**Temasek Polytechnic**

**School of Informatics and IT**

**Diploma in Information Technology (IT)**

**Project Particulars**

|  |  |
| --- | --- |
| **Tutor** | Mel Goh |
| **Class** | P02 |
| **Project Title** | Delonix Regia Hotel Management System |

**Project Team’s Particulars**

|  |  |
| --- | --- |
| **Matric Number** | **Student Name** |
| **1502698B** | **Gideon Ler** |
| **1506489B** | **Tan Hai Kang** |
| **1502041E** | **Brandon Chew** |
| **1500632C** | **Eugene Sim** |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 16/05/17 | 1.1 | Meeting with Mr Wang, gather info for meeting of minutes | HaiKang, Gideon, Eugene, Brandon |
| 18/05/17 | 1.2 | Completed software design specification | HaiKang,Gideon,Eugene,Brandon |
| 20/05/17 | 1.3 | Completed software requirement specification | HaiKang,Gideon,Eugene,Brandon |
| 21/05/17 | 1.4 | Completed operational and quality requirement | HaiKang,Gideon,Eugene,Brandon |
| 22/05/17 | 1.5 | Finished formatting and finalizing the project | HaiKang,Gideon,Eugene,Brandon |

**Table of Contents**

[**1. Meeting Of Minutes** 1](#_Toc483382055)

[**2. Software Requirement Specification (SRS)** 7](#_Toc483382056)

[**2.1. System Functions** 7](#_Toc483382057)

[**2.2. User Characteristic** 10](#_Toc483382058)

[**2.3. General Constraint** 11](#_Toc483382059)

[**2.4. Functional Requirements** 11](#_Toc483382060)

[**2.5. Non-Functional requirements** 12](#_Toc483382061)

[**2.6. Data Requirements** 13](#_Toc483382062)

[**2.7. User interface requirements** 14](#_Toc483382063)

[**2.8. Interface with Other Systems** 14](#_Toc483382064)

[**3. Operational And Quality Requirement** 16](#_Toc483382065)

[**3.1. Operating Environment** 16](#_Toc483382066)

[**3.2. Development Constraints** 16](#_Toc483382067)

[**3.3. Performance** 17](#_Toc483382068)

[**3.4. Availability** 18](#_Toc483382069)

[**3.5.Security and Access Control Requirements** 18](#_Toc483382070)

[**4. Software Design Specifications (DS)** 20](#_Toc483382071)

[**4.1. Architecture Design** 20](#_Toc483382072)

[**4.2.** **User Interface (UI) Design** 24](#_Toc483382073)

[**4.3.** **Program Design** 32](#_Toc483382074)

[**4.4.** **Database Design** 43](#_Toc483382075)

[**5. References** 45](#_Toc483382076)

[**5.1. Gideon Ler:** 45](#_Toc483382077)

[**5.2. Tan Hai Kang:** 45](#_Toc483382078)

[**5.4. Eugene Sim** 45](#_Toc483382079)

[**6. Peer Evaluation** 46](#_Toc483382080)

[**6.1. Tan Hai Kang** 46](#_Toc483382081)

[**6.2. Gideon Ler** 48](#_Toc483382082)

[**6.3. Brandon Chew** 50](#_Toc483382083)

[**6.4. Eugene Sim** 52](#_Toc483382084)

|  |  |  |  |
| --- | --- | --- | --- |
|  | **1. Meeting Of Minutes** **Delonix Regia Hotel**  Minutes of the meeting with Mr. Wang on functionalities of the Delonix Regia Hotel Management system, held in IT1-6-59 on Friday, 12 May 2017 at 1.30 p.m. | |  |
|  |  | |  |
| Date: | 12/05/2017 | |  |
| Venue: | Lab IT1-6-59 | |  |
|  |  | |  |
| Present: | Brandon Chew (Chairperson)  Gideon Ler (Recording Secretary)  Tan Hai Kang (Member)  Eugene Sim (Member) | |  |
|  |  |  | |
| Absent with apologies: | **No Apologies were received**  No Apologies for absence | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **S/No** | **Item** | | **Action By** |
| **1.** | Meeting started at 1:30 pm | |  |
| **2.**  **3.** | **Main functions of the system**  Mr. Wang shared that he wants 3 main functions. The first function would be the room availability and the booking module. What the room availability function does is it manages the number of accommodation left available for easy identification of rooms that are vacant. The second function would be the housekeeping and staff management module which allows employees to keep track of rooms that are required for cleaning, also to update the housekeepers on rooms that are yet to be clean. The last function would be the report module. Gideon and Hai Kang had volunteered to work on the staff management module, while Brandon and Eugene had volunteered to work on the room availability and booking module. Gideon, Hai Kang,  **Housekeeping and staff management module**  Mr. Wang stated that he wants 2 key features for this module. Firstly, he wants to keep records of his staff (e.g name, dob, bank no., home address, phone number, duty types). For the second feature, he wants to keep track of the 5 staff duty types which are General maintenance, Room maintenance, Estate maintenance and Security. | | Gideon  Hai Kang  Brandon  Eugene |
| **4.** | **System budget**  Mr. Wang states that, the budget required to produce this product would estimate to be $70,000. | |  |
| **5.** | **Users of the system**  Mr. Wang wants the system to be accessed by 3 types of users. The first type users are the end users to be the reception staff at counter to select parts of the reporting module, and full accessibility to the booking room module. The second type of users would be the management users which would have access to all 3 modules. The third users would be the hotel administrators and Mr Wang himself, which they would have access to all 3 modules and user credentials to create new accounts for new staff. | |  |
| **6.** | **Room availability**  Mr. Wang also requested for the software to be able to handle guest check-ins and check-outs. While checking in, Mr. Wang also wants the guest to be able to request for a king or queen size bed.  During the check-in, Mr. Wang wants to obtain the guest’s last and first name respectively and the number of guest that would be staying in the room. The guest is also required to enter their contact number, Email, Billing address, Guest’s house address and country. For payment purposes, Mr. Wang requested for the software to be able to receive payment through credit or cash. If the guest choose credit, the software would prompt the guest to enter their credit card number and expiry date of the card. Additionally, Mr Wang wants the software to be able to accept check in details such as the check in date and time and the desired check out date and time. Lastly, Mr. Wang also requested to have a remark function that guest can request for, for example if the guest request to have a non-smoking room or request for late check-out, the software would display the remarks to the employee to accommodate to the guest. | |  |
| **7.** | **Backup** | |  |
|  | Mr Wang stated that he wants the system to be backed up on every last Sunday of the month at 3 am. He also wishes to keep the information from the system for 5 years, before it would be deleted. | |  |
| **8.** | **Reporting module**  Mr Wang has stated that he requires a total of 5 different report types. These are Room status, Guest report per room, All guest report, Room occupancy report and finally the Housekeeping report.  The Room status report will be capable of listing all rooms in the hotel and its current status. There are only three type of status that it should indicate - Occupied, Vacant or Schedule for cleaning.  The Guest report per room will list out the age group and guests in a specific room.  The All Guest report will list out all the guest in all the rooms at a particular date.  The Room occupancy report will display the statistics on room occupancy at a weekly or monthly basis. This report is to be only accessible by management users and administration users.  Finally, the Housekeeping report will list out all the duties that staff members are allocated in a weekly or monthly basis. This report is to be only accessible by management users and administration users.  Check out from 11am to 12pm  Generate payment invoice for consumption in minibar |  |  |

Meeting ended at 1:40 pm

Recorded by:

Recording Secretary

12 May 2017

Gideon Ler

Vetted by:

Brandon Chew

12 May 2017

**DISTRIBUTION OF WORKLOAD**

|  |  |
| --- | --- |
| **Requirement Gathering** | **Members** |
| System Functions  User Characteristics  Operating Environment  Development Constraint  Reference  Formatting  --------------------------------------------------------------  Architecture Design  Program Design (Guest Management) | Gideon Ler |
| Functional Requirements  Non-functional Requirements  User interface requirements  Interface with Other Systems  Reference  Formatting  --------------------------------------------------------------  User Interface (UI) Design  Program design (Use case diagram) | Hai Kang |
| General Constraint  Assumptions  Performance  Formatting  --------------------------------------------------------------  Program design (Main) | Eugene Sim |
| Data Requirements  Availability  Security and Access Control  Requirements  --------------------------------------------------------------  Database Design | Brandon Chew |

# **2. Software Requirement Specification (SRS)**

## **2.1. System Functions**

1. Login/Register

* Enter username
* Enter password
* Login button

In the login system, all new users would be required to register for a new account. Admins have the job of registering the staff, they would first have to enter in their full names, username, house address, DOB, phone number, house address, duty type and bank account number. Only admins can register the staff for accounts, and staffs get their account password and username from the admins.

2.Home page

* Link to the different management pages
* Exit button
* Booking and room Availability section
* Housekeeping section
* Report section

When users have logged in, they would be first directed to the user page. At the home page, the functions are grouped up 3 different sections. In the booking and room availability section it contains booking and guest management In the Housekeeping section it contains the staff management, and in the report section it contains 5 different reports which are room status, guest report per room, all guest report, room occupancy report and housekeeping report. The exit button would log the user out.

3.Staff management (Housekeeping module)

* Display staff information from database as table form
* CRUD buttons for functions
* Room maintenance
* Security
* Estate maintenance
* General maintenance

The staff’s DOB, first and last names, bank account number, phone number, house address and duty types would be displayed. The different types of staff would be allocated to their different duty types which include the room maintenance, estate maintenance, general hotel maintenance and security. The administrator would be able to add, edit and remove staff.

4.Room Availability Management (Booking and room availability module)

* Display list of rooms
* Display room details
* Display occupied or unoccupied rooms

The room availability management would be inside the page called booking. When reception staffs want to book a room for their guests, they would first enter this page, which is only viewable for the reception staff and administrators. The list of occupied and unoccupied rooms would be in a display, showing the room details such as the check in and checkout time and dates, as well as the room type. The room availability management will be linked to the booking management, and if the rooms are occupied it would show the guest details.

5.Booking management (Booking and room availability module)

* Display room details
* CRUD functions
* Display hotel check in/out details
* Link to guest management

For the booking management, guest details such as contact number, email, billing address, house address, NRIC and country would be keyed into the database. The guest check in and check out date is also keyed into the database. Additionally, guests have the option of having a smoking or nonsmoking room which would be added in the additional remarks. Only admins and reception staffs can perform CRUD functions.

6.Guest Management (Booking and room availability module)

* CRUD function
* Display guest information in table form

Once the guests have already booked their rooms, their details would be displayed in the guest management. Showing the room, they are staying in and their personal details. Only admins and the reception staffs are able to perform the CRUD functions in this section.

7.Hotel Report (Report module)

* Room Status
* Guest Report per room
* All guest report
* Room occupancy report
* Housekeeping report
* CRUD function

This section of the system would showcase the list of reports in the hotel. The Room status report will list our all rooms in the hotel and its current status. There are only three type of status that it is indicated for cleaning, which are occupied, vacant or schedule for cleaning. The Guest report per room will list out the age group and guests staying in the specific room, as well as the room charges that guests would have to pay. The All Guest report will list out all the guest in all the rooms at a date. The Room occupancy report will display the statistics on room occupancy at a weekly or monthly basis. This report is to be only accessible by management users and administration users. Finally, the Housekeeping report will list out all the duties that staff members are allocated in a weekly, monthly and daily basis. The report only accessible to reception staffs and administrators, which they are also able to perform CRUD functions on.

8.FAQ function

* Page which shows FAQ when clicked

If staff are unsure of how to use the system or they do not understand what some functionalities or terms in the system, they would be able to click on the help button which shows the definitions of the pages in the system as well as their functionalities.

## **2.2. User Characteristic**

End Users (Reception Staff)

Roles-

* In charge of entering guest information into the booking and guest management within the system.
* Only able to do CRUD functions for booking module
* Full access to room availability and booking module
* Access the reporting module

User characteristics-

* Have access to read certain important reports in the reporting module

Management users

Roles-

* Unable to perform any CRUD functions.
* Have access to all modules of the system

User characteristics

* Only able to view the modules within the system

Administrators (Mr Wang)

Roles-

* Access to all parts of the system
* Access to all the modules of the system
* Able to perform CRUD functions for all modules
* User account and login creation module

User characteristics-

* Able to create accounts for new users of the system, able to perform CRUD functions to all parts of the system.

## **2.3. General Constraint**

* Scalability
  + Software able to maintain its function regardless of size.
* Adaptability
  + Software being able to adapt to the hotel changes.
* All functions to be done in 3 weeks’ time.

## **2.4. Functional Requirements**

Functional requirements are fundamental actions that the system must perform. There are three main categories functional requirements for hotel management system, booking and room availability module, housekeeping module and lastly reporting module.

The following are functional requirements of the system: -

* **Booking and room availability module**

1. The system shall record the customer information such as first name, last name, number of people, contact number, email, postal code, billing address, payment mode.
2. The system shall list the booking record, display the guest’s information and room’s information by doing a search on the hotel room number.
3. The system shall allow the guest and room information to be changed.
4. The system shall generate a confirmation id and hotel room number after a reservation is made.
5. The system shall generate a payment invoice after the guest has checkout, indicate guest’s information, check-in date, checkout date, number of night stayed, room rate, additional cost and payment mode by the hotel room number.
6. The system shall display the number of room which are available in the hotel.

* **Housekeeping module**

1. The system shall store the hotel staff record which allow management user and hotel administrator to list the staff record by searching the staff name or job title.
2. The system shall allow the staff record to be changed.

* **Reporting module**

1. The system shall be able to generate the following report: -

* **Room status report** to list all the room in the hotel**,** such as if the room are available or occupied and is the room vacant but scheduled for cleaning.
* **Guests report** to list all the guest’s information staying in a particular room.
* **Housekeeping report** tolist the duties the staff are allocated and generate housekeeping schedule on daily, monthly, weekly, basic.
* **Occupancy statistics report** provide statistics of room occupancy on daily, weekly, monthly and yearly period.
* **All guest report** to list all guests in all rooms at any point of time.

## **2.5. Non-Functional requirements**

Non-functional requirement is a requirement that specifies how the system should work rather than what the system should do.

The following are non-functional requirement of HMS: -

* **Security requirements:**
* The system must have authorization level to ensure that certain feature of the system is only accessible by the administrator.
* The staff information and customer information must be safe and secure.
* **Safety requirements:**
* The system must authenticate user before they can access the system.
* The database must able to backed up.
* **Usability**
* The system must be easy to understand and learned by its intended users.

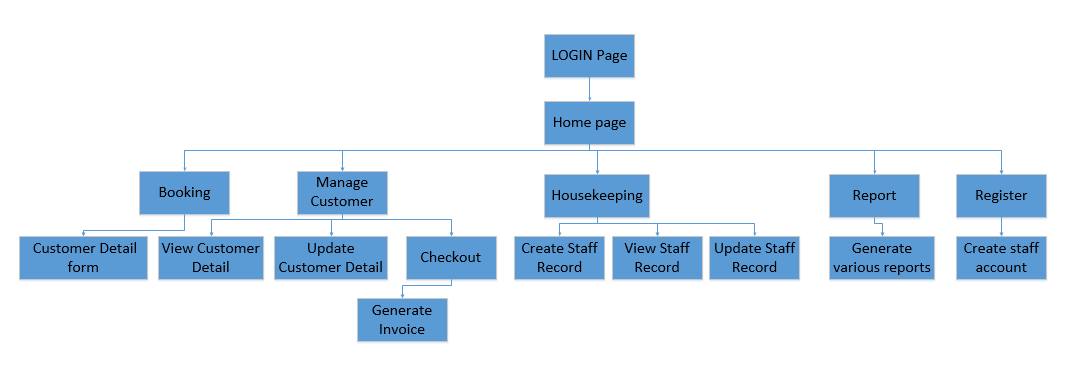
## **2.6. Data Requirements**

These are the following data that is required in the system:

* Type of rooms in the hotel
* Department details
  + For example: Type of departments, No. of staffs in each
* Room’s details
  + For example: Number of rooms, rooms sizes, maximum people in the room
* Staff’s details
  + For example: Staff’s name, Staff’s NRIC, Staff’s role
* Guest’s details
  + For example: Guest’s NRIC, Guest’s name, Guest’s country, Guest’s room number
* Info about work distribution per staff

[Data that is stored within the system e.g. Info about books in library, member’s details]

## **2.7. User interface requirements**

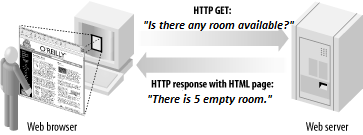


(2.8.1. flow of UI*)*

## **2.8. Interface with Other Systems**

One of the requirements is to eliminate the need to manually check the system on room availability and email to reply the customers. Hence, to allow potential guest to enquiry about room availability online we can integrate Window form with Web Services.

Because Web services feature is based on standard protocols, your Web service applications can communicate with a wide variety of implementations, platforms, and devices as it is not tied to any one Operating system or programming language. Web Services uses the Internet for direct application-to-application information exchange by using a combination of open protocols and standards, mostly Extensible Markup Language(XML), Hypertext Markup Language(HTML), Hypertext Transfer Protocol(HTTP), Universal Description, Discovery and Integration(UDDI), Web Services Description Language(WSDL) and Simple Object Access Protocol(SOAP).



(2.8.1. illustrates a *human-centric Web)*

The diagram is an example, Delonix Regia Hotel website could provide information, enabling customers to check on room availability.

To check on the room availability, a cust**o**mer logs into the company web site via a web browser and receives the results as an HTML page.

However, there are some potential security architecture and design issues for Web service, but not limited to: -

1. **Authentication**

* Identity spoofing
* Password cracking
* Unauthorized access

1. **Message encryption**

* Sniffing of confidential data off the network
* Stealing users’ credentials or session information

1. **Authorization**

* Access to confidential or restricted data
* Tampering
* Execution of unauthorized operations

**2.9. Assumptions**

* Completing every single function before the deadline.
* Any software version is compatible with each other.
* Assuming Visual Studio 2015 can run on Windows 8 and 10 environment

# **3. Operational And Quality Requirement**

## **3.1. Operating Environment**

System Operating Environment

* Windows 10
* Windows 8

The system is a windows form and once it has been developed, it would be launched from Visual Studio’s 2015 and it is able to run on both windows 10 and windows 8 environment.

## **3.2. Development Constraints**

Due to lack of time and budget, here are some of the following constraints that might occur:

System performance-

* Might have unforeseen bugs and errors
* Might run slowly
* 1GB RAM on current computer
* 160GB Hard disk

Platforms unable to be run on-

* Apple OS
* Linux

Schedule-

* Limited meetup days
* Project to be completed in 2 weeks

Additional constraints-

* Room availability online
* Budget of only $70,000
* Reporting module able to export to Microsoft excel
* Might have to get additional computers due to using 3-tier architecture

## **3.3. Performance**

Login - 0.1 to 1 second response time would be ideal during average time of operation. The acceptable response time for peak hours would be 1 to 5 seconds. We chose this response time as there would be many employees logging into the software during peak hours which would increase the waiting time of the system.

Homepage - Since the homepage appear after the Login, the average response time for the system would also be 0.1 to 1 second and 1 to 5 seconds response time during peak hours.

Staff management - As Staff would consistently be needing to use the software, the acceptable response time has to be 0.1 to 0.5 seconds in order to accommodate with the fast pace work environment. During peak hour, the work pace would be even faster. However traffic would also be increased which therefore still requiring the system to have a response time of less than 1 second.

Guest management - Customer’s satisfactory is highly important to the company’s reputation. An average response time of 0.1 seconds is needed when obtaining information on a particular guest and at most a 0.5 second response time during peak hour.

Housekeeping management - As customers returning to their room is unpredictable, it is advisable to have an average system response time of 0.1 seconds to reduce time wastage on room cleaning and a 0.5 second response time during peak hour as traffic would be increased during that period.

Room availability - Assigning a room is as of high importance to the company too. If the hotel took too long to obtain information on vacant room, customers might be unsatisfied and might leave a bad rating or even looking for a different hotel to stay in. Thus, the average response time of when obtaining room vacancy should be 0.1 seconds as the process would be quite quick. However, during peak hours, the system response time might be congested as of that time thus, allowing a 0.5 second response time.

FAQ function - Since FAQ does not have a live chat function, the average system response time require does not need to be fast, thus the ideal response time would be 0.1 to 2 second. However, during peak hour there would bound to have an increased amount of users online which is recommended to have a system response time of 1 second to 3 second

## **3.4. Availability**

The system is required to run at 8am to 8pm everyday unless the system is on maintenance. This is to ensure that the system will be available to use at any point of time such that staffs will be able to check and use the system at appropriate timing and guests are able to check in and check out slightly earlier if required so.

The system will have a monthly maintenance every last Sunday at 3am. Not only that, there will be a data backup of data during the past 5 years. Data backup will be scheduled together with the maintenance to ensure that atleast data within a month will be backed up for safety.

## **3.5.Security and Access Control Requirements**

**Staff members**

Access to:

Login page

Booking module

Report module: Room status, Guest report per room, All guest report

Customer management page

Denied Access to:

Staff management page

Report module: Room occupancy report, Housekeeping report.

Ability to start system maintenance

**Management staff members**

Access to:

Login page

Booking module

Report module: Room status, Guest report per room, All guest report, Room occupancy report, Housekeeping report.

Customer management page

Staff management page

Denied Access to:

Ability to start system maintenance

**Administrative members**

Access to:

Login page

Booking module

Report module: Room status, Guest report per room, all guest report, Room occupancy report, Housekeeping report.

Customer management page

Staff management page

Ability to start system maintenance

Denied Access to:

Ability to Edit or Delete customer or staff management information

Security Requirements

Data encryption

Authentication

# **4. Software Design Specifications (DS)**

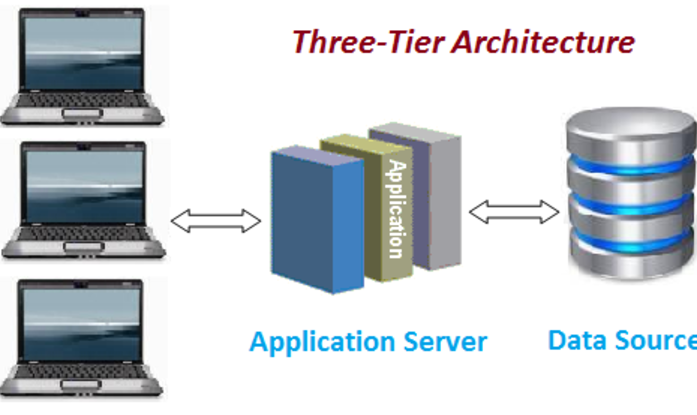
## **4.1. Architecture Design**

The proposed architecture diagram that we recommend for Mr. Wang would be the three-tier architecture diagram. A three-tier diagram is a diagram comprising of 3 tiers, the presentation, business and data tier.

The first tier is also known as the presentation layer, is meant for the user’s usage. It is responsible for the user interaction and system presentation. In this layer, data would be presented to the users, and so is input which can be taken from the users. Most tier one applications are created using html or windows form. An example of first tiers that is used in the real world include Mozilla Firefox.

The second tier is also known as the application logic layer, and it is responsible for managing the business logic of the application. Business logic includes validation of data, calculations, data insertion, analysis, etc. This acts as an interface between Client layer and the data access layer. The second tier also helps to make communication faster between client and data layer. Most tier two applications are created using ASP.net with C# programming language. An example of the second tier would include Internet Information Services (IIS).

The third tier also known as the data layer is responsible for the application’s database and data access. Data Access Layer contains methods to connect with database and to perform CRUD functions on data from database based on the input data. An example of a third tier would include MySQL server.



(Figure 4.1.1 illustrates how a 3-tier diagram functions)

In the diagram above, it illustrates how the three tiers are inter connected. Tier 1 applications are only able to be connected to tier 2 applications, and tier 2 are only able to connect with tier 3 applications. In Mr. Wang’s case, when the admin is trying to add a new staff into the system, the information is being transferred to the second tier, which performs the task at the database by updating it. At tier 2, it is the “business logic layer”, and it decides whether the user has the right to add in a new staff. If the user is using an admin account, then it would allow the update to happen. Finally, at the tier 3 it is the database layer and it stores all the updated info from the second tier.

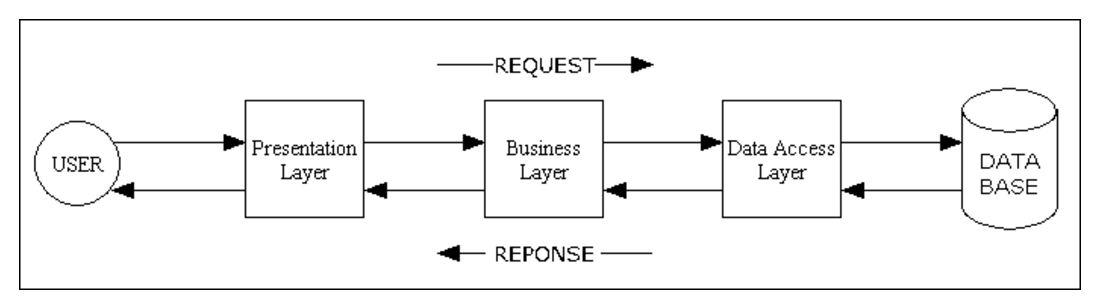
Advantages of 3-tier includes:

* **High performance**- Lightweight and persistent objects
* **Scalability** – System can be enhanced since a separate connection from each client is not required making it loosely coupled, whereas connections from few application servers are sufficient. Thus, any future upgrades to the system would be much easier.
* **Performance** – Because first tier is able to cache requests, network utilization is minimized, and the load is reduced on the Application and Data tiers.
* **High flexibility** in deployment platform and configuration
* **Improved Security** – Implementation of several tiers would help enhance the data security since clients do not have direct access to the database, it provides less risk and confliction with unauthorized data.

3 tier architecture in action:

Splitting the codes of a system into 3 parts is bad if the code within each tier does not behave in a certain way. Hence here are some rules that must be followed-

* The code for each layer must be contained with separate files which can be maintained separately.
* Each layer may only contain codes which belong to that layer. For example, business logic can only reside in the business layer, presentation logic in the presentation layer, and data access logic in the data access layer separately.
* The presentation layer can only receive requests from an outside user.
* The presentation layer can only send requests to, and receive responses from the business layer. It cannot have direct access to either the database or the Data Access layer.
* The Business layer can only receive requests from, and return responses to, the Presentation layer.
* The Business layer can only send requests to, and receive responses from, the data Access layer. However, it is unable to access the database directly.
* The data access layer can only receive requests from, and return responses to, the business layer. It is unable to issue any requests to anything other than the database management system which it supports.
* Each layer should be totally unaware of the inner workings of the other layers. For example, in the business layer, it must not know about the inner workings of the data access object. It must also not know or care how its data will be handled. It should not process its data differently based on what the receiving component will do with that data. The presentation layer may take the data and construct an HTML document, windows form or process it in some other way, but it has to be totally irrelevant to the Business layer.

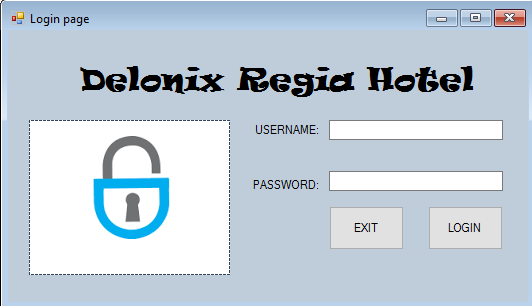


(Figure 4.1.2 shows the different response times in the 3 tier)

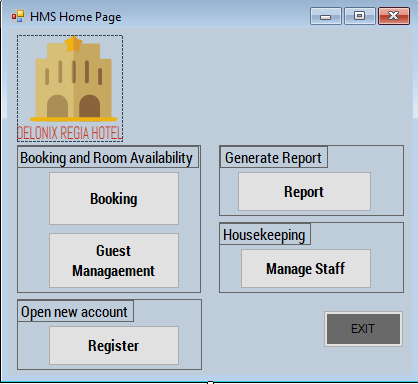
The diagram above shows how the data is transferred between each of the layers. Each of the layers also require different sets of skills to use. For example, in the presentation layer, the business layer requires skills in programming, so that the system would be able to process the business rules. And finally, the data access area requires the developer to have rules in SQL and skills in database designing.

Hence, a three-tier diagram would help benefit Mr. Wang’s hotel system as firstly it is well secured. If hackers would want to attack the system, they would need to bypass all 3 tiers. Secondly a 3 tier is scalable and expandable, meaning if Mr. Wang’s hotel staff ever expands in the future and more people would be using the system, it would still be able to support a large quantity of users. Thirdly the system would be less prone to breaking down since, leaving Mr. Wang to be at ease. However, having a 3-tier architecture means that Mr. Wang would have to get 3 computers in order to implement the tiers.

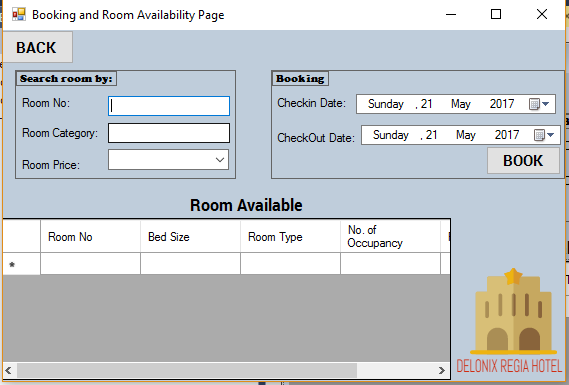
## **4.2.** **User Interface (UI) Design**



The system, should be secure there is a need for authentication and giving the right authorization level to the right user. The form consists of a username textbox which will be use to capture the user input and a password textbox that are password protected.



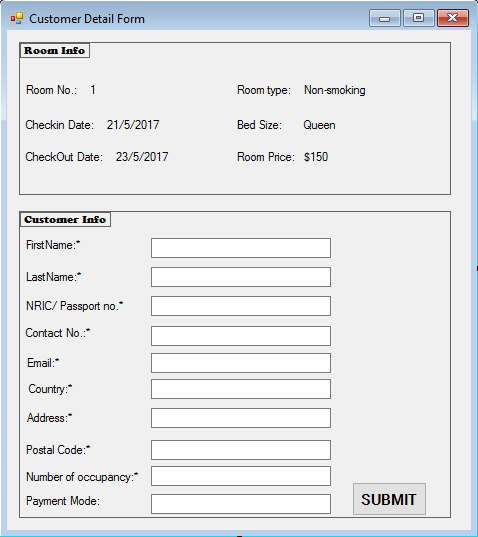
This home interface is designed to be easy to understand and learned by its intended users. The UI uses minimum UI controls so that user can navigate the system with ease. It consist of booking and guest management button, this 2 button is mostly use by receptionist to capture and manage guest detail. Report and manage staff button are usually used by management user and the hotel administrator.



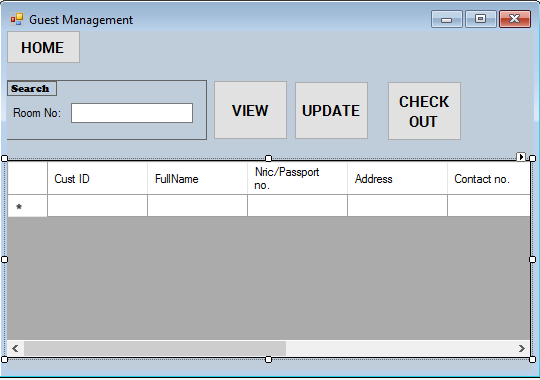
As the booking and room availability page require the user to look for room availability, there are a search function which allow the user to search by room no., room category or room price. A search will allow the user to choose how they want to look for the room detail, it consist of 2 textboxes and 1 combo box search. The combo contains the hotel room rate which allow the user to choose the desired room.

For the search result, we decided to use DataGridView form. A DataGridView form can list data from the database to the UI. This would allow the user to check the room availability.

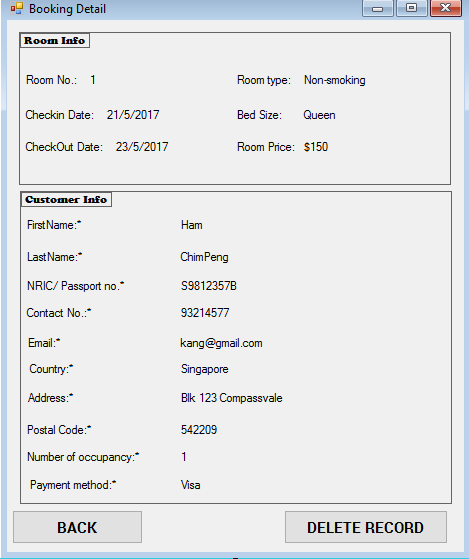
For the booking of room, the user have to select the room available from the DataGridView form and pick the checkin date and checkout date from the date and time picker. This will allow the system to record the number of day the user are booking the room. Lastly, the book button will create the room selected and process to Customer detail form.



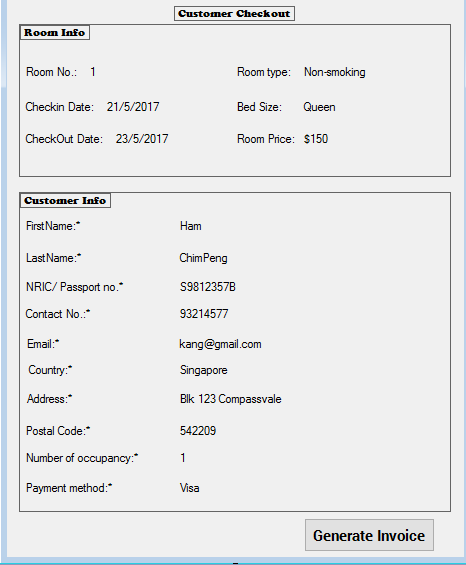
Guest detail page is where the user key in the customer information. The information will then be sent and stored in the db.



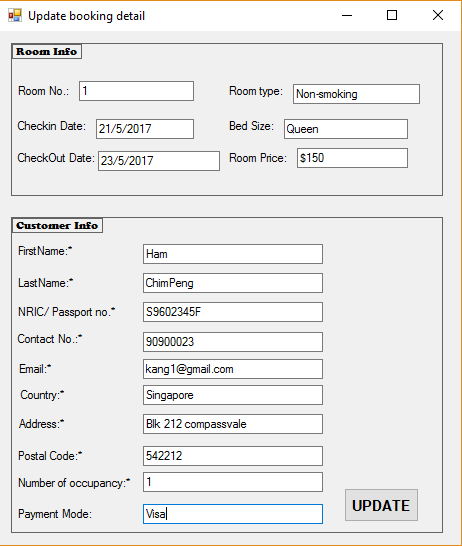
Guest management allow the user to manage the guest information and room detail such as changing of room or adding number of occupancy.



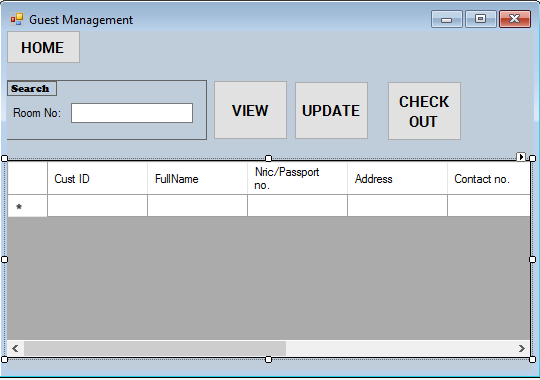
The booking detail allow the user to get an overview of the customer information and room information. It consists of the function to delete the record.



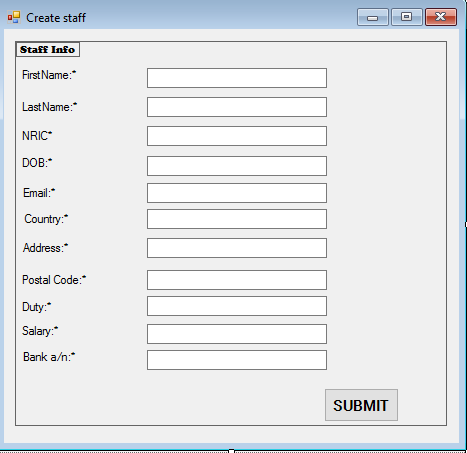
Allow the user to checkout the guest and generate an invoice which will be sent to the customer.



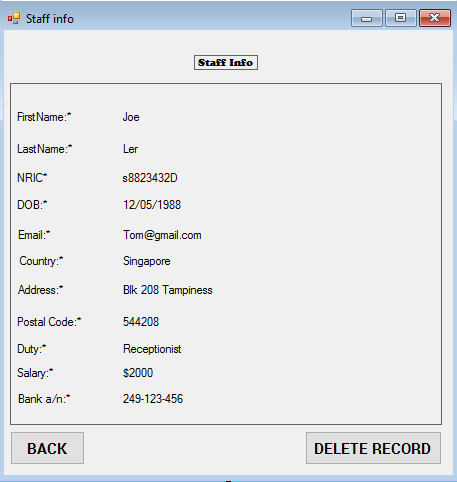
Allow the user to update the customer info and room info.



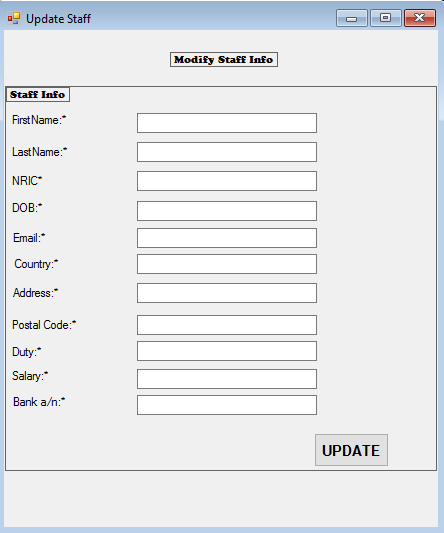
The Staff management page is for creating new staff, modify and delete staff information from the db. It has a search function where the user can look for a particular staff by the name or it department.



This page allows the user to create staff information.

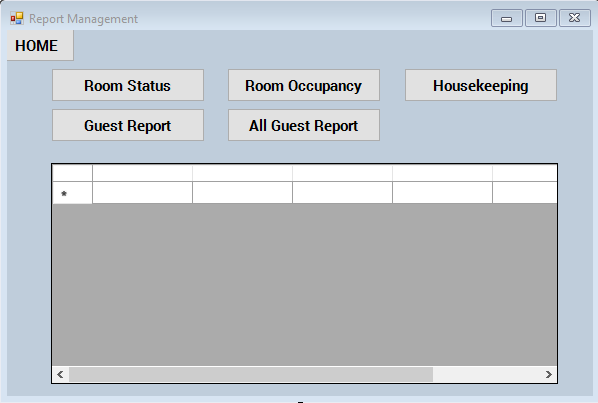


Provide an overview of staff information and the deletion of staff record.



(4.2.9. illustration of Staff info)

Allow the administrator user to modify the staff information. Example, changing of bank account number.



The report management page allows the user to view and generate various report. This page consist of 5 different button to view various reports. We used DataGridView Form to retrieve data from the db.

After careful consideration and thought, I had decided to use Window Forms to design the UI. This is because a Hotel Management System should be highly intensive and take full advantage of the system functionality.

Hence, Window Forms can perform better as it run on client machine which make exchanging of data slower. On the other hand, Web forms run on Web server which make the processing slower than Window Forms

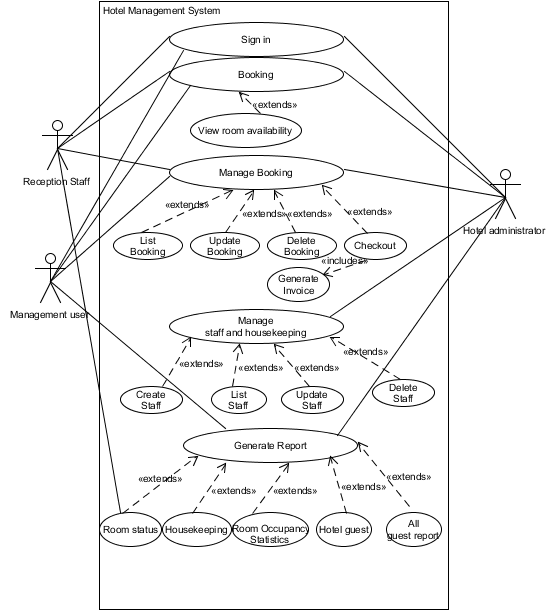
In addition, a Windows application that uses Windows Forms is built around a Windows framework, it has access to system resources on the client computer, including local files, the Windows registry, the printer, and so on. This level of access can be restricted to eliminate any security risks or potential problems that arise from unwanted access.

The following are some justification pertaining to my selection: -

* **Responsiveness** as window form run entirely on window client, it is able to provide a highly-intensive response speed and is more suitable for a hotel management system that require a high degree of interactivity.
* **Security**, window forms use granular permission to protect sensitive information, which allows careful exposure of functionality while retaining the security. For example, the file permission, reception staff should not be granted permission to access the housekeeping or room occupancy report. While the administrator can access to all files.
* **Access to local resources** as the system run locally which allow a complete access to local computer resources.

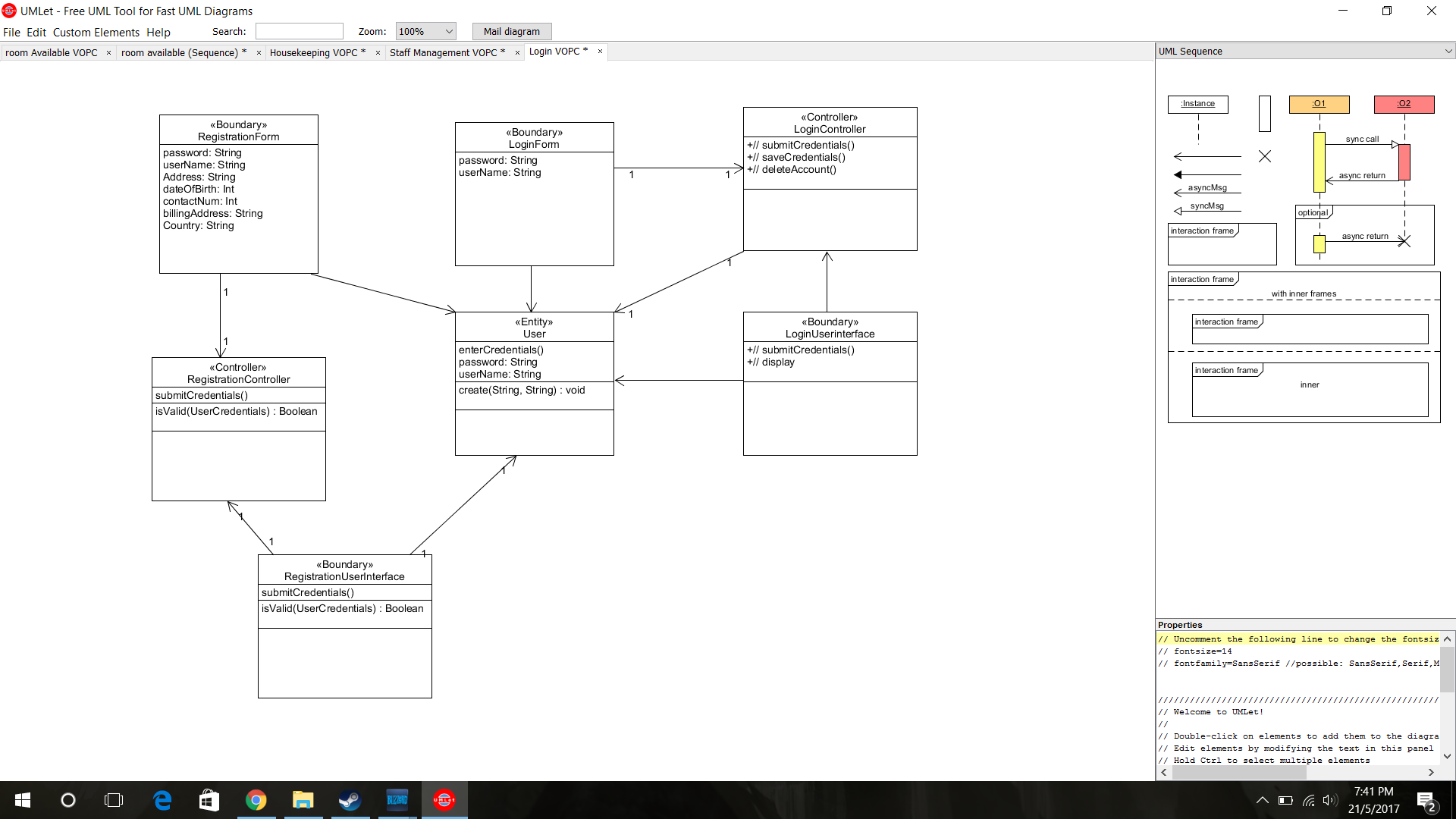
## **4.3.** **Program Design**

**Use Case Diagram**



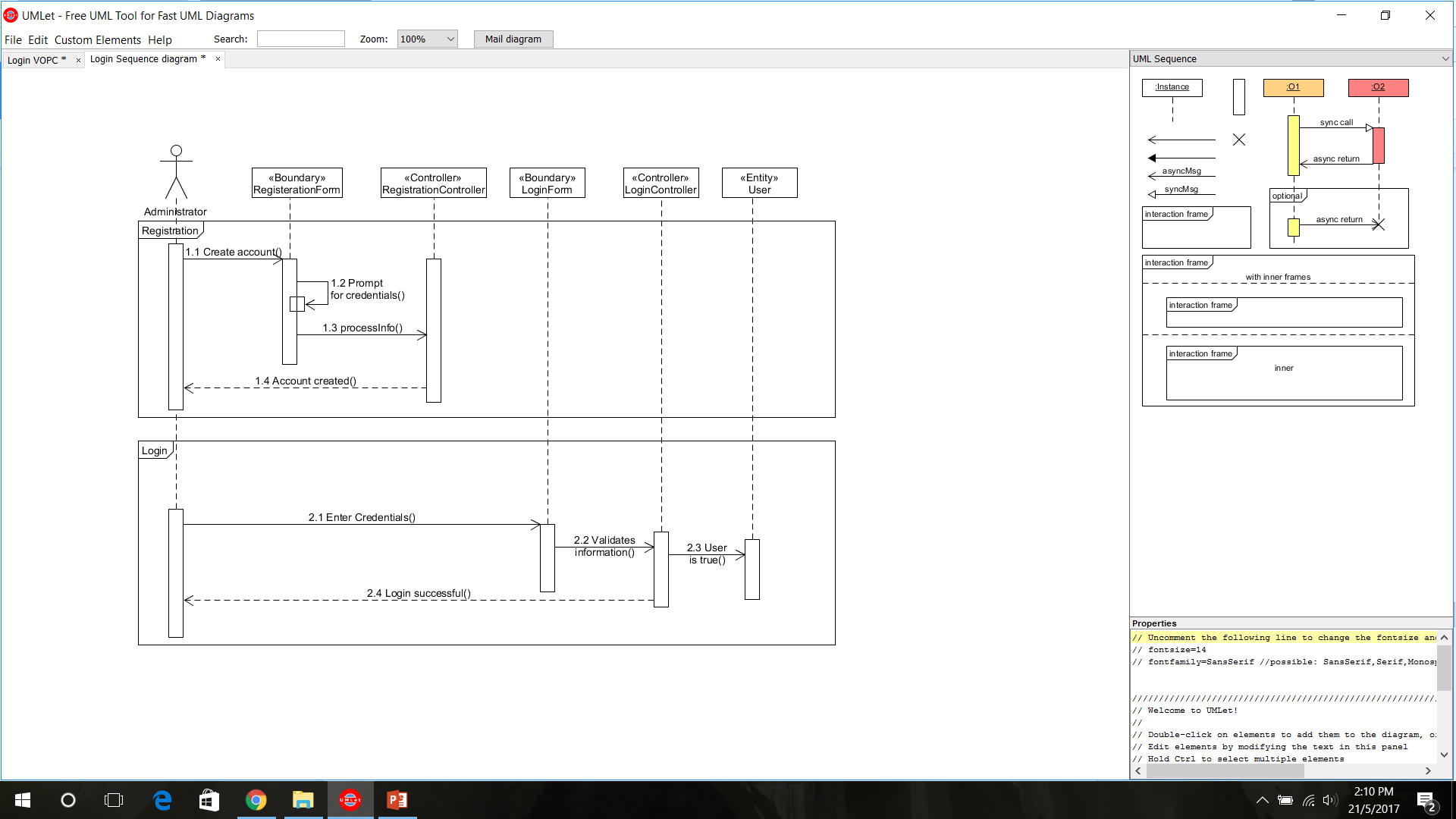
**Login**

VOPC



When a new staff is created, the above diagram shows main purpose of the function which is the login or registering for a staff account. The administrator will have the ability to create, retrieve, update and delete any information of the staff.

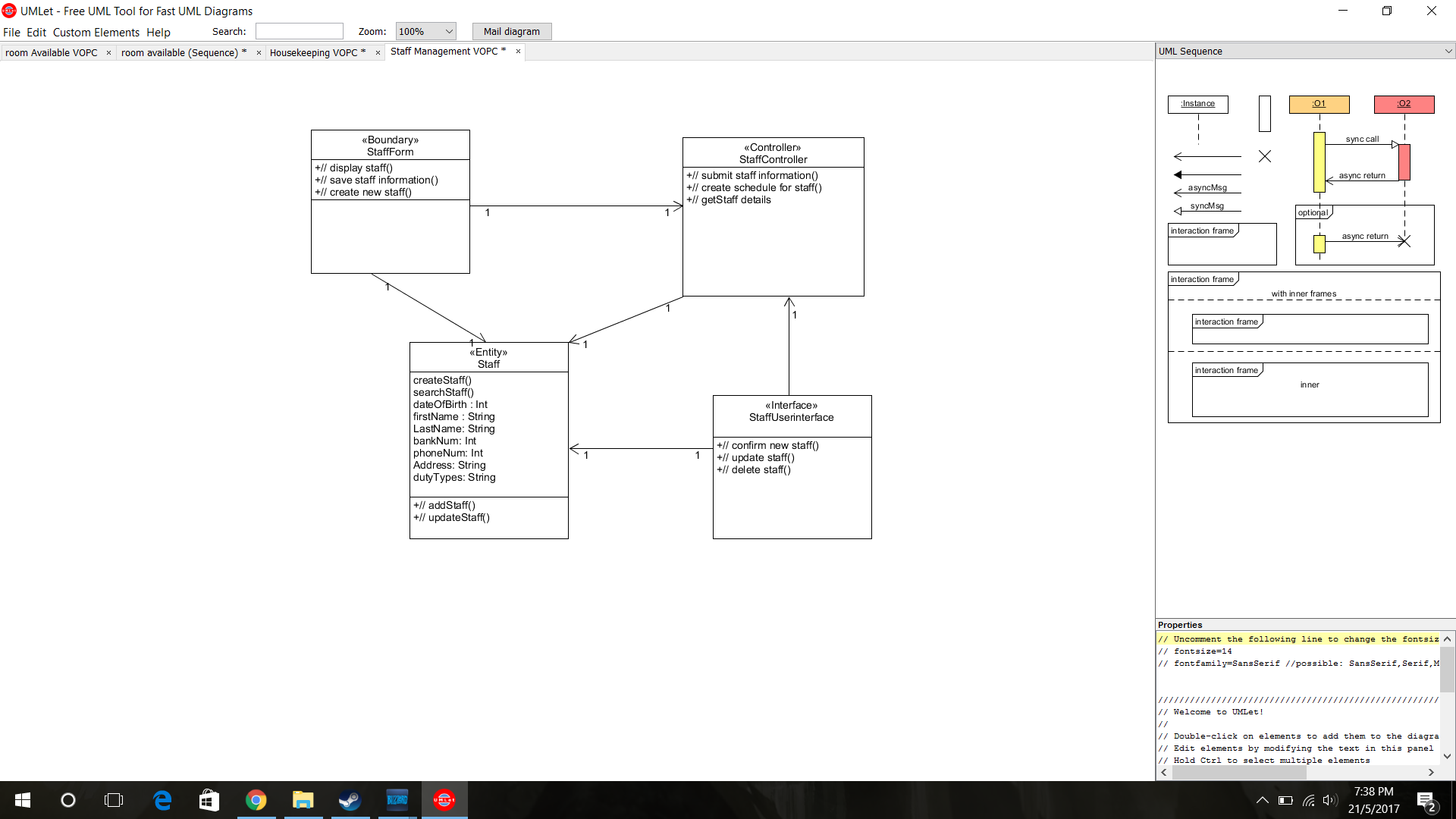
Sequences diagram



The sequence diagram above shows the process of Registering and Login of the staff account.

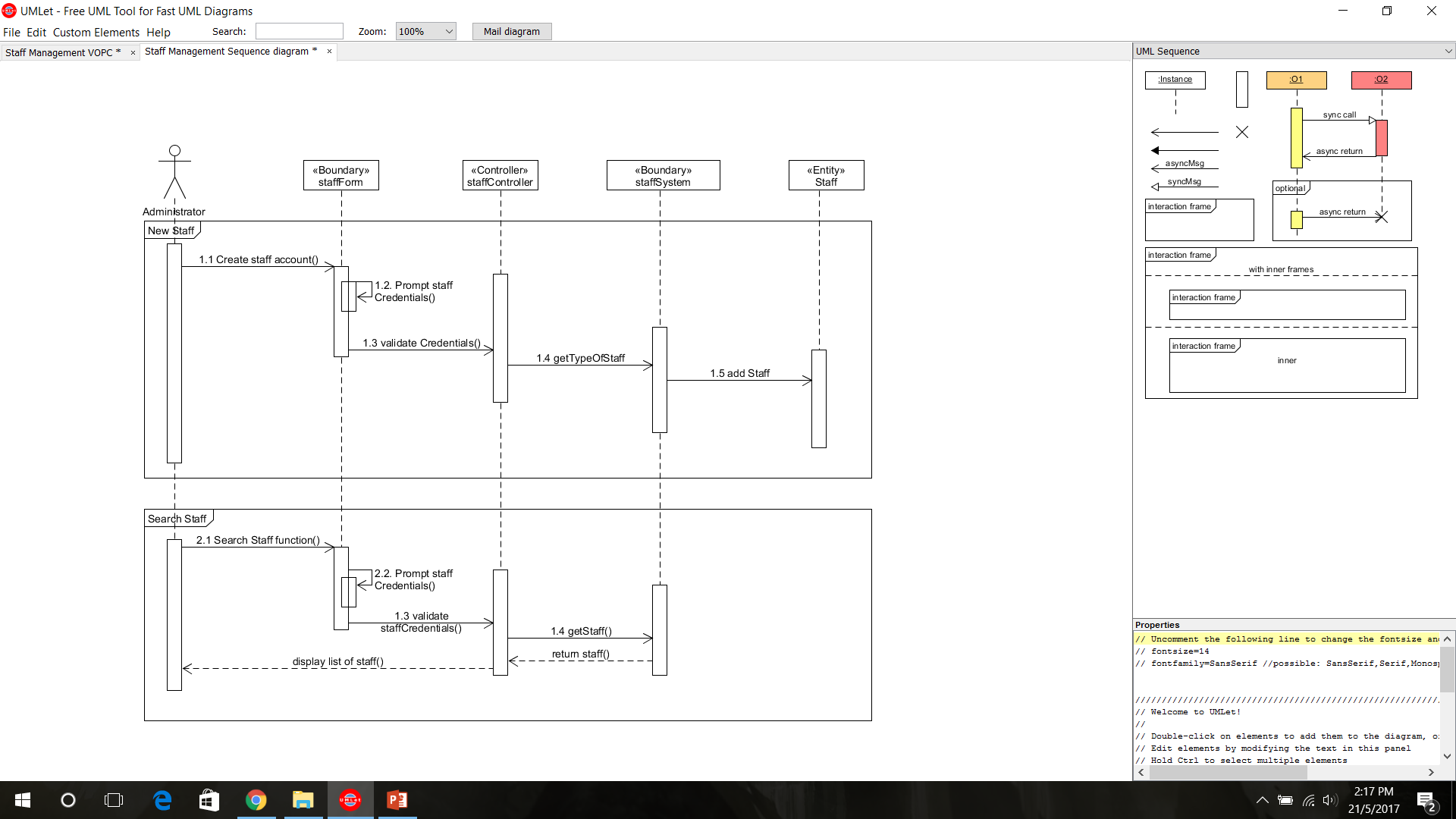
**Staff Management**

VOPC:



The function fo Staff management is used for assigning staff to their designated jobs. The administrator of staff management will be able to create, retrieve, update and delete information regarding staff details such as staff’s schedule, duty types or staff’s particulars.

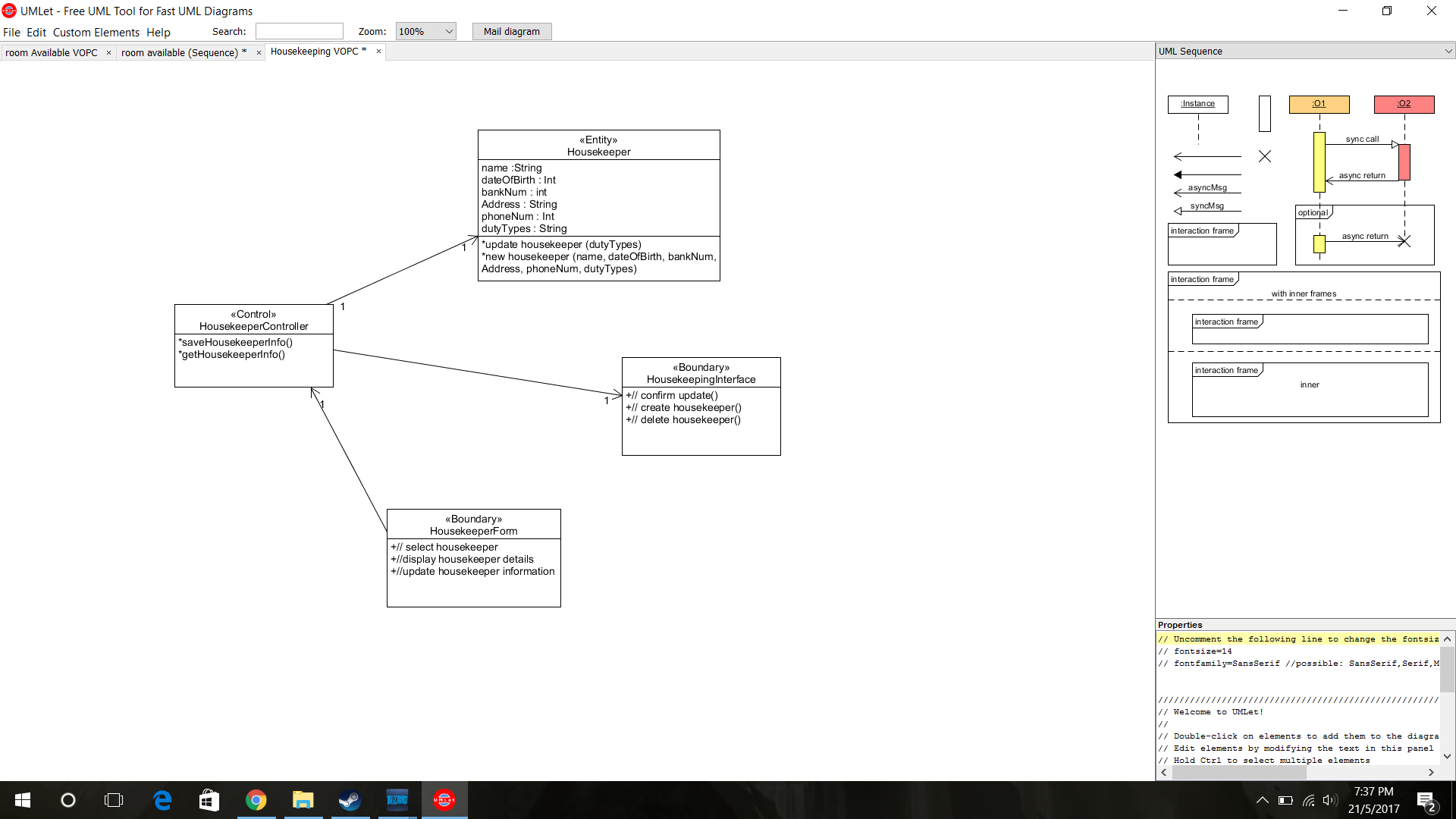
Sequence Diagram:



The sequence diagram shows the function of adding a staff into the company system and the ability to search for a staff.

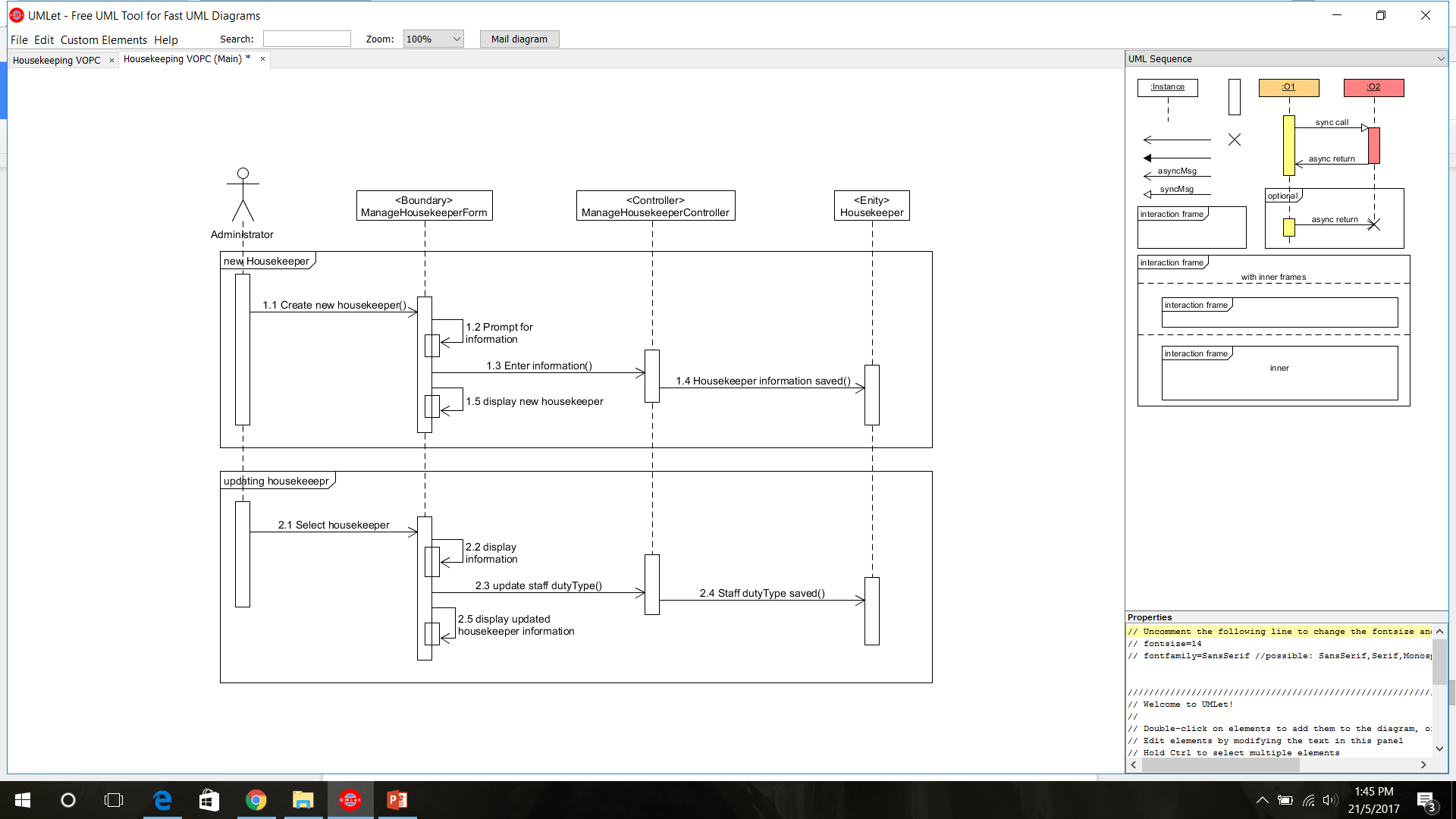
**Housekeeping management**

VOPC



Housekeeping management is solely used for staffs that belongs to the room service department. The administrator would be able to create, retrieve, update and delete any housekeeper within the company. The housekeeping management also allow the staff to view which room are being clean or unavailable for cleaning to accommodate with the guest to prevent obstruction.

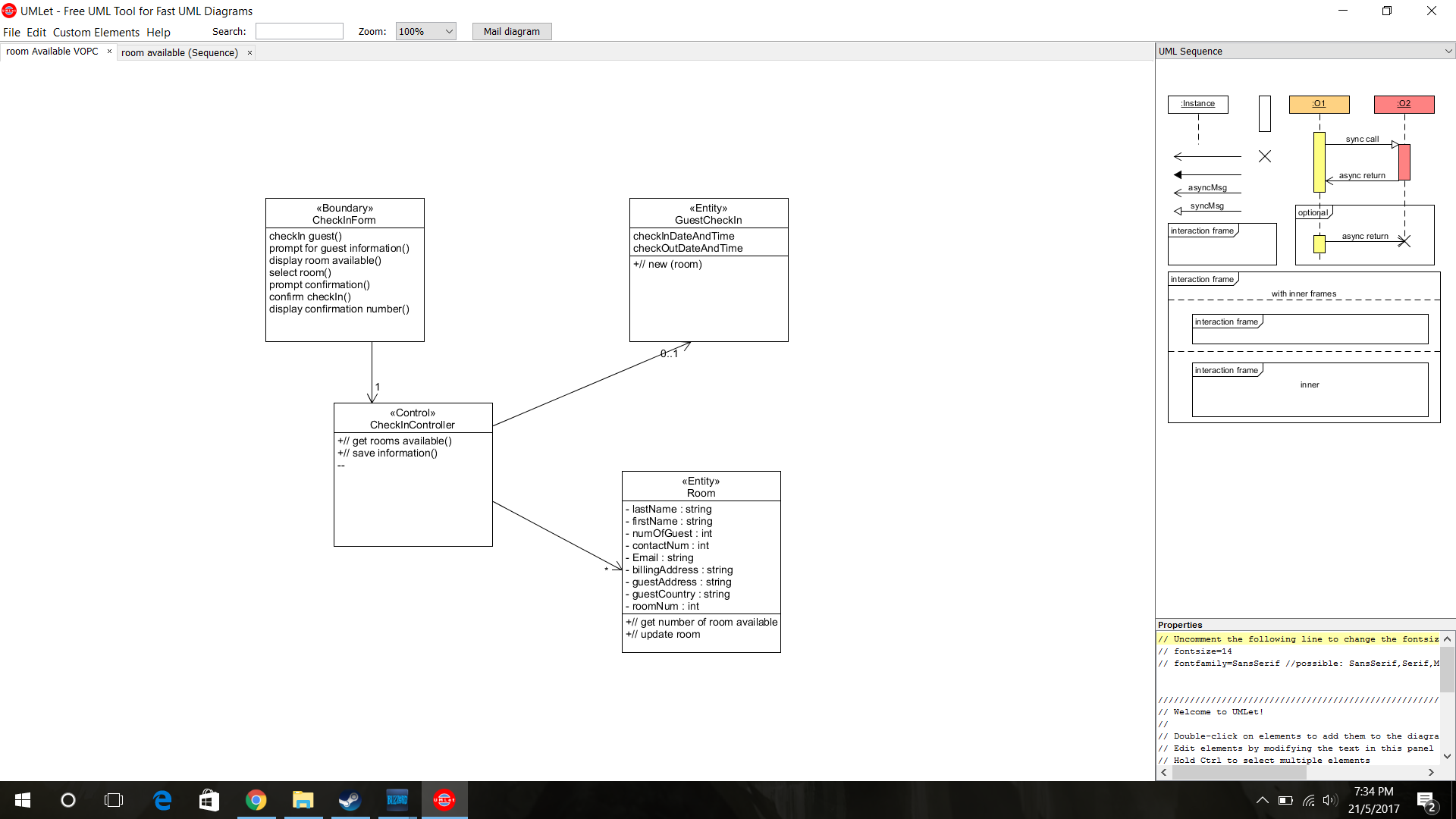
Sequence diagram:



The sequence diagram shows the administrator creating an account for a new housekeeper and updating a housekeeper to different duty types.

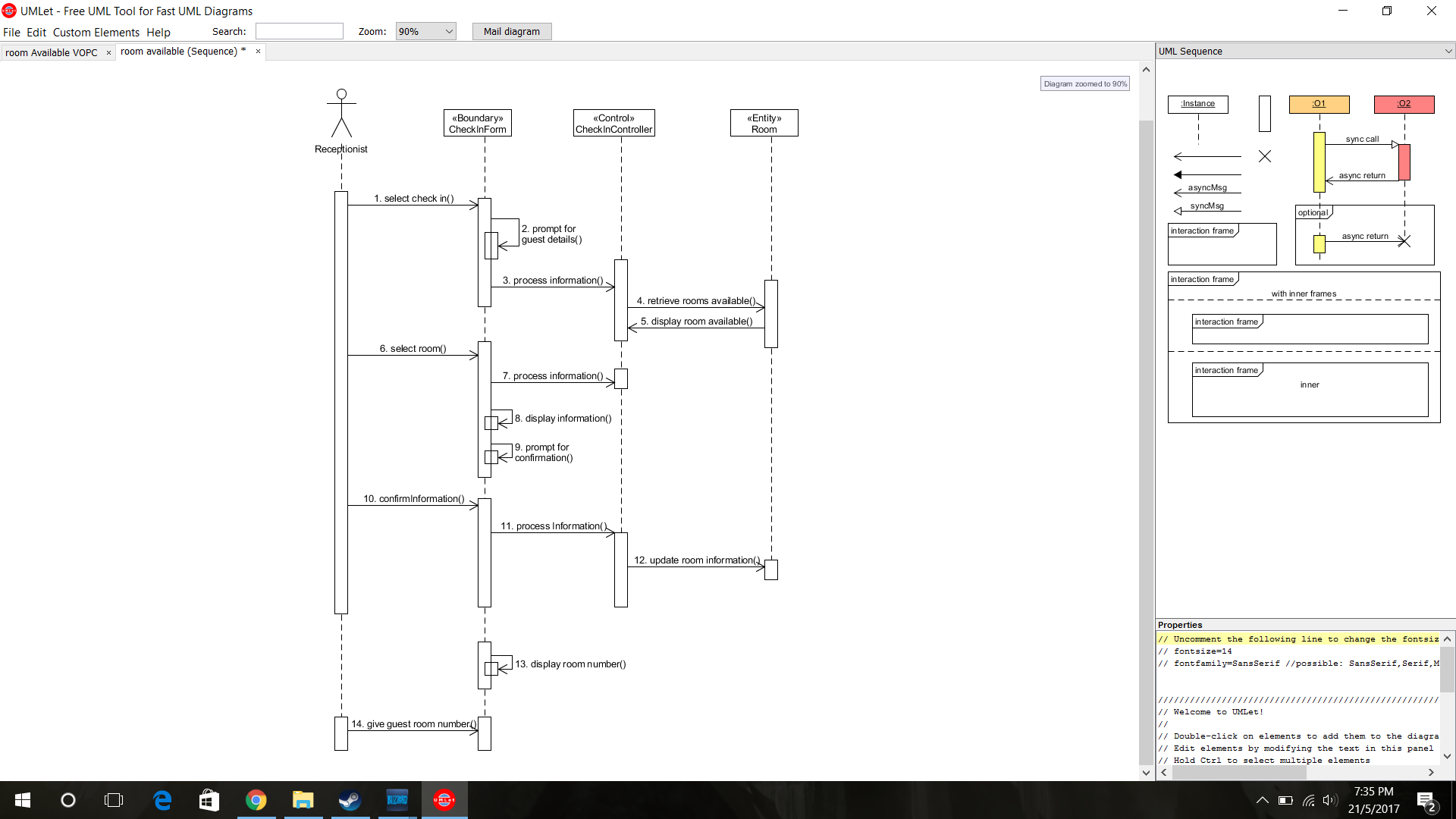
**Room Availability management**

VOPC



The room availability functions allows the receptionist to assign a room to a guest, where the guest has to give their particulars to the receptionist. The receptionist will be able to add, retrieve, update and delete customer particulars through the rooms they reserve.

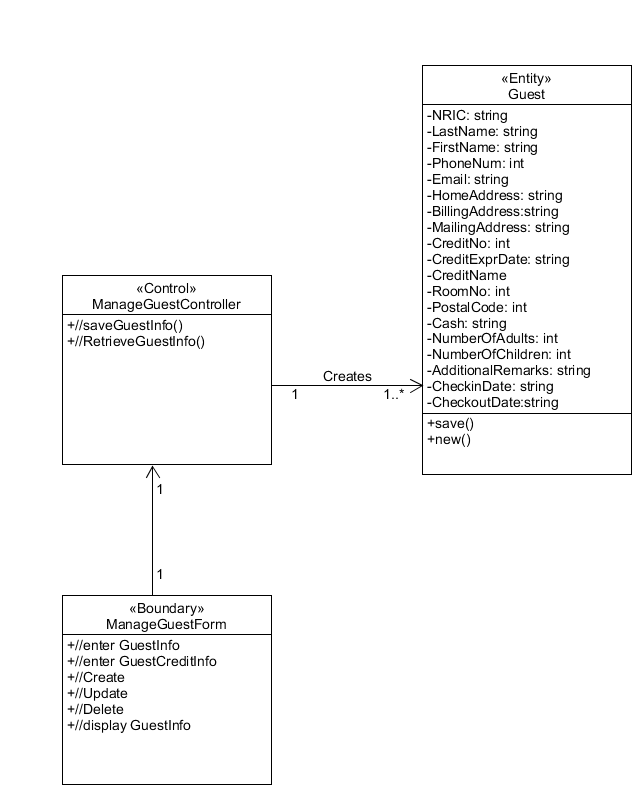
Sequence diagram



The sequence diagram above displays the process the receptionist booking a room for a guest.

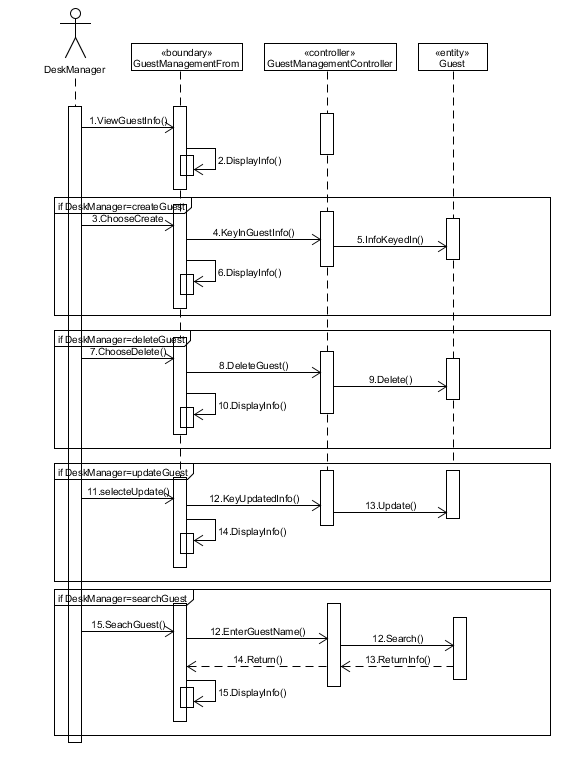
**Guest Management**

VOPC:



Once the hotel rooms have been booked, and the reception staff would enter the guest information into the booking system, the guest management page would display all their guest’s details. The page can also be edited through the CRUD functions of needed by the reception staff.

Sequence Diagram



The sequence diagram above shows the CRUD functions of the guest management system.

# **4.4.** **Database Design**

Database Schema:

Guest (**Guest\_ID**, Guest\_FirstName, Guest\_LastName, Guest\_Age, Guest\_DOB, Guest\_Address, Guest\_Country, Guest\_PostalCode, Guest\_Email\*)

Room (**Room\_ID**, Room\_Size, Room\_TotalOccupantsNo, AdultsNumber, ChildNumber, Room\_Price,Room\_Location, Guest\_ID)

Booking (**Booking\_ID**, Booking\_StartDate, Booking\_EndDate, Booking\_Extention, Room\_ID, Customer\_ID )

Customer (**Customer\_ID**,Customer\_NRIC, Customer\_FirstName, Customer\_LastName, Customer\_Age, Customer\_DOB, Customer\_Address, Customer\_Country, Customer\_PostalCode, Customer\_Email\*, Customer\_Requests, Booking\_ID, Invoice\_ID, PaymentMethod\_ID )

Invoice (**Invoice\_ID**, Status, Description, Bill\_ID)

Bill (**Bill\_ID**, Price\_Of\_Consumptions, No\_Of\_Nights\_Stayed, Room\_Price, Total\_Price)

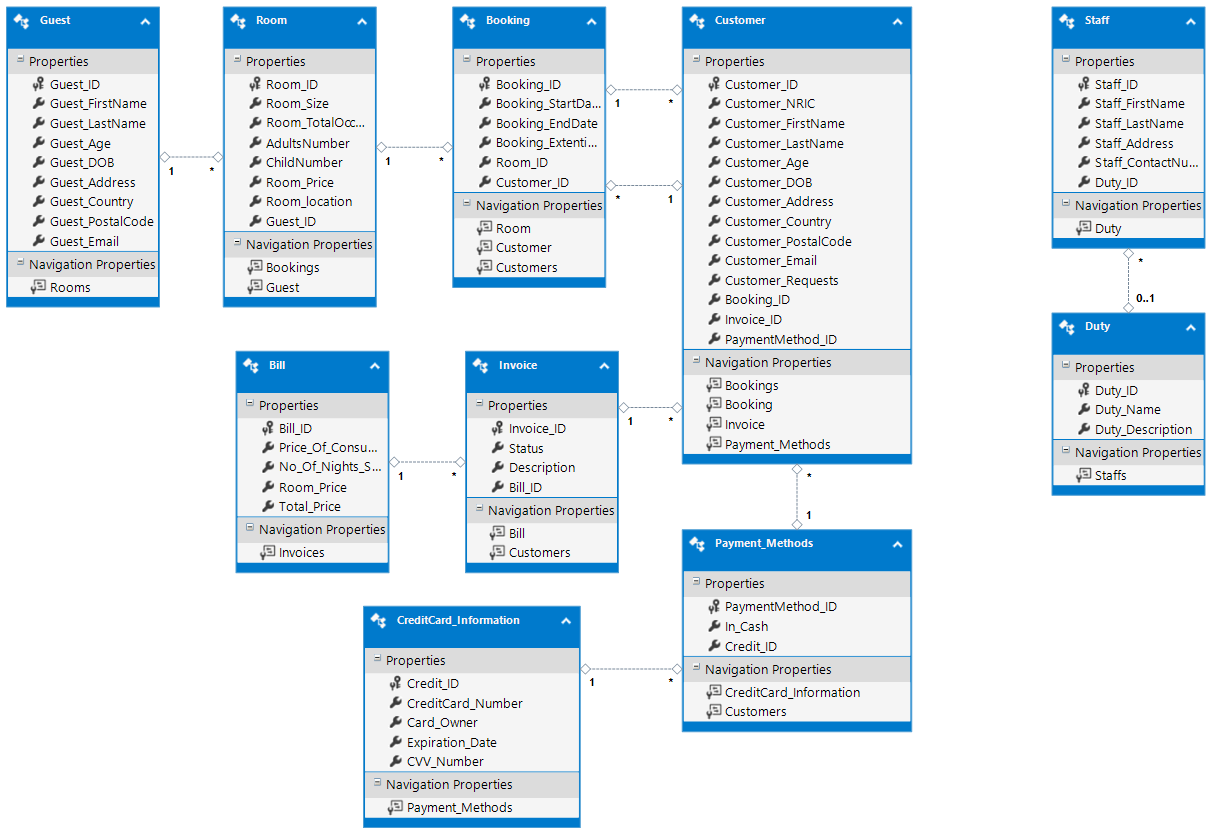
Payment\_Methods ( **PaymentMethod\_ID**, In\_Cash , Credit\_ID )

CreditCard\_Information ( **Credit\_ID**, CreditCard\_Number, Card\_Owner, Expiration\_Date, CVV\_Number )

Staff ( **Staff\_ID**, Staff\_FirstName, Staff\_LastName, Staff\_Address, Staff\_ContactNumber, Duty\_ID )

Duty ( **Duty\_ID**, Duty\_Name, Duty\_Description )

Database Entity Relational Diagram



Customer Requests Added into Customer Table

This field is added such that customers will be able to request for any changes or additions when checking into their rooms.

Meeting the Data Storage Requirements

All tables are normalized and ensured minimal repetitive values. This is such that there will not be waste of data space in the database due to repetitive values in the fields. As such, a duty table is created where all the duty of staff members is stored in such that it is not repeated in the staff table multiple times.

# 

# **5. References**

## **5.1. Gideon Ler:**

(n.d.). Retrieved May 22, 2017, from

<https://www.ibm.com/support/knowledgecenter/en/SSAW57_8.5.5/com.ibm.websphere.nd.doc/ae/covr_3-tier.html>

Kayal, S. (n.d.). Understand 3-Tier Architecture in C#. Retrieved May 22, 2017, from

<http://www.c-sharpcorner.com/UploadFile/dacca2/understand-3-tier-architecture-in-C-Sharp-net/>

Kayal, S. (2013, October 02). Understand 3- Tier Architecture in C#. Retrieved May 22,

2017, from <https://www.codeproject.com/Tips/662107/Understand-Tier-Architecture-in-Csharp>

Marston, T. (n.d.). Retrieved from[http://www.tonymarston.net/php-mysql/3-tier- architecture.html#3-tier-in-operation](http://www.tonymarston.net/php-mysql/3-tier-%20%20%20%20%20%20%20%20%20%20%20%20%20architecture.html#3-tier-in-operation)

## **5.2. Tan Hai Kang:**

Interface Requirements. (n.d.). Retrieved from

<https://www.dlsweb.rmit.edu.au/toolbox/ecommerce/tbn_respak/tbn_e2/html/tbn_e2_interface.htm>

(n.d.). Retrieved from

[https://msdn.microsoft.com/en-](https://msdn.microsoft.com/en- us/library/scf355x6(v=vs.90).aspx)

[us/library/scf355x6(v=vs.90).aspx](https://msdn.microsoft.com/en- us/library/scf355x6(v=vs.90).aspx)

(n.d.). Retrieved from

[https://msdn.microsoft.com/en-](https://msdn.microsoft.com/en- us/library/5t6z562c(v=vs.71).aspx)

[us/library/5t6z562c(v=vs.71).aspx](https://msdn.microsoft.com/en- us/library/5t6z562c(v=vs.71).aspx)

## **5.4. Eugene Sim**

Acceptable waiting time for users in time sensitive actions. (n.d.). Retrieved

from <https://ux.stackexchange.com/questions/58163/acceptable-waiting-time-for-users-in-time-sensitive-actions>

# **6. Peer Evaluation**

## **6.1. Tan Hai Kang**

|  |  |
| --- | --- |
| **Project Title:**  Delonix Regia Hotel Management System | |
| **Student No:**  **1506489B** | **Student Name:**  **Tan Hai Kang** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rate the overall team performance against each criterion. Circle one number from  1 (inadequate) to 5 (superior) | | | | | |
| Team spirit | **1** | **2** | **3** | **4** | **5** |
| Overall effectiveness | **1** | **2** | **3** | **4** | **5** |
| Rewarding experience | **1** | **2** | **3** | **4** | **5** |
| Team productivity | **1** | **2** | **3** | **4** | **5** |
| Process quality | **1** | **2** | **3** | **4** | **5** |
| Product quality | **1** | **2** | **3** | **4** | **5** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rate the contribution of each team member (including yourself). Circle one number from  1 (inadequate) to 5 (superior) | | | | | |
| Myself | **1** | **2** | **3** | **4** | **5** |
| <Gideon Ler> | **1** | **2** | **3** | **4** | **5** |
| <Brandon Chew> | **1** | **2** | **3** | **4** | **5** |
| <Eugene Sim> | **1** | **2** | **3** | **4** | **5** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rate the quality of work (including timeliness) of each team member (including yourself). Circle one number from 1 (inadequate) to 5 (superior) | | | | | |
| Myself | **1** | **2** | **3** | **4** | **5** |
| <Gideon Ler> | **1** | **2** | **3** | **4** | **5** |
| <Brandon Chew> | **1** | **2** | **3** | **4** | **5** |
| <Eugene Sim> | **1** | **2** | **3** | **4** | **5** |
| Rate the help and support you have received from each team member. For yourself, rate the support and help you have given to other team members. Circle one number from  1 (inadequate) to 5 (superior) | | | | | |
| Myself | **1** | **2** | **3** | **4** | **5** |
| <Gideon Ler> | **1** | **2** | **3** | **4** | **5** |
| <Brandon Chew> | **1** | **2** | **3** | **4** | **5** |
| <Eugene Sim> | **1** | **2** | **3** | **4** | **5** |

|  |
| --- |
| **Comments:** |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## **6.2. Gideon Ler**

|  |  |
| --- | --- |
| **Project Title:**  Delonix Regia Hotel Management System | |
| **Student No:**  **1502698B** | **Student Name:**  **Gideon Ler** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rate the overall team performance against each criterion. Circle one number from  1 (inadequate) to 5 (superior) | | | | | |
| Team spirit | **1** | **2** | **3** | **4** | **5** |
| Overall effectiveness | **1** | **2** | **3** | **4** | **5** |
| Rewarding experience | **1** | **2** | **3** | **4** | **5** |
| Team productivity | **1** | **2** | **3** | **4** | **5** |
| Process quality | **1** | **2** | **3** | **4** | **5** |
| Product quality | **1** | **2** | **3** | **4** | **5** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rate the contribution of each team member (including yourself). Circle one number from  1 (inadequate) to 5 (superior) | | | | | |
| Myself | **1** | **2** | **3** | **4** | **5** |
| <Tan Hai Kang> | **1** | **2** | **3** | **4** | **5** |
| <Brandon Chew> | **1** | **2** | **3** | **4** | **5** |
| <Eugene Sim> | **1** | **2** | **3** | **4** | **5** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rate the quality of work (including timeliness) of each team member (including yourself). Circle one number from 1 (inadequate) to 5 (superior) | | | | | |
| Myself | **1** | **2** | **3** | **4** | **5** |
| <Tan Hai Kang> | **1** | **2** | **3** | **4** | **5** |
| <Brandon Chew> | **1** | **2** | **3** | **4** | **5** |
| <Eugene Sim> | **1** | **2** | **3** | **4** | **5** |
| Rate the help and support you have received from each team member. For yourself, rate the support and help you have given to other team members. Circle one number from  1 (inadequate) to 5 (superior) | | | | | |
| Myself | **1** | **2** | **3** | **4** | **5** |
| <Tan Hai Kang> | **1** | **2** | **3** | **4** | **5** |
| <Brandon Chew> | **1** | **2** | **3** | **4** | **5** |
| <Eugene Sim> | **1** | **2** | **3** | **4** | **5** |

|  |
| --- |
| **Comments:** |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## **6.3. Brandon Chew**

|  |  |
| --- | --- |
| **Project Title:**  Delonix Regia Hotel Management System | |
| **Student No:**  **1502041E** | **Student Name:**  **Brandon Chew** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rate the overall team performance against each criterion. Circle one number from  1 (inadequate) to 5 (superior) | | | | | |
| Team spirit | **1** | **2** | **3** | **4** | **5** |
| Overall effectiveness | **1** | **2** | **3** | **4** | **5** |
| Rewarding experience | **1** | **2** | **3** | **4** | **5** |
| Team productivity | **1** | **2** | **3** | **4** | **5** |
| Process quality | **1** | **2** | **3** | **4** | **5** |
| Product quality | **1** | **2** | **3** | **4** | **5** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rate the contribution of each team member (including yourself). Circle one number from  1 (inadequate) to 5 (superior) | | | | | |
| Myself | **1** | **2** | **3** | **4** | **5** |
| <Tan Hai Kang> | **1** | **2** | **3** | **4** | **5** |
| <Gideon Ler> | **1** | **2** | **3** | **4** | **5** |
| <Eugene Sim> | **1** | **2** | **3** | **4** | **5** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rate the quality of work (including timeliness) of each team member (including yourself). Circle one number from 1 (inadequate) to 5 (superior) | | | | | |
| Myself | **1** | **2** | **3** | **4** | **5** |
| <Tan Hai Kang> | **1** | **2** | **3** | **4** | **5** |
| <Gideon Ler> | **1** | **2** | **3** | **4** | **5** |
| <Eugene Sim> | **1** | **2** | **3** | **4** | **5** |
| Rate the help and support you have received from each team member. For yourself, rate the support and help you have given to other team members. Circle one number from  1 (inadequate) to 5 (superior) | | | | | |
| Myself | **1** | **2** | **3** | **4** | **5** |
| <Tan Hai Kang> | **1** | **2** | **3** | **4** | **5** |
| <Gideon Ler> | **1** | **2** | **3** | **4** | **5** |
| <Eugene Sim> | **1** | **2** | **3** | **4** | **5** |

|  |
| --- |
| **Comments:** |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## **6.4. Eugene Sim**

|  |  |
| --- | --- |
| **Project Title:**  Delonix Regia Hotel Management System | |
| **Student No:**  **1500632C** | **Student Name:**  **Eugene Sim** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rate the overall team performance against each criterion. Circle one number from  1 (inadequate) to 5 (superior) | | | | | |
| Team spirit | **1** | **2** | **3** | **4** | **5** |
| Overall effectiveness | **1** | **2** | **3** | **4** | **5** |
| Rewarding experience | **1** | **2** | **3** | **4** | **5** |
| Team productivity | **1** | **2** | **3** | **4** | **5** |
| Process quality | **1** | **2** | **3** | **4** | **5** |
| Product quality | **1** | **2** | **3** | **4** | **5** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rate the contribution of each team member (including yourself). Circle one number from  1 (inadequate) to 5 (superior) | | | | | |
| Myself | **1** | **2** | **3** | **4** | **5** |
| <Tan Hai Kang> | **1** | **2** | **3** | **4** | **5** |
| <Gideon Ler> | **1** | **2** | **3** | **4** | **5** |
| <Brandon Chew> | **1** | **2** | **3** | **4** | **5** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Rate the quality of work (including timeliness) of each team member (including yourself). Circle one number from 1 (inadequate) to 5 (superior) | | | | | |
| Myself | **1** | **2** | **3** | **4** | **5** |
| <Tan Hai Kang> | **1** | **2** | **3** | **4** | **5** |
| <Gideon Ler> | **1** | **2** | **3** | **4** | **5** |
| <Brandon Chew> | **1** | **2** | **3** | **4** | **5** |
| Rate the help and support you have received from each team member. For yourself, rate the support and help you have given to other team members. Circle one number from  1 (inadequate) to 5 (superior) | | | | | |
| Myself | **1** | **2** | **3** | **4** | **5** |
| <Tan Hai Kang> | **1** | **2** | **3** | **4** | **5** |
| <Gideon Ler> | **1** | **2** | **3** | **4** | **5** |
| <Brandon Chew> | **1** | **2** | **3** | **4** | **5** |

|  |
| --- |
| **Comments:** |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**