3/16/2016

The Lucid Palace

Hotel Information Management System

Kevin Ma (300867968), Vinood Persad (300861530), Ostap Harmarnyk (300836326), Alexey Garin (300866451), Kareem Marzouk (300845740)

COMP225 Software Engineering Methodologies I

Professor VIJI Tiruchengode Angamuthu

Contents

[Part A: The Business/Domain Model 3](#_Toc448240999)

[Problem: 3](#_Toc448241000)

[Needs: 3](#_Toc448241001)

[Project Subsystems: 3](#_Toc448241002)

[System Vision Document 3](#_Toc448241003)

[Problem Description 3](#_Toc448241004)

[System Capabilities 3](#_Toc448241005)

[Business Benefits 3](#_Toc448241006)

[Work Breakdown Structure (WBS) 4](#_Toc448241007)

[Gantt Chart 4](#_Toc448241008)

[Work Flows 5](#_Toc448241009)

[1.0 Cancelling Reservation 5](#_Toc448241010)

[2.0 Making Reservations 7](#_Toc448241011)

[3.0 Check out / Credit Card Charge 9](#_Toc448241012)

[Part B: Use Case Model – Domain Classes 11](#_Toc448241013)

[Event Table 11](#_Toc448241014)

[UML Use-Case Diagram 13](#_Toc448241015)

[Prioritized, Consolidated List of Use Cases in the Two Essential Workflows 14](#_Toc448241016)

[Brief Use Case Descriptions 15](#_Toc448241017)

[Domain Class Diagram 16](#_Toc448241018)

[Part C: Procedural Model 17](#_Toc448241019)

[ERD 17](#_Toc448241020)

[Table Structures 18](#_Toc448241021)

[System Sequence Diagram 20](#_Toc448241022)

[Making Reservation 20](#_Toc448241023)

[Cancelling Reservation 21](#_Toc448241024)

[State Machine Diagrams 22](#_Toc448241025)

[Payment 22](#_Toc448241026)

[Reservation 23](#_Toc448241027)

[Customer 24](#_Toc448241028)

[Front Desk 25](#_Toc448241029)

[Room 26](#_Toc448241030)

[Gantt Chart 27](#_Toc448241031)

[Team Meeting Minutes 28](#_Toc448241032)

# Part A: The Business/Domain Model

### Problem:

Application is able to keep track of guest reservations, reservation cancellation and credit card transactions.

### Needs:

* Maintain and track hotel reservations
* Maintain and track hotel cancellations
* Maintain and record credit card transactions

### Project Subsystems:

* Reservation database maintenance (booking/cancellation)
* Credit card transaction
* Room availability database

## System Vision Document

### Problem Description

With globalization constantly on the growth, hotel stays are becoming an integral part of everyday life. The efficiency of hotel booking and check-ins are very important aspects of the hotel industry. They allow hotels to facilitate excellent customer service and develop the trust of guests. However, The Lucid Palace’s information processing rate is unable to keep up with the growing demand of customers. This has slowly but surely deterred regulars from shying away from The Lucid Palace. Thus, it is important for The Lucid Palace to enhance its information management behind the scenes.

In order to accomplish these goals, it is recommended that a new information management system be created to collect and store guest information. The system must be able to collect credit card information and store them with high security. It must also be able to edit reservations, cancel reservations and perform quick reservation confirmations. This system should be deployed on stationary equipment to ensure security of the information network.

### System Capabilities

The new system should be capable of:

* Collecting and storing guest information
* Collecting and storing credit card information. (Following Privacy Act)
* Collecting guests’ preferences of their stay at the hotel
* Connecting through Ethernet connection only
* Connecting to Hotel Reservation Database
* Connecting to Hotel Rooms Database
* Processing Credit Cards
* Cancelling reservations
* Editing current or upcoming reservations

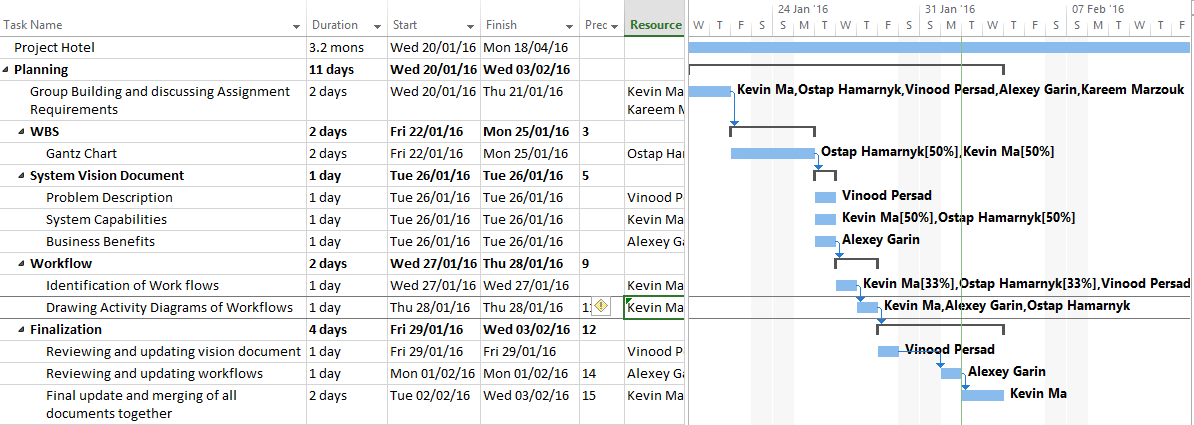
### Business Benefits

The business benefits that are anticipated from the deployment of the system are as follows:

* Maintain new and current reservation information about guests, thereby improving the quality and speed of future hotel bookings and check-ins
* Maintain future and current credit card processing, thereby improving the security and speed of payment when checking out of a hotel
* Expedite booking process to allow faster guest check in and check out, thereby improving customer turnover rate and increasing hotel goodwill
* Expand booking options for customers at check in, thereby improving customer satisfaction and increasing hotel goodwill
* Enable and track cross-hotel discounts/promotions (at the discretion of the hotel), thereby catching business trends more rapidly and being able to match or provide better deals
* Allow customer ‘loyalty’ reward system and alike (at the discretion of the hotel), thereby increasing customer satisfaction and increasing hotel goodwill

## Work Breakdown Structure (WBS)

### Gantt Chart



## Work Flows

### 1.0 Cancelling Reservation

* 1. Customer requests reservation cancellation
  2. Employee requests reservation information
  3. Customer provides reservation information
  4. Employee checks for reservation in the system
     1. If there is such a reservation record, proceed to 1.5, if not employee tells customer about no reservation existing and proceed to 1.2
  5. System sends validation message about reservation being found in record
  6. Employee sends request to system to cancel the reservation
  7. System removes reservation from the database
  8. System sends confirmation that reservation has been removed
  9. Employee notifies customer that the reservation has been cancelled

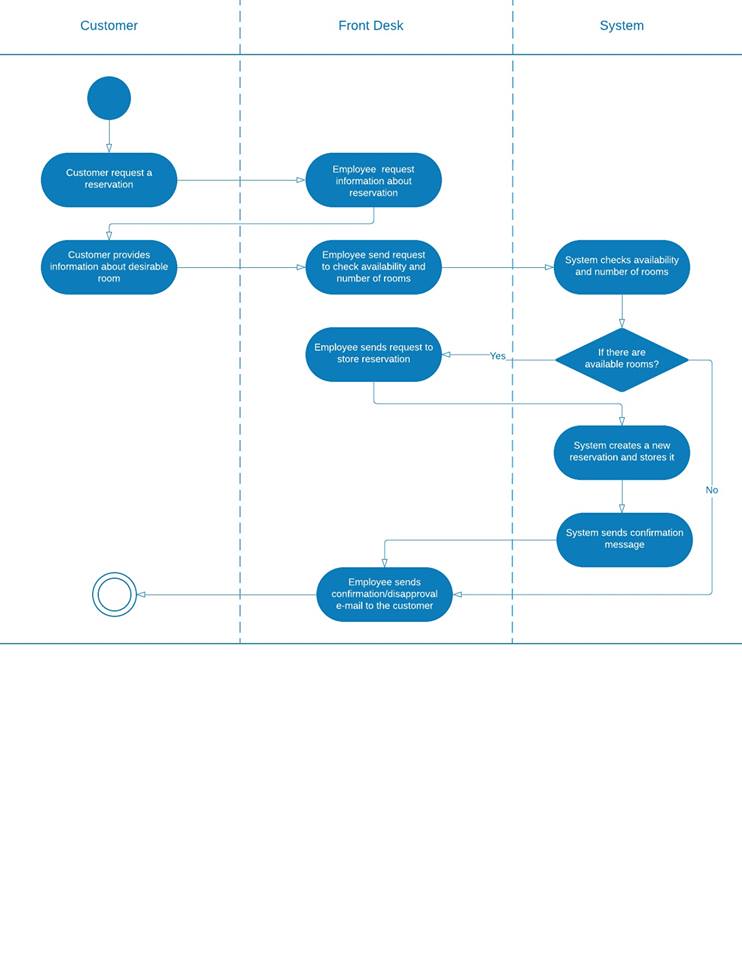


#### Interview Questions:

1. What kind of information about the reservation does the employee need to solicit from the customer?
2. How does the system check for record of reservations in the database?
3. How does the reservation get removed from the database? Is there a sub-system that needs to be set up for this?
4. Is there a way to retrieve cancelled reservations in case the customer changes his/her mind? Or will the customer need to re-book his/her reservation from the start again?

### 2.0 Making Reservations

* 1. Customer requests a reservation.
  2. Employee requests information about reservation.
  3. Customer provides information about desirable room
  4. Employee sends request to check availability and number of rooms
  5. System checks availability and number of rooms
     1. If there are available rooms, proceed to 2.6, if there are no available rooms, proceed to 2.9
  6. Employee sends request to store reservation.
  7. System creates a new reservation and stores it.
  8. Systems sends confirmation message
  9. Employee sends confirmation or disapproval (based on step 2.5) e-mail to the customer

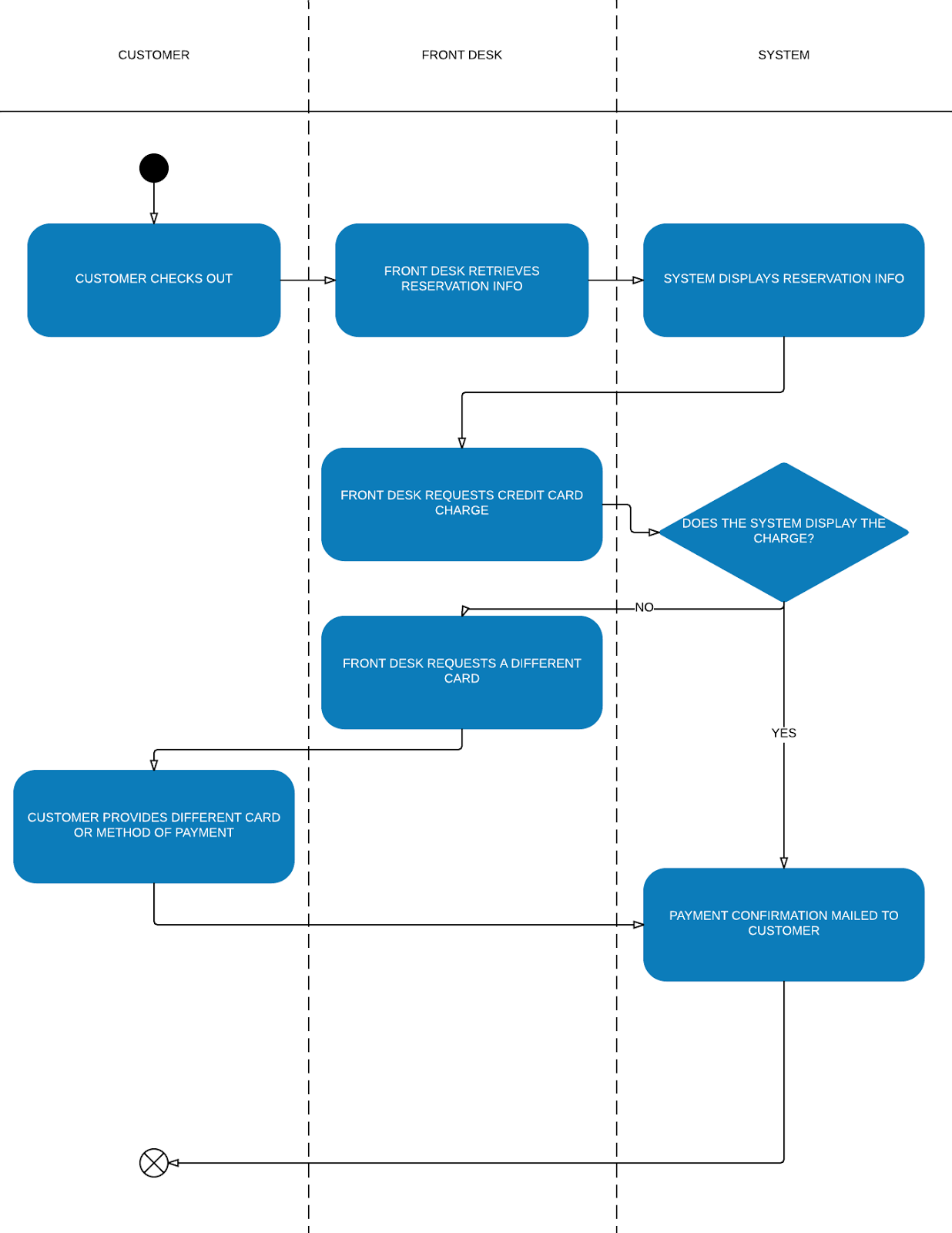


#### Interview questions:

1. What kind of information does the employee request from the customer about the reservation?
2. How does the employee send request to check availability and request to store reservation?
3. How does the system check the number of rooms and checks availability?

### 3.0 Check out / Credit Card Charge

1. Customer requests to check out at the front desk
2. Front desk retrieves reservation info for the specified customer
3. Front desk requests the system to display reservation info
4. Front desk requests credit card charge from the system
5. Does the system display the credit card charge to the front desk? Yes go to 3.8, No go to 3.6
6. Front desk requests a different card from the one provided
7. Customer provides a different card or different method of payment
8. Payment information mailed to customer



#### Interview Questions:

1. What happens if the customer has no other methods of payment on hand?
2. What if the system reservation network is down?
3. How does the system accommodate external charges and/or extended stays?

# Part B: Use Case Model – Domain Classes

## Event Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **The Lucid Palace Information Management System Event Table** | | | | | |
| **Event** | **Trigger** | **Source Actor** | **Use Case** | **Response** | **Destination Actor** |
| 1. Customer wants to cancel a reservation. | Cancellation request (online through the hotel’s website) | Customer | Request cancellation | Reservation record | Customer |
| 2. Front desk wants to cancel a reservation for a customer. | Cancellation request (Customer approaches front desk) | Front Desk | Request Cancellation | Reservation record | Customer |
| 3. Customer confirms to cancel reservation. | Cancellation confirmation. | System | Delete Reservation Record | Deletion notification | Customer |
| 4. Cancellation Update | Reservation cancelled or created. | System | Update room availability | List of room availability | System |
| 5. Customer checks out of hotel | Two hours pass after any customer checks out of hotel | System | Update room availability | List of room availability | System |
| 6. Customer wants to make a reservation. | There was no record of the customer in the system database. | Customer | Create Customer Details | Customer Details | System |
| 7. Customer wants to change his/her details. | Customer detail change request | Customer | Update Customer Details | Customer Details | System |
| 8. Customer requested reservation | Reservation request | System | Make Reservation Record | Notification that a reservation has been made. | Customer |
| 9. Customer wants to make a reservation | Customer makes a reservation request online through the hotel’s website. | Customer | Request Reservation | Reservation Confirmation | Customer |
| 10. Front desk wants to schedule a reservation for a customer. | Customer approaches front desk with reservation request. | Front Desk | Request Reservation | Reservation Confirmation | Front Desk |
| 11. Customer wants to check in | Check in request | Front Desk | Check-in Guest | Checked-in Notification | System |
| 12. Customer wants to check out | Check out request | Front Desk | Check out Guest | Hotel stay summary | Customer |
| 13. Customer checks-out from the hotel | Bill Notice | Customer | Process Payment | Receipt | Customer |
| 14. Customer wants to know if a room is available | Customer inquires if room is available | Customer | Check for available rooms | Room availability | Customer |
| 15. Front desk wants to know if a room is available | Front desk inquires if room is available | Front Desk | Check for available rooms | Room availability | Customer |
| 16. Customer checks in for a reservation | Checked-in Notification | Front Desk | Identify Reservation | Reservation Record | Front Desk |
| 17. Customer checks out | Hotel Stay Summary | Front Desk | Send bill to customer | Bill | Customer |
| 18. Customer checks-in to the hotel | Checked-in status | Front Desk | Collect Credit Card Information | Updated customer info | system |

## UML Use-Case Diagram



## Prioritized, Consolidated List of Use Cases in the Two Essential Workflows

|  |  |
| --- | --- |
| **Priority Table** | |
| **Use Case** | **Priority** |
| Check Room Availability | H |
| Check-in Guest | H |
| Collect Credit Card Information | H |
| Create Customer Details | H |
| Delete Reservation Record | H |
| Identify Reservation | H |
| Make Reservation Record | H |
| Process Payment | H |
| Update room availability | H |
| Check out Guest | M |
| Request cancellation | M |
| Request Reservation | M |
| Send bill to customer | M |
| Update Customer Details | M |

## Brief Use Case Descriptions

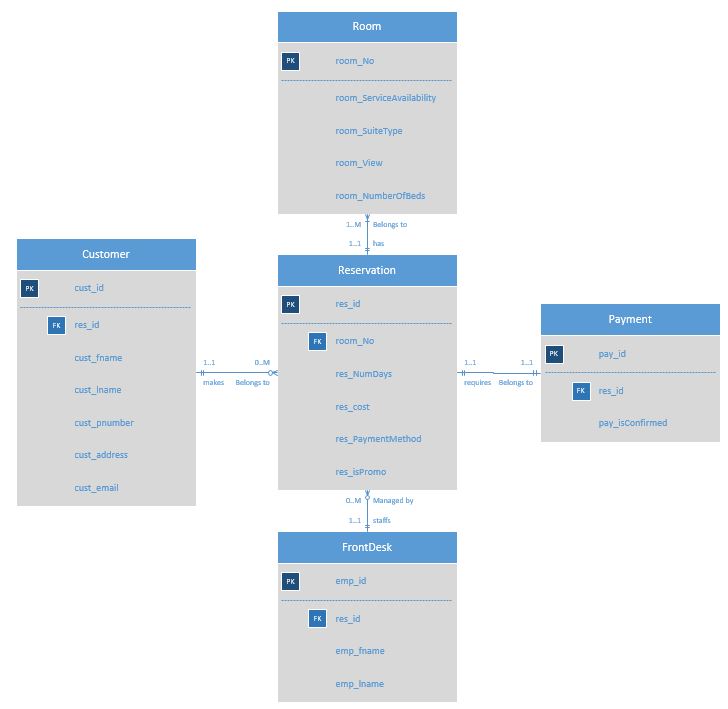
|  |  |
| --- | --- |
| USE CASE | BRIEF DESCRIPTION |
| Request Cancellation | Customer sends a request to cancel reservation either through phone, in person or online. |
| Process Payment | Front desk requests system to process credit card on file. |
| Request Reservation | Customer sends request to hotel to make a reservation either over the phone, in person or online. |
| Check-in Guest | Front Desk requests customer information to locate reservation and records credit card information. |
| Check-out Guest | Front desk sends hotel stay summary to customer and processes payment. |
| Check Room Availability | Customer/Front desk checks availability of room on reservation request. |
| Identify Reservation | Front desk identifies reservation when customer arrives and provides reservation information. |
| Collect Credit Card Info | Front desk collects credit card information from customer when they check-in |
| Send Bill To Customer | System sends bill confirmation email to customer after check out |
| Delete Reservation Record | System deletes reservation record from database |
| Update Room Availability | System updates room availability |
| Make Reservation Record | System creates reservation record in database |
| Create Customer Details | System creates and stores customer information in the database |
| Update Customer Details | Customer changes information such as contact information, address, etc. |

## Domain Class Diagram



# Part C: Procedural Model

## ERD



## Table Structures

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Customer | | | | |
| **Field** | **Type** | **Length** | **Constraint** | **Description** |
| cust\_id | NUMBER | 10 | PK | Unique Customer ID for each customer |
| res\_id | NUMBER | 10 | FK | Reference to Reservation ID in Reservation Table |
| cust\_fname | VARCHAR | 15 | NOT NULL | Customer's First Name |
| cust\_lname | VARCHAR | 30 | NOT NULL | Customer's Last Name |
| cust\_pnumber | NUMBER | 10 | NOT NULL | Customer's Phone Number |
| cust\_address | VARCHAR | 30 | NOT NULL | Customer's Address |
| cust\_email | VARCHAR | 30 |  | Customer's Email |
|  |  |  |  |  |
| Reservation | | | | |
| **Field** | **Type** | **Length** | **Constraint** | **Description** |
| res\_id | NUMBER | 10 | PK | Unique Reservation ID for each reservation |
| room\_No | NUMBER | 4 | FK | Reference to Room Number in Room Table |
| res\_NumDays | NUMBER | 2 | NOT NULL | Number of days for the reservation |
| res\_cost | NUMBER(6,2) | 6 | NOT NULL | Total cost of the reservation ($9999.99)(6,2) |
| res\_PaymentMethod | VARCHAR | 4 | NOT NULL | Type of payment for the reservation |
| res\_isPromo | CHAR | 1 | "Y"||"N",NOT NULL | "Y"es or "N"o for determining if the price is promotional |
|  |  |  |  |  |
| Payment | | | | |
| **Field** | **Type** | **Length** | **Constraint** | **Description** |
| pay\_id | NUMBER | 10 | PK | Unique Payment ID for each payment |
| res\_id | NUMBER | 10 | FK | Reference to Reservation ID in Reservation Table |
| pay\_isConfirmed | CHAR | 1 | "Y"||"N",NOT NULL | "Y"es or "N"o for determining if payment is confirmed |
|  |  |  |  |  |
| Room | | | | |
| **Field** | **Type** | **Length** | **Constraint** | **Description** |
| room\_No | NUMBER | 4 | PK | Unique Room Number for each Room |
| room\_ServiceAvailability | BOOL | 1 | NOT NULL | Determines whether room service is available (True/False) |
| room\_SuiteType | VARCHAR | 6 | NOT NULL | Type of room (ex:"Queen","Double","Penths" |
| room\_View | VARCHAR | 15 |  | Description of view from the room, if applicable |
| room\_NumberOfBeds | NUMBER | 1 | NOT NULL | Number of beds in the room |
|  |  |  |  |  |
| Front Desk | | | | |
| **Field** | **Type** | **Length** | **Constraint** | **Description** |
| emp\_id | NUMBER | 4 | PK | Unique Employee ID for each Employee |
| res\_id | NUMBER | 10 | FK | Reference to Reservation ID in Reservation Table |
| emp\_fname | VARCHAR | 15 | NOT NULL | Employee First Name |
| emp\_lname | VARCHAR | 15 | NOT NULL | Employee Last Name |

|  |  |  |
| --- | --- | --- |
| Relationships | | |
| **Table Name** | **Relationship** | **Table Name2** |
| Reservation | 1:M | Room |
| Customer | 1:M | Reservation |
| Reservation | 1:01 | Payment |
| Reservation | 1:01 | Front Desk |

## System Sequence Diagram

### Making Reservation



### Cancelling Reservation



## State Machine Diagrams

### Payment



### Reservation



### Customer



### Front Desk



### Room



## Gantt Chart

The chart is too large to display. Please see the attached .mpp file.

## Team Meeting Minutes

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Team Meeting Minutes | | | | **Date** | **Participants** | **Discussion/Activities** | | January 20, 2016 | Alexey | Exchanged contact information. | |  | Kevin | Discussed assignment requirements. | |  | Ostap |  | |  | Vinood |  | | January 25, 2016 | Kevin | Discussed assignment requirements. | |  | Ostap |  | |  | Vinood |  | | January 25, 2016 | Kevin | Created Gantz Chart using MS Project. | |  | Ostap | Distributed work to team members. | | January 26, 2016 | Kevin | Discussed system capabilities. | |  | Ostap |  | |  | Vinood |  | | January 27, 2016 | Alexey | Revised Vision document and made | |  | Kevin | sure everyone knew what they had | |  | Ostap | to do for their allcoated assignments. | |  | Vinood |  | | January 28, 2016 | Alexey | Decided which essential workflows | |  | Kevin | our system should cover. | |  | Ostap | Allocated work pertaining to the | |  | Vinood | essential work flows | | January 30, 2016 | Alexey | Revised System Vision Document | |  | Kevin |  | |  | Ostap |  | |  | Vinood |  | | February 2, 2016 | Kevin | Revised and enhanced activity | |  | Ostap | diagrams | |  | Vinood |  | | February 3, 2016 | Kevin | Consulted Professor about differences | |  | Ostap | in software used for activity diagrams | |  | Vinood | Discussed FURPS for three subsystems | | February 10, 2016 | Alexey | Discussed requirements for assignment | |  | Kevin | b. | |  | Ostap | Analyze assignment a to decide on | |  | Vinood | events for event able. | | February 24, 2016 | Alexey | Made rough plan for events. | |  | Kareem | Started on use case diagram | |  | Kevin |  | |  | Ostap |  | | March 10, 2016 | Kevin | Decided on order in which to complete | |  | Ostap | Assignment B. | |  |  | Started working on the event table | |  |  | Confused about difference between | |  |  | Triggers and Events | | March 12, 2016 | Kevin | Finished Use Case Model | |  |  | Finished Event Table | |  |  | Finished Priority Table | | March 14, 2016 | Kevin | Started UML class diagrams | |  | Vinood | Started Brief Descriptions | | March 15, 2016 | Alexey | Edited Use case descriptions | |  | Kevin |  | |  | Ostap |  | |  | Vinood |  | | March 15, 2016 | Kevin | Edited Event table | |  |  | Edited Use Case diagram | |  |  | Edited Priority Table | | March 16, 2016 | Kevin | Organized Team Meeting Minutes | |  |  | Compiled and appended documents to | |  |  | assignment A. | | March 16, 2016 | Alexey | Edited Event table | |  | Kevin | Edited UML Class Diagram | |  | Ostap | Edited Priority Table | |  | Vinood |  | | March 23, 2016 | Kevin | Created ERD | |  | Ostap | Finished ERD | | April 5, 2016 | Kevin | Decided to do SSD on the 3 core activities | |  | Ostap | from our Activity diagrams. | |  | Vinood | Decided to do State machine diagrams | |  |  | on our main classes from ERD | | April 6, 2016 | Ostap | Created State Machine Diagrams for | |  |  | Room, Payment | | April 6, 2016 | Kevin | Created SSD for making reservations and | |  |  | cancelling reservations | | April 6, 2016 | Vinood | Created State Machine Diagrams for | |  |  | Reservation, Customer, Front Desk | | April 6, 2016 | Alexey | Created table Structures | | April 8, 2016 | Alexey | Delegated areas to create ppt slides for | |  | Kevin | to present | |  | Ostap |  | |  | Vinood |  | | April 12, 2016 | Kevin | Finished Gantt Chart | |  |  | Appended all documents into word doc. | |