |
| **Flow of Events**:   1. Manager logs into the system. 2. Manager clicks on check shift button 3. System shows the manager a list of all the employees. 4. The manager selects an employee to check his shift. 5. The system shows the selected employee’s working hours and days. |
| **Entry condition**: Manager clicks on check shift button |
| **Exit conditions**: The system shows the selected employee’s working hours and days. |
| **Quality requirement:**   * System interface gives meaningful feedbacks. * The system returns the list of employees in seconds. |

|  |
| --- |
| **Use case name**: Check weekly schedule. |
| **Participating actors**: Employee |
| **Flow of Events**:   1. Employee logs into the system. 2. Employee clicks on weekly schedule. 3. The system shows the employee’s schedule. |
| **Entry condition**: Employee clicks on weekly schedule |
| **Exit conditions**: The system shows the employee’s schedule |
| **Quality requirement:**   * System should respond in a short time of seconds. * System interface gives meaningful feedbacks. |

|  |
| --- |
| **Use case name**: Create Guest account |
| **Participating actors**: Guest |
| **Flow of Events**:   1. The Guest opens the website. 2. Click the Create Account Button. 3. System shows a form with necessary information. 4. Guest enters necessary information and submits the form. 5. System validates if the guest already exists or not. 6. System saves guest to the database and redirects the guest to the login page. |
| **Entry condition**: The guest clicks on the register button. |
| **Exit conditions**: Trying to register an already existing guest or successfully registered. |
| **Quality requirement:**   * System should respond in a short time of seconds. * System interface gives meaningful feedbacks. |

|  |
| --- |
| **Use case name**: Edit employee’s information |
| **Participating actors**: Manager |
| **Flow of Events**:   1. Manager logs in to the system. 2. Manager clicks on show employees’ list. 3. The system shows the list of all the employees. 4. Manager chooses the employee from the employee list. 5. System shows the chosen Employee’s detailed information. 6. Manager edits the information wanted on that page. 7. Manager clicks on the save button to save the changes on employee’s information. 8. System saves the changes on the screen and shows them to the Manager. |
| **Entry condition**: Clicking edit employee’s information button. |
| **Exit conditions:** Entering invalid format of information to different areas, deleting necessary information, or successfully editing and saving the information tab. |
| **Quality requirement:**   * System should respond in a short time of seconds. * System interface gives meaningful feedbacks. * System must take the compulsory information. |

|  |
| --- |
| **Use case name**: Refund payment |
| **Participating actors**: Guest, Manager |
| **Flow of Events**:   1. Guest logs in to the system with his/her user ID and password. 2. Guest clicks on the refund button. 3. System opens the cancellation tab according to hotels policy. 4. System shows a form to guest about cancellation. 5. Guest fills the form and clicks on the save button. 6. System saves and sends this request form to the Manager. 7. According to hotel policy and hotel rules Manager can approve the form. 8. Once the system gets the approval from the manager, refunds the payment to the Guest, and deletes the reservation from the system. |
| **Entry condition**: Clicking cancel reservation button. |
| **Exit conditions:** Cancellation is not possible according to hotel policy and rules. Or request successfully sent to the Manager. |
| **Quality requirement:**   * System should respond in a short time of seconds. * System interface gives meaningful feedback * In case of refunding, system should send the money to Guest’s account within a reasonable time interval. |

|  |
| --- |
| **Use case name**: Request room service |
| **Participating actors**: Guest, Staff |
| **Flow of Events**:   1. Guest logs in to the system with his/her user ID and password. 2. Guest clicks on the request room service button. 3. System sends this request to the available Staff page. 4. System adds the task to the Staff’s task list. 5. Staff goes to the Guest’s room with necessary items according to the request. 6. Staff access their page and mark the new task as complete. 7. The guest’s bill is updated in accordance with the service provided. |
| **Entry condition**: Clicking request room service button. |
| **Exit conditions**: Request successfully added to the current task list. |
| **Quality requirement:**   * System should respond in a short time of seconds. * System interface gives meaningful feedbacks. |

|  |
| --- |
| **Use case name**: Check task for specific time interval |
| **Participating actors**: Employee |
| **Flow of Events**:   1. Employee clicks the see All Tasks Button. 2. System shows two inputs for starting and end-date. 3. Employee enters the starting date and end-date. 4. System creates a task list for specified time interval. 5. System displays the list to the employee. |
| **Entry condition**: Clicking see All Tasks button. |
| **Exit conditions**: System show the list for specific time interval, or no invalid date was given. |
| **Quality requirement:**   * System should respond in few seconds * System interface gives meaningful feedbacks. |

|  |
| --- |
| **Use case name**: Request Bill |
| **Participating actors**: Guest, Receptionist |
| **Flow of Events**:   1. Guest logs into the system 2. The guest clicks on request bill button. 3. The system sends the request to the receptionist. 4. The receptionist accesses the guest’s request. 5. The receptionist approves the guest’s request. 6. The system sends a bill to the guest’s account. |
| **Entry condition**: Guest clicks request bill button |
| **Exit conditions**: Bill successfully sent to the Guest, there is no such a bill. |
| **Quality requirement:**   * System should respond in few seconds * System interface gives meaningful feedbacks. |

1. **Data-flow diagram**
   1. ***Booking room***

A diagram of a hotel room

Description automatically generated with low confidence

* + 1. **Data flow:**
* D1: Room information (number of beds, room type, price, opacity), guest information (name, email, phone number).
* D2: None.
* D3: D1.
* D4: D1 + room status.
* D5: D4.
* D6: Feedback message (room status confirmation).
* D7: D6.
* D8: D6.
  + 1. **Algorithms:**
* Step 1: Receive D1 from user.
* Step 2: Connecting to database.
* Step 3: Read D4 from auxiliary memory.
* Step 4: Checking the availability of room information.
* Step 5: Checking the availability of user information.
* Step 6: If step 4 or 5 invalid, move to step 9.
* Step 7: Set status room to booked and save to auxiliary memory.
* Step 8: Print D6.
* Step 9: Close database connection.
* Step10:End.

* 1. ***See Guest Information***

A diagram of a flowchart

Description automatically generated with low confidence

* + 1. **Data flow:**
* D1: Receptionist choose see guest information function.
* D2: None.
* D3: A list of guest information.
* D4: D1.
* D5: D3.
* D6: D3.
  + 1. **Algorithms:**
* Step 1: Receive D1 from user.
* Step 2: Connecting to database.
* Step 3: Checking the number of guest.
* Step 4: If there is no guest, move to step 7.
* Step 5: Creating a list of guest information include room, name, price, opacity, number of bed, email, phone number.
* Step 6: Print D5.
* Step 7: Close database connection.
* Step 8: End.
  1. ***Check Dairy Task***

A diagram of a dairy task

Description automatically generated with low confidence

* + 1. **Data flow:**
* D1: Employee information, choosing dairy task function.
* D2: None.
* D3: A list of task.
* D4: D1.
* D5: D3 + D1.
* D6: D3.
  + 1. **Algorithm:**
    - Step 1: Receive D1 from user.
    - Step 2: Conencting to database.
    - Step 3: Checking the availability of employee information.
    - Step 4: If invalid, move to step 7.
    - Step 5: A list of task from database.
    - Step 6: Print D5.
    - Step 7: Close database connection.
    - Step 8: End.
  1. ***Check The Events***

A diagram of a flowchart

Description automatically generated with low confidence

* + 1. **Data flow:**
    - D1: Guest information(name, phone number, email, room number) and choosing check event function.
    - D2: None.
    - D3: List of available events.
    - D4: D1.
    - D5: D3 + D1.
    - D6: D3.
    1. **Algorithm:**
    - Step 1: Receive D1 from user.
    - Step 2: Connecting to database.
    - Step 3: Checking the availability of D1.
    - Step 4: If D1 is invalid or there is no events from database, move to step 7.
    - Step 5: List of events.
    - Step 6: Print D5.
    - Step 7: Close database connection.
    - Step 8: End.
  1. ***Change Salary***

**A diagram of a change salary

Description automatically generated**

* + 1. **Data flow:**
    - D1: Manager information (name, role, phone number, email), number of salary.
    - D2: None.
    - D3: Employee’s salary after changing notification.
    - D4: D1.
    - D5: D3 + D1.
    - D6: D3.
    1. **Algorithm:**
    - Step 1: Receive D1 from user.
    - Step 2: Connecting to database.
    - Step 3: Checking the availability of D1.
    - Step 4: Checking the number of salary if suit up to minimum and maximum salary.
    - Step 5: If invalid, move to step 8.
    - Step 6: Set salary for employee and save to auxiliary memory.
    - Step 7: Print D5.
    - Step 8: Close data connection.
    - Step 9: End.
  1. ***Edit Room Detail***

**A diagram of a system

Description automatically generated with low confidence**

* + 1. **Data flow:**
    - D1: Admin information and room information (room, number of beds, room type).
    - D2: None.
    - D3: Change room detail notification.
    - D4: D1.
    - D5: D1 + D3.
    - D6: D3.
    1. **Algorithm:**
    - Step 1: Receive D1 from user.
    - Step 2: Connecting to database.
    - Step 3: Checking the availability of D1.
    - Step 4: If there is any wrong information, move to step 7.
    - Step 5: Set new detail for room and save to auxiliary memory.
    - Step 6: Print D5.
    - Step 7: Close data connection.
    - Step 8: End.
  1. ***Cancel An Event***

**A picture containing diagram, line, technical drawing, text

Description automatically generated**

* + 1. **Data flow:**
    - D1: Manager information and list of event information.
    - D2: None.
    - D3: D1 + cancel an event.
    - D4: notification about event which is cancelled.
    - D5: D4.
    - D6: None.
    - D7: D1 + D4.
    - D8: D4.
    1. **Algorithm:**
    - Step 1: receive D1 from user.
    - Step 2: Connecting to database.
    - Step 3: Checking the availability of D1.
    - Step 4: If the information from manager or event is wrong, move to step 7.
    - Step 5: Cancel the event the save to auxiliary memory.
    - Step 6: Print event cancelled notification.
    - Step 7: Close database connection.
    - Step 8: End.
  1. ***Edit Employee’s Task***

A picture containing diagram, text, line, technical drawing

Description automatically generated

* + 1. **Data flow:**
    - D1: Employee’s information (name, role, phone number, email).
    - D2: None.
    - D3: D1 + new task for employee.
    - D4: Edit task and save.
    - D5: Change in task notification.
    - D6: None.
    - D7: D1 + D5.
    - D8: D5.
    1. **Algorithm:**
    - Step 1: receive D1 from user.
    - Step 2: connecting to database.
    - Step 3: Checking the availability of D1.
    - Step 4: If invalid, move to step 9.
    - Step 5: Set new task for employee.
    - Step 6: Send notification for employee.
    - Step 7: Print D7.
    - Step 8: Save to auxiliary memory.
    - Step 9: Close database connection.
    - Step 10: End.
  1. ***Log Into System***

A diagram of a system

Description automatically generated with low confidence

* + 1. **Data flow:**
    - D1: Account information (username, password, role).
    - D2: None.
    - D3: Log in status notification.
    - D4: D1.
    - D5: D1 + D3.
    - D6: D4.
    1. **Algorithm:**
    - Step 1: Receive D1 from user.
    - Step 2: Connecting to database.
    - Step 3: Checking the account.
    - Step 4: If invalid, move to step 7.
    - Step 5: Send notification to user.
    - Step 6: Print D5.
    - Step 7: Close database connection.
    - Step 8: End.
  1. ***Get Report Of Employee’s Task***

**A diagram of a work flow

Description automatically generated with low confidence**

* + 1. **Data flow:**
    - D1: Manager/admin account information (username, password, role), get report function.
    - D2: None.
    - D3: List of task.
    - D4: Permission of user access.
    - D5: D3.
    - D6: D3.
    1. **Algorithm:**
    - Step 1: Receive D1 from user.
    - Step 2: Connecting to database.
    - Step 3: Checking the account availability.
    - Step 4: Checking the amount of task.
    - Step 5: If account is invalid or empty task, move to step 8.
    - Step 6: Send the list of task to the user .
    - Step 7: Print D5.
    - Step 8: Close database connection.
    - Step 9: End.
  1. ***Request Day Off***

**A diagram of a company

Description automatically generated with low confidence**

* + 1. **Data flow:**
    - D1: Employee’s information (name, phone number, email, role).
    - D2: None.
    - D3: D1 , date of absence.
    - D4: send report of request .
    - D5: D3.
    - D6: Request confirmation.
    - D7: D4 + D6.
    - D8: D6.
    1. **Algorithms:**
    - Step 1: Receive D1 from user.
    - Step 2: Connecting to database.
    - Step 3: Check the avalaible of employee.
    - Step 4: If invalid, move to step 8.
    - Step 5: Send report of request to manager.
    - Step 6: Send manager’s confimation to database and save.
    - Step 7: Print D7.
    - Step 8: Close database connection.
    - Step 9: End.
  1. ***Create Employee Shift***

**A picture containing text, diagram, line, plan

Description automatically generated**

* + 1. **Data flow:**
    - D1: Manager information and employee information.
    - D2: None.
    - D3: Create new shift base on D1 and update.
    - D4: D1.
    - D5: Creating shift notifications.
    - D6: D5.
    1. **Algorithm:**
    - Step 1: Receive D1 from user.
    - Step 2: Connecting to database.
    - Step 3: Checking the availability of manager account.
    - Step 4: If invalid, move to step 8.
    - Step 6: Creating the new shift and save to database.
    - Step 7: Print D5.
    - Step 8: Close database connection.
    - Step 9: End.
  1. ***Check Employee Shift***

A diagram of a company

Description automatically generated with low confidence

* + 1. **Data flow:**
    - D1: Employee’s information which manager want to check.
    - D2: None.
    - D3: The list of employee shift.
    - D4: Permisstion to access emplopee shift.
    - D5: D3 + D1.
    - D6: D3.
    1. **Algorithm:**
    - Step 1: Receive D1 from user.
    - Step 2: Connecting to database.
    - Step 3: Checking the availability of employee information.
    - Step 4: If invalid, move to step 8.
    - Step 5: Creating a list of shift from database.
    - Step 6: Send to manger the list.
    - Step 7: Print D5.
    - Step 8: Close database connection.
    - Step 9: End.
  1. ***Check Weekly Schedule***

A diagram of a schedule

Description automatically generated with low confidence

* + 1. **Data flow:**
    - D1: Employee information (name, phone number, email, role).
    - D2: None.
    - D3: List of weekly schedule.
    - D4: D1.
    - D5: D1 + D3.
    - D6: D3.
    1. **Algorthm:**
    - Step 1: receive D1 from user.
    - Step 2: Connecting to database.
    - Step 3: checking the availability of employee.
    - Step 4: if invalid, move to step 8.
    - Step 5: Creating a list of weekly schedule.
    - Step 6: send to user.
    - Step 7: print D5.
    - Step 8: Close database connection.
    - Step 9: end.
  1. ***Create Guest Account***

A diagram of a guest account

Description automatically generated with medium confidence

* + 1. **Data flow:**
    - D1: Guest information (name, phone number, email, username, password).
    - D2: None.
    - D3: Send craeting account notification.
    - D4: D1.
    - D5: D1 + D3.
    - D6: D3.
    1. **Algorithm:**
    - Step 1: Receive D1 from user.
    - Step 2: Connecting to database.
    - Step 3: Checking the availability of guest information.
    - Step 5: Checking the availability of username.
    - Step 6: If step 4 or 5 is invalid, move to step 10.
    - Step 7: Create the account.
    - Step 8: Save to auxiliary memory.
    - Step 9: Print D5.
    - Step 10: Close database connection.
    - Step 11: End.
  1. ***Edit Employee’s Information***

A picture containing text, diagram, line, plan

Description automatically generated

* + 1. **Data flow:**
    - D1: Employee based information before editting.
    - D2: None.
    - D3: List of information after editing.
    - D4: D1.
    - D5: D1 + D3.
    - D6: D3.
    1. **Algorithm:**
    - Step 1: Receive D1 from user.
    - Step 2: Connecting to database.
    - Step 3: Checking the availability of employee information.
    - Step 4: If invalid, move to step 8.
    - Step 5: Changing the base information.
    - Step 6: Save to auxiliary memory.
    - Step 7: Print D5.
    - Step 8: Close database connection.
    - Step 9: End.

***4.17. Refund Payment***

A diagram of a refund payment

Description automatically generated

* + 1. **Data flow:**
    - D1: Room information(room type, price), guest information(name, phone number) .
    - D2: None.
    - D3: D1 + form submit.
    - D4: report of refund payment.
    - D5: D4 + D1.
    - D6: Confirmation report .
    - D7: D6 + D5.
    - D8: D6.
    1. **Algorithm:**
    - Step 1: Receive D1 from user.
    - Step 2: Connecting to database.
    - Step 3: Checking the availability of booked room.
    - Step 4: If invalid, move to step 9.
    - Step 5: Fill in information to a report.
    - Step 6: Send report to manager.
    - Step 7: Manager confirmation.
    - Step 8: Print D7.
    - Step 9: Close database connection.
    - Step 10: End.
  1. ***Request Room Service***

A diagram of a service

Description automatically generated with low confidence

* + 1. **Data flow:**
    - D1: Room request service function and room information.
    - D2: None.
    - D3: D1.
    - D4: The list of available services.
    - D5: D1 + D4.
    - D6: Support service and send bill.
    - D7: D4 + bill.
    - D8: D7.
    1. **Algorithm:**
    - Step 1: Receive D1 from database.
    - Step 2: Connceting to database.
    - Step 3: Checking the room has what services.
    - Step 4: If invalid, move to step 8.
    - Step 5: Supporting sevice.
    - Step 6: Create a list of service and bill from database.
    - Step 7: Print D7.
    - Step 8: Close database connection.
    - Step 9: End.
  1. ***Request Bill***

**A picture containing diagram, technical drawing, line, plan

Description automatically generated**

* + 1. **Data flow:**
    - D1: Guest information(name, phone number, email and room information(number, capacity, number of bed, price).
    - D2: None.
    - D3: D1.
    - D4: Send notification .
    - D5: Receive D4.
    - D6: Bill.
    - D7: D1 + D6.
    - **D8: D6.**
    1. **Algorithm:**
    - Step 1: Receive D1 from user.
    - Step 2: Connecting to database.
    - Step 3: Checking the availability of guest information.
    - Step 4: Checking the availability of room booked.
    - Step 6: If 4 or 5 invalid, move to step 9.
    - Step 7: Create a list of bill from database.
    - Step 8: Print D7.
    - Step 9: Close database connection.
    - Step 10: End.