Temasek Polytechnic
School of Informatics and IT

Diploma in Information Technology (IT)

Software Design Specifications (DS)

Project Particulars

Tutor	Mr Mel Goh
Class	P03
Project Title	Delonix Regia Hotel Management System

Project Team's Particulars

Matric Number	Student Name
1505894D	**Ron Soon JiaJun**
1501312A	**Nicholas Foo**
1505448I	**Low Xin Yi**
1501003F	**Joseph Koh**

Revision History

Date	Version	Description	Author
<18/05/17>	<1.0>	<Completed Distribution of Workload, research on UI Design>	<Ron Soon JiaJun>
<22/05/17>	<1.1>	<Drafted UI Design>	<Ron Soon JiaJun>
<23/05/17>	<1.2>	<Completed UI Design>	<Ron Soon JiaJun>
<24/05/17>	<1.3>	<Touched up UI Design, compiled documents>	<Ron Soon JiaJun

Table of Contents

1. DISTRIBUTION OF WORKLOAD

[Determine which members of the team will be responsible for which areas of design. Individual's responsibilities should be clearly spelt out.]

Design	**Members**
User Interface Design	Ron Soon JiaJun (Leader)
Architecture Design	Nicholas Foo
Database Design	Low Xin Yi
Program Design	Joseph Koh

2. ARCHITECTURE DESIGN

The proposed system architecture that we will be using is an N-tier application design. The software we will be using is Visual Studio 2017. This will help us with Mr Wang’s different systems like

As for the database design these are the difference between the logical and physical database design. I have chosen to use the physical database design for Mr Wang’s company. The reason why I choose physical database design is because Mr Wang’s company is using MySQL and it would be less complicated for the employees and to minimize any data transfer error if we were to convert to a different SQL software like SQL Server 2015.

Logical Database Design

1. Data Requirements is a list of data items that needs to be stored in the system.
2. Normalization. Why? Come out with an Entity Relation Diagram (ERD)
3. Data Dictionary can define all your data elements.

Physical Database Design

1. Select database engine (SQL Server, MySQL, Orcale, etc;)
2. Consider implementation details

When we are creating our software it is a good common practice to always have a set of patern. It is a good practice as this will not only benefit the system, it will help other software helper in future in maintaining the system.

Pattern

- Reusable
- Predictable
- Efficient
- Testable

Presentation tier

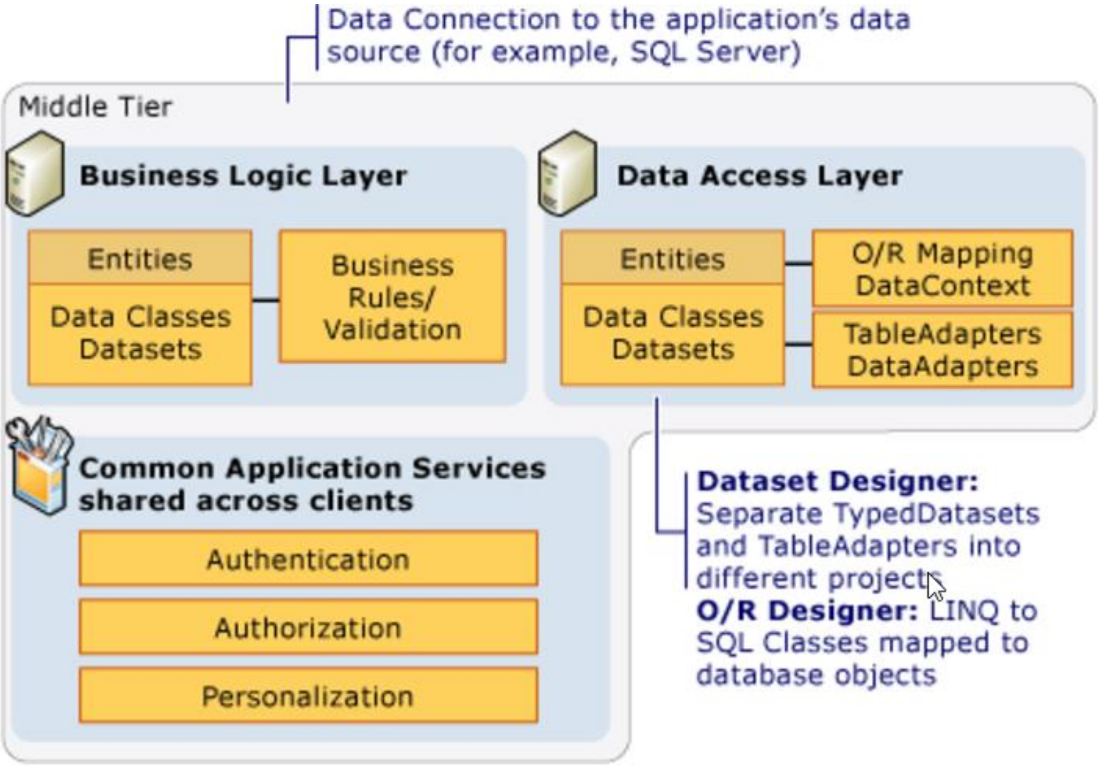
Presentation tier is the the tier which communicates with the application
It often contains other logic such as

- Data binding components, such as Bindingsource and BindingNavigator
- Object representation data such as LINQ to SQL

This tier will be Mr Wang’s application design

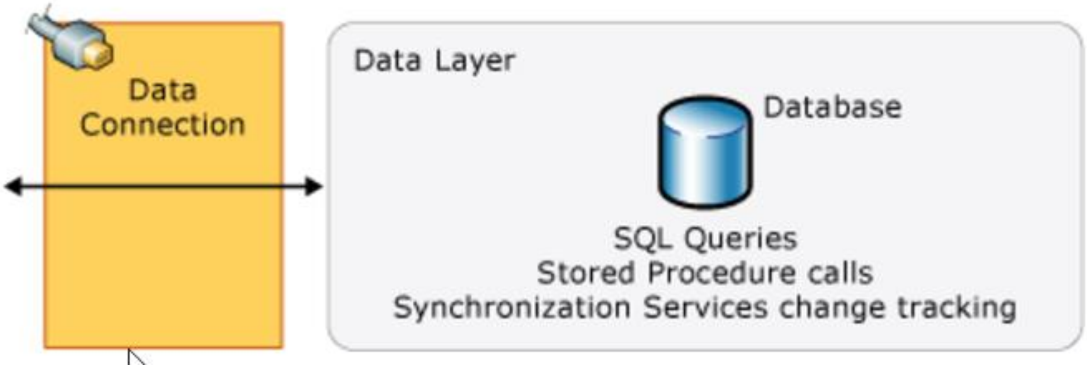
Middle tier

Middle tier is the tier that the presentation tier and data tier use to communicate with each other.


Data Connection to the application's data source (for example, SQL Server)
Middle Tier
Business Logic Layer
Entities
Data Classes Datasets
Business Rules/ Validation
Data Access Layer
Entities
Data Classes Datasets
O/R Mapping DataContext
TableAdapters DataAdapters
Common Application Services shared across clients
Authentication
Authorization
Personalization
Dataset Designer: Separate TypedDatasets and TableAdapters into different projects
O/R Designer: LINQ to SQL Classes mapped to database objects

Data tier

The data tier is basically the server the stores the application's data. For this instances, it will be Mr Wang's database.



3. USER INTERFACE (UI) DESIGN

(All Images are referenced to those in the UI Requirements)

The UI design was developed with a user terminal idea in mind, which is hosted on a webpage/window UI as these forms of UI are relatively common, meaning that it is familiar to a wide user group, therefore reducing difficulty in usage. Furthermore, as the system is fully local and only installed on a single computer as per current design, the usage of a terminal seems to be most efficient as the limited resources can be better optimized for performance due to it only having to run a single task focused program without the need for other programs and add-ons running with it.

The texts and fields of the UI design are moderately big and use a standardized font, allowing for users to easily view, utilize, and navigate through the pages within little difficulty. Learning curve wise, the software is likely to be easily picked up as most of the content in the software is relatively well labelled, majority using simple, straightforward words, as well as instructions provided on certain UI and pages such as:

- The header (Image 2.6.3), which offer a general summary of what each module tab in the menu leads to, such as room information which contains Information, Statuses, Bookings, and Checkout. By dividing the functions of each module into simpler sub-functions, users will be better able to find the links which they require for their tasks
- the home page (Image 2.6.4), which contains a brief navigation overview along with restrictions for pages and functions if any. This provides for users a place to reference to should they be unsure of where to navigate
- buttons in room information (Image 2.6.5), along with the other buttons in the system which are labelled simply to provide better understanding of its purpose
- All input fields are labeled with their respective accepting data, making it easier for users to better figure what each field serves for

As all functions which the system can perform are divided into their own pages, more information and functionalities can be provided on a single page without overcrowding the pages, reducing navigation needs and providing more usage per page. Examples of such pages would be Room Bookings (Image 2.6.8) and Housekeeping (Image 2.6.15.1 and 2.6.15.2), which provide simple but purposeful UI of a more specific group of functionalities. Furthermore, by keeping the data which need not be edited or is automatically set hidden, necessary fields become easier to find and input into without much difficulty. Should a page require certain information, they are usually provided with their respective input fields, such as in New /Edit Booking Pages (Image 2.6.10 and 2.6.11), Housekeeping, and Staff Records, reducing the need for other referential sources.

For this systems' confirmation, error, success, and warning messages, they will be done using pop-up windows as less pages and page space will be required if these simple messages are displayed on a temporary, responsive, and convenient page. Examples of these are as such:

- Incorrect logins on the terminal login page (Image 2.6.1) will lead to a pop-up indicating that the incorrect username/password was used
- Trying to access unauthorized pages and functions such as Housekeeping (2.6.15) as an end user or Accounts (2.6.21) as anything other than an admin will lead to a pop-up indicating access restriction
- Failure to store or retrieve data will lead to an error message to notify users of the errors
- Empty or incorrectly inputted required fields in forms will generate notices to inform users of what is needed
- When submitting forms or deleting data, a pop-up will be generated to confirm the decision, allowing for users to recover from an accidental selection should there ever be sure a case

Layout wise, all content is aligned to the left with even spaces between each field, providing consistency throughout layout, reducing navigational difficulties. The pages are also use standardized fonts and have simple attributes, lowering the chances of confusion while viewing said attributes. Navigation is done using mostly buttons, other than the links provided in the header menu, further improving consistency throughout the system.

For each input field type, they were chosen on how effective a field type would be it. For input types with already available fixed values, dropdown lists are provided for choosing, removing the chance for incorrect inputs. For dates, a calendar is used so as to allow users to have both a reference and a method of selection other than typing, reducing error chances such as incorrect date inputs and unaccepted input formats.

The UI and navigation have been roughly drafted out and is available on this link:
https://epialesnightmare.wixsite.com/uidesign
However, it was designed on 75% zoom and as such has formatting issues in other zoom percentages.

4. PROGRAM DESIGN

There comprises of 3 main software modules in Delonix Regia's Hotel Management System: Room & Availability Module, Housekeeping & Staff Management Module and Reporting Module.

Room & Availability Module

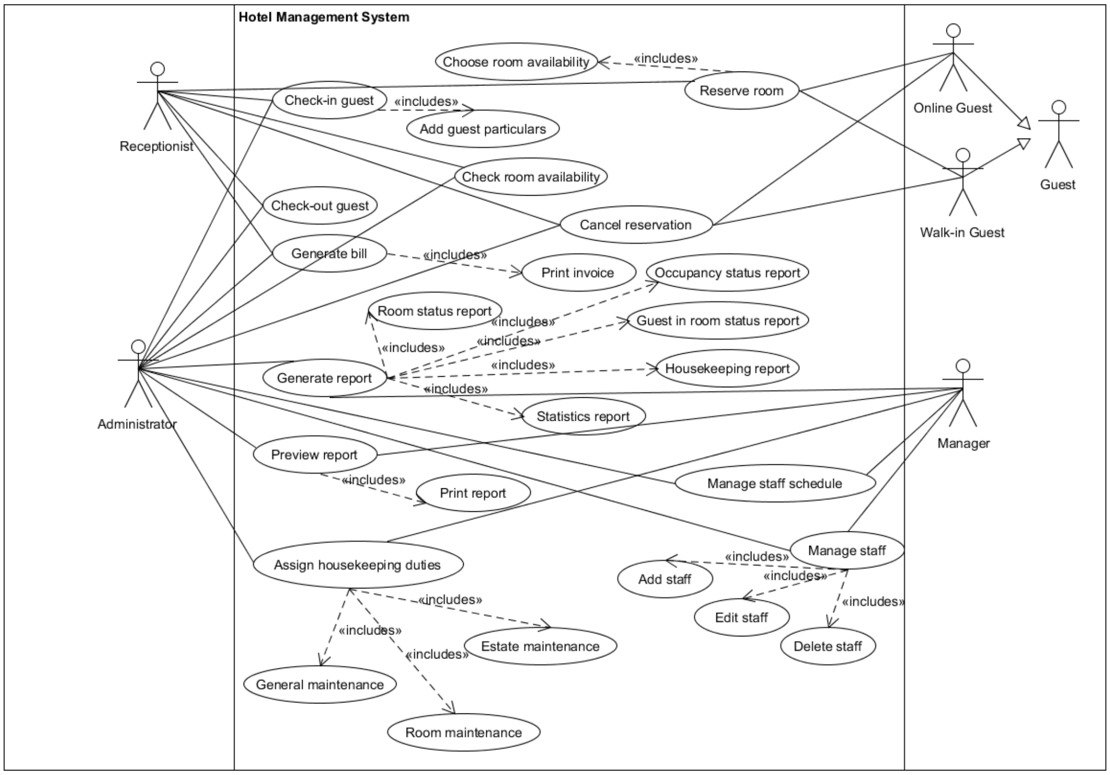
The Room & Availability Module allows the front desk receptionist for checking room availability, reserve room for guest, check-in guest which includes adding in guest particulars, check-out guest, cancel reservation if need be and also generate bill and print the receipt for the guest.

Housekeeping & Staff Management Module

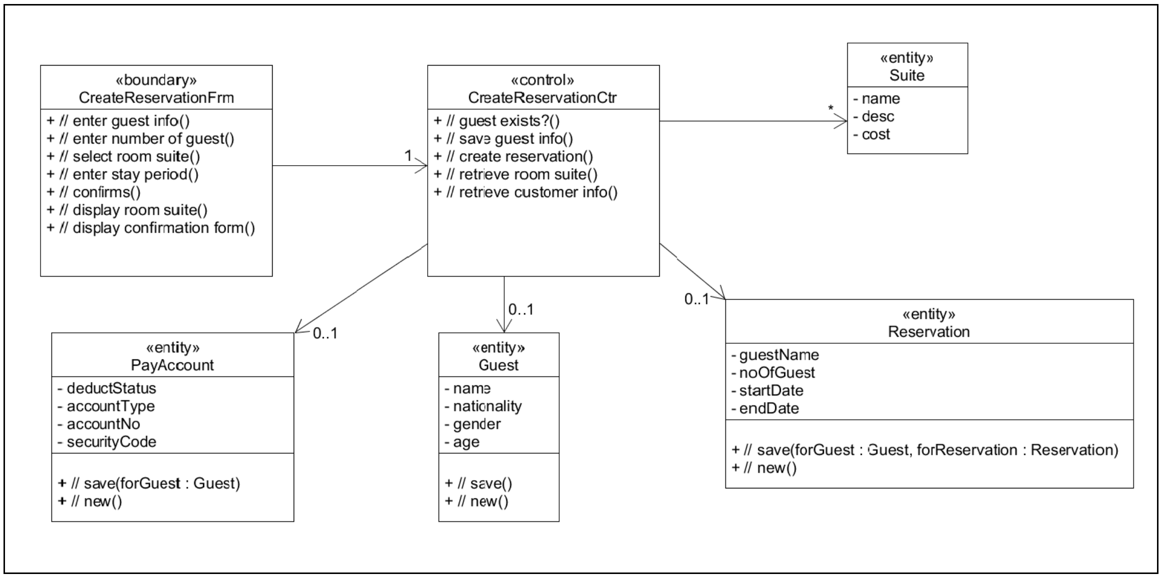
This module is the back end of the hotel management system where the management staff in Delonix Regia manage staff schedule by allocating staff's working shift, managing staff by adding, editing and deleting staff accounts and details if deem necessary.

Reporting Module

This module allows the user to analyze the Hotel's performance on a daily or monthly basis. Administrators can generate 5 different types of report: Room status report, Occupancy status report, Guest in room status report, Housekeeping report and Statistics report. Additionally, the administrator is able to preview the report before printing it.



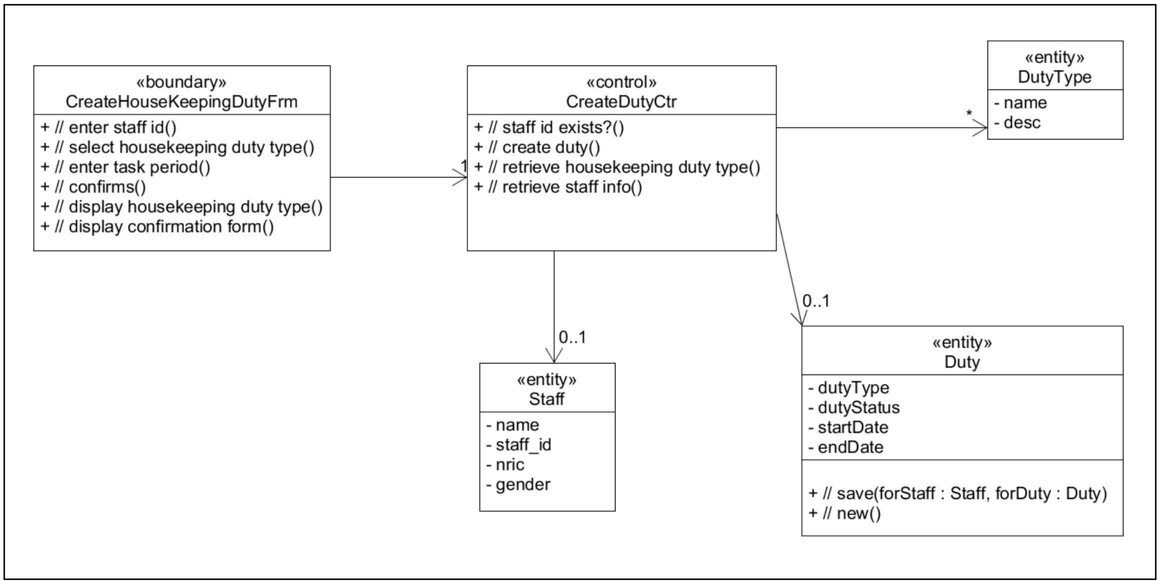
Delonix Regia's Hotel Management System Use Case Diagram



Room Reservation VOPC Diagram

The receptionist staff uses the CreateReservationFrm which communicates with the CreateReservationCtr. For a single run of the reservation use case, the class CreateReservationCtr is associated with:

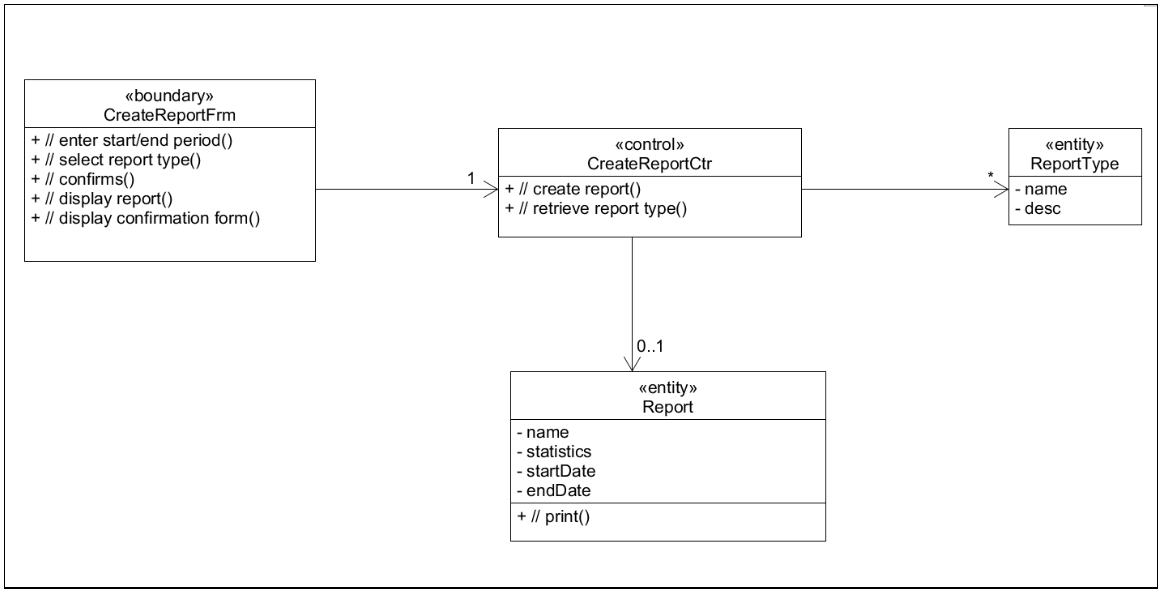
- At most one Guest
- At most one PayAccount
- At most one Reservation
- Many room suite (E.g. Ordinary, Deluxe) as a list is being retrieved



Assigning Housekeeping Duty VOPC Diagram

The manager uses the CreateHouseKeepingDutyFrm which communicates with the CreateDutyCtr. For a single run of assigning housekeeping duty use case, the class CreateDutyCtr is associated with:

- At most one Staff
- At most one Duty type
- Many duty type (E.g. General Maintenance, Room Maintenance & Estate Mainetenance) as a list is being retrieved



Generate Report VOPC Diagram

The manager or administrator uses the CreateReportFrm which communicates with the CreateReportCtr. For a single run of generate report case, the class CreatReportCtr is associated with:

- At most one report
- At most one report type
- Many report type (E.g.Room Status Report, Occupancy Status Report, Guest-in-room Report, Housekeeping Report, Statistics Report) as a list is being retrieved

5. DATABASE DESIGN

I have created Delonix Regia Hotel database system that cover guest, invoice, account, reservation, room, price, promotion, staff and duty roster. When designing the database, I have to first, identify the entity, followed by the attributes stored in each entity, and lastly, the relationship between entities. For example, hotel guest make reservation; a link is created between Guest entity and Reservation entity, so the foreign key in Reservation entity will match the primary key in Guest Entity.

Guest entity: It stores all the guest details such as their name, address and credit card info. Credit card is an optional since not everyone owns a credit card, hence it accepts null value in the database.

Reservation entity: It stores all the reservation details made by the guest also the guest Id, type of room that the guest has select, room number that the guest is allocate to and the staff number who handles the guest's reservation. In case if there is any problem like guest is allocated to an occupied room, the staff will need to be responsible and handle the situation.

Invoice table: It has its own unique id and information of the guest's stay. Invoice is issued to the guest to summarise their period of stay and charges before they make payment

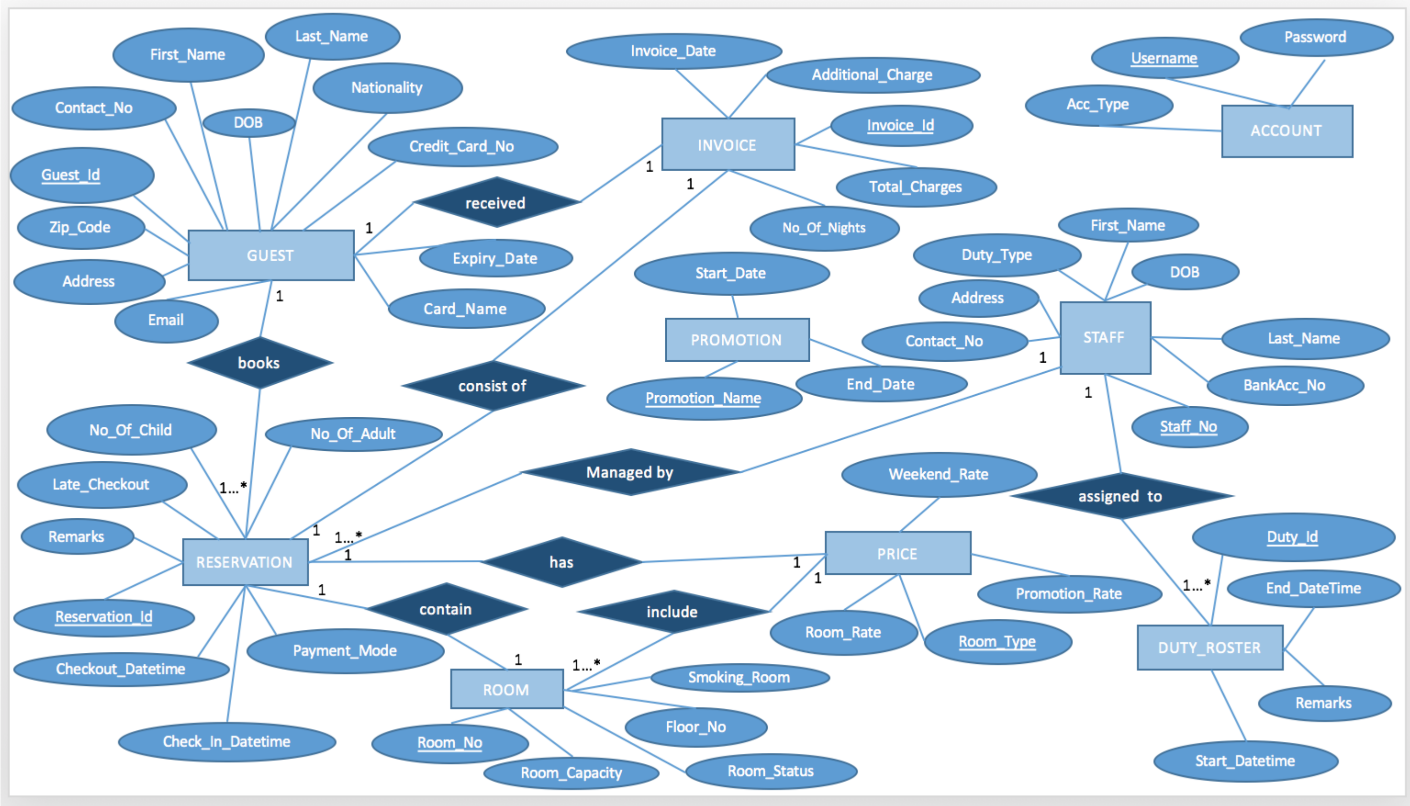
Additional features in the database:

Price entity: Room type is the primary key for this table because different room type has different prices. In a hotel, there are at least 2 – 3 room type which includes Deluxe, Premier and Suite. There is also different in rate for each room type such as the standard room rate, promotion rate and weekend/holiday rate. Promotion rate is usually when guest book a room in advance or special occasion and weekend/holiday rate are usually the peak period like school holidays and public holiday whereby prices will increase.

Promotion entity: It does not have any foreign key but this table stores the name of the promotion and the duration. It is important for Mr. Wang to keep track of the promotions that his hotel offers.

Room entity: Room number is a primary key since every hotel has no duplicate room number and Room type is the foreign key. It stores all the details of a room whether if it's a smoking room, the room level, capacity and status of the room. If the room is vacant, the staff can allocate the arrival guest to the room.

Delonix Regia Hotel Management System ER Diagram



Guest Entity

Field Name	**Data Type**	**Field Length**	**Constrain**	**Description**	**Example**
Guest_Id	VARCHAR	10	PRIMARY KEY	Guest identification number/ Passport number	S9807489J
First_Name	VARCHAR	30	NOT NULL	Guest first name	John
Last_Name	VARCHAR	30	NOT NULL	Guest last name	Tan
Nationality	VARCHAR	20	NOT NULL	Guest nationality	Singaporean
DOB	DATE		NOT NULL	Guest date of birth	1990-05-09
Contact_No	INT	15	NOTNULL	Guest Mobile number	98370924
Address	VARCHAR	45	NOTNULL	Guest home address; street name	Blk 123 Pasir Ris street 12 #09-89
Zip_Code	INT	7	NOT NULL	Postal code number	567890
Email	VARCHAR	40	NOT NULL	Email address	john@hotmail.com
Credit_Card_No	INT	12		Credit card number for payment	5264908767892345
Card_Name	VARCHAR	30		Credit card full name	John Tan
Expiry_Date	CHAR	5		Credit card expiry date	09/20

Reservation Entity

Field Name	**Data Type**	**Field Length**	**Constrain**	**Description**	**Example**
Reservation_Id	CHAR	5	PRIMARY KEY	Unique number for reservation	R1299
Guest_Id	VARCHAR	10	FOREIGN KEY	Guest identification number/ Passport number	S9807489J
Room_No	INT	3	FOREIGN KEY	Unique hotel room number	305
Room_Type	VARCHAR	20	FOREIGN KEY	The types of room available	Deluxe
Staff_No	CHAR	4	FOREIGN KEY	Staff unique number	1458
Check_In_Datetime	DATETIME		NOT NULL	Guest check in date and time	2017-08-09 14:15:12
Checkout_Datetime	DATETIME		NOT NULL	Guest check out date and time	2017-08-13 11:15:12
No_Of_Child	INT	2	NOTNULL	Number of child staying in the room	1
No_Of_Adult	INT	2	NOTNULL	Number of adult staying in the room	2
Late_Checkout	TIME			Required late checkout	13:00:00
Payment_Mode	VARCHAR	11	NOT NULL	Payment made by guest either cash or credit card	Credit Card
Remarks	VARCHAR	45		Additional requirements from guest	Extra bed

Staff Entity

Field Name	**Data Type**	**Field Length**	**Constrain**	**Description**	**Example**
Staff_No	CHAR	4	PRIMARY KEY	Staff unique number	1458
First_Name	VARCHAR	30	NOT NULL	Staff first name	Amy
Last_Name	VARCHAR	30	NOT NULL	Staff last name	Goh
DOB	DATE		NOT NULL	Staff date of birth	1980-09-01
BankAcc_No	INT	15	NOT NULL	Staff bank account number	1408976534
Address	VARCHAR	45	NOTNULL	Staff home address; street name	Blk 145 Tampines street 14 #19-19
Contact_No	INT	8	NOTNULL	Staff contact number	90876753
Duty_Type	VARCHAR	20	NOT NULL	Staff is assigned to 4 different duty type either General Maintenance, Room Maintenance, Estate Maintenance, Security	General Maintenance

Account Entity

Field Name	Data Type	Field Length	Constrain	Description	Example
Username	VARCHAR	30	PRIMARY KEY	Unique username for staff to log in to their account	YP3458
Password	VARCHAR	40	NOT NULL	Staff log in to their account with a password	ABc123
Acc_Type	VARCHAR	20	NOT NULL	Whether the account is for receptionist, administrator or management	Administrator

Room Entity

Field Name	Data Type	Field Length	Constrain	Description	Example
Room_No	INT	3	PRIMARY KEY	Hotel room number	305
Room_Type	VARCHAR	20	FOREIGN KEY	The type of room available in the hotel	Deluxe
Floor_No	INT	1	NOT NULL	The floor number of the room	3
Room_Capacity	INT	2	NOT NULL	The maximum number of people in a room	11
Smoking_Room	VARCHAR	3	NOT NULL	Guest can choose whether their want a smoking or non-smoking room	Yes
Room_Status	VARCHAR	15	NOT NULL	Status of the room whether is Vacant, Booked, Occupied, Cleaning or Unavailable	Vacant

Invoice Entity

Field Name	Data Type	Field Length	Constrain	Description	Example
Invoice_Id	INT	5	PRIMARY KEY	Unique number for invoice issued to guest	12345
Guest_Id	VARCHAR	10	FOREIGN KEY	Guest identification number/ Passport number	S9807489J
Reservation_Id	CHAR	5	FOREIGN KEY	Unique number for reservation	R1299
Invoice_Date	DATE		NOT NULL	Date of invoice issued	2017-06-20
No_Of_Nights	INT	2	NOT NULL	The number of nights that guest is staying	4
Additional_Charge	DECIMAL	4,2		Extra charge for damages or consuming anything from minibar	30.00
Total_Charges	DECIMAL	4,2	NOT NULL	Final amount to be paid by guest including room rate, additional charges.	370.90

Duty Roster Entity

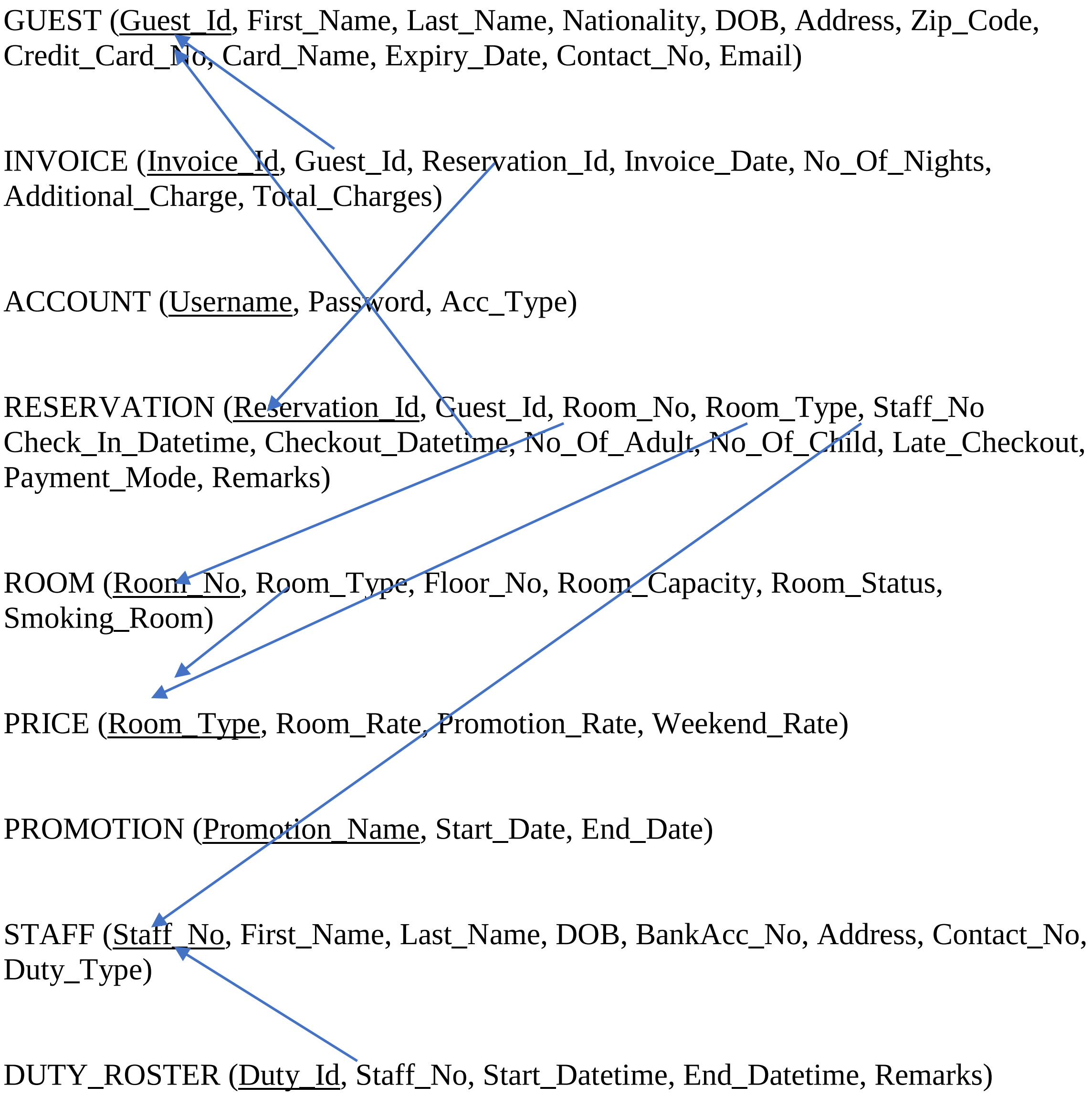
Field Name	**Data Type**	**Field Length**	**Constrain**	**Description**	**Example**
Duty_Roster_Id	INT	3	PRIMARY KEY	Unique number for duty roster	010
Staff_No	INT	4	FOREIGN KEY	Staff unique number	1458
Start_Datetime	DATETIME		NOT NULL	Staff start work date and time	2017-08-13 11:00:00
End_Datetime	DATETIME		NOT NULL	Staff end work date and time	2017-08-13 22:00:00
Remarks	VARCHAR	45		Additional information to take note	Clean room 306 before 12pm

Price Entity

Field Name	**Data Type**	**Field Length**	**Constrain**	**Description**	**Example**
Room_Type	VARCHAR	20	PRIMARY KEY	The type of room available in the hotel	Deluxe
Room_Rate	DECIMAL	3,2	NOT NULL	The standard rate of the room	100.90
Promotion_Rate	DECIMAL	3,2	NOT NULL	The promotion rate of the room decrease to 0.8 especially for advance booking	80.72
Weekend_Rate	DECIMAL	3,2	NOT NULL	The rate of the room increase to 1.2 during weekends or holidays	121.08

Promotion Entity

Field Name	**Data Type**	**Field Length**	**Constrain**	**Description**	**Example**
Promotion_Name	VARCHAR	20	PRIMARY KEY	The name of the promotion for promotion rate	CELEBRATION
Start_Date	DATE		NOT NULL	The first day of the promotion	2017-05-30
End_Date	DATE		NOT NULL	The last date of the promotion	2017-06-09



GUEST (Guest_Id, First_Name, Last_Name, Nationality, DOB, Address, Zip_Code, Credit_Card_No, Card_Name, Expiry_Date, Contact_No, Email)

INVOICE (Invoice_Id, Guest_Id, Reservation_Id, Invoice_Date, No_Of_Nights, Additional_Charge, Total_Charges)

ACCOUNT (Username, Password, Acc_Type)

RESERVATION (Reservation_Id, Guest_Id, Room_No, Room_Type, Staff_No Check_In_Datetime, Checkout_Datetime, No_Of_Adult, No_Of_Child, Late_Checkout, Payment_Mode, Remarks)

ROOM (Room_No, Room_Type, Floor_No, Room_Capacity, Room_Status, Smoking_Room)

PRICE (Room_Type, Room_Rate, Promotion_Rate, Weekend_Rate)

PROMOTION (Promotion_Name, Start_Date, End_Date)

STAFF (Staff_No, First_Name, Last_Name, DOB, BankAcc_No, Address, Contact_No, Duty_Type)

DUTY_ROSTER (Duty_Id, Staff_No, Start_Datetime, End_Datetime, Remarks)