Temasek Polytechnic

School of Informatics and IT

**Diploma in Information Technology (IT)**

Meeting Minutes

**Project Particulars**

|  |  |
| --- | --- |
| **Tutor** | Qi Yutao |
| **Class** | P01 |
| **Project Title** | Delonix Regia Hotel Management System |

**Project Team’s Particulars**

|  |  |
| --- | --- |
| **Matric Number** | **Student Name** |
| **1602691F** | **Myron Low** |
| **1605809A** | **Zafrulla** |
| **1605894C** | **Zulhilmi** |

|  |  |  |  |
| --- | --- | --- | --- |
| Date: | 11/11/2017 | |  |
|  |  | |  |
| Venue: | Meeting Room | |  |
|  |  | |  |
| Present: | Mr Wang, Interviewer, Myron, Zafrulla, Zulhilmi | |  |
|  |  |  | |
| Absent with apologies: | NIL | |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **S/No** | **Item** | | **Action By** |
|  | Meeting started at 1:00 pm    3 modules: room availability and booking module, housekeeping and staff management and a reporting module  3 general users:  **Reception** staff - access parts of reporting module, full access to room availability and booking  **Management** staff - can have full access to all 3 modules, **Administrative** staff - full access to all 3 modules and a user account and login creation module.  Room availability: system should accept the data inputs like first name, last name, number of adult guest and children, phone number email and address, billingaddress include street block house number, postal code and country. Payment details like credit card, credit card number, cardholder name, expiration date.  Check in details like check in date and time and desired check out date and time. Additional remarks like whether king or queen sized is needed, a smoking room or non-smoking room**, Included in data input additional remarks** Indication if guest is asking for a late checkout.  Process of booking: stipulated check out time is before 12pm, pass room key to reception, ask guest if they consume any items from the room, generate payment invoice, calculate number of days, whether anything was taken from **hotel(minibar in the room)**. Include details of guests, room rates and additional cost.  Additional features for booking module: let admin and reception staff edit guest records. Example if guest wants change room, or another person joins in, small details like number of children or adult should be able to change  Housekeeping management: 2 features : keep record of staff details, name, date of birth, bank account number, all home address details, phone number, duty types the staff can do. 1)general maintenance  2)room maintenance  3)estate maintenance  4)security  Reporting module: 5 reports:  1) **room status** - list all rooms and their room status, vacant, occupied or vacant and scheduled for cleaning  2) **list all guest in a room** - example 2 adult 1 child  3) **list all the guest** - in **all** the rooms in **any point of time**  4) **room occupancy report** - statistics indicating room occupancy daily weekly monthly or yearly. (Only available to admin and management)  5) **housekeeping report** - list the 4 duties of the staff has been allocated, generate housekeeping schedule based on a daily, weekly or monthly (available to the admin and management)  Reporting module add features:  1) preview the reports before sending to the printer  Budget: $70k  Software used: installed in a single computer at the reception area. Pantheon (**or pentium**) 4, pc windows xp SP2, 1gb ram and 160gb hard disk, internet broadband connection  Integration with other systems: not needed  Backups: system shouldn't go down... Doing backup in the morning preferably 2-3am.. First Sunday of every month to have a backup record to be kept for 5 years. | | **Myron**   * Room availability and booking module * Reception Staff * Process flow when checking in and out * Additional features for Room Availability * Integration with other systems   **Zafrulla**   * Housekeeping and staff management module * Management Staff * Type of duties * Budget for the system * Backups for the system   **Zulhilmi**   * Reporting Module * Administrative Staff * Type of reports * Additional features for Reporting Module * Software operating environment |
|  |  | |  |
|  |  | |  |
|  |  |  |  |
|  |  |  |  |

Meeting ended at 1.20pm

Recorded by: Myron Low

Vetted by: Zaf and Zul

Temasek Polytechnic

School of Informatics and IT

**Diploma in Information Technology (IT)**

Software Requirement Specifications (SRS)

**Project Particulars**

|  |  |
| --- | --- |
| **Tutor** | Qi Yutao |
| **Class** | P01 |
| **Project Title** | Delonix Regia Hotel Management System |

**Project Team’s Particulars**

|  |  |
| --- | --- |
| **Matric Number** | **Student Name** |
| **1602691F** | **Myron Low** |
| **1605809A** | **Zafrulla** |
| **1605894C** | **Zulhilmi** |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 15/11/2017 | 1.0 | Started and finalized the System Functions and User Characteristic | Myron  Zafrulla  Zulhilmi |
| 17/11/2017 | 1.5 | Started a draft on General Constraints, Functional Requirements and Data Requirements. | Zafrulla  Zulhilmi |
| 18/11/2017 | 2.0 | Finalized all the General Constraints, Functional Requirements and Data Requirements. | Zafrulla  Zulhilmi  Myron |
| 19/11/2017 | 2.5 | Started and finalized Interface with Other Systems, Operating Environment and Development Constraints. | Zafrulla  Zulhilmi |
| 20/11/2017 | 3.0 | Started and finalized all the Performance, Availability and Security and Access Control Requirements. | Zafrulla  Zulhilmi  Myron |
| 21/11/2017 | 3.5 | Started and finalized all the User Interface Requirements, Assumption and Special Requirements. | Zafrulla  Zulhilmi |
| 21/11/2017 | 4.0 | Finalized the formatting of the report. | Myron |

**Table of Contents**

1. DISTRIBUTION OF WORKLOAD. 4

2. OVERVIEW OF REQUIREMENTS. 4

2.1 System Functions. 4

2.2 User Characteristics. 4

2.3 General Constraints. 4

2.4 Functional Requirements. 4

2.5 Data Requirements. 4

2.6 User Interface Requirements. 5

2.7 Interface with Other Systems. 5

2.8 Assumptions. 5

3 OPERATIONAL AND QUALITY REQUIREMENTS. 5

3.1 Operating Environment 5

3.2 Development Constraints. 5

3.3 Performance. 5

3.4 Availability. 5

3.5 Security and Access Control Requirements. 6

4 SPECIAL REQUIREMENTS. 6

5 REFERENCES. 6

# **1.** **DISTRIBUTION OF WORKLOAD**

|  |  |
| --- | --- |
| **Requirement Gathering** | **Members** |
| * System Functions * Functional Requirements * Security and Access Control Requirements | Myron |
| * User Characteristic * General Constraints * User Interface Requirements * Assumptions * Special Requirements | Zafrulla |
| * Data Requirements * Interface with Other Systems * Operating Environment * Development Constraints * Performance * Availability | Zulhilmi |

# 

# **2.** **OVERVIEW OF REQUIREMENTS**

## **2.1** **System Functions**

**Room availability and booking module**

**Room availability and booking:** System should accept the data inputs like first name, last name, number of adult guest and children, phone number email and address, mailing address include street block house number, postal code and country. Additional remarks such as whether king or queen sized is needed and a smoking room or non-smoking room. Check in details like check in date and time and desired check out date and time.

**Payment:** System should accept details like credit card number, cardholder name and expiration date. Payment by cash is also an available option.

**Additional features for booking:** Let admin and reception staff edit guest records. Example if guest want change room, or another person joins in, small details like number of children or adult should be able to change.

**Housekeeping and staff management module**

**Keep record of staff details:** First name, last name, date of birth, bank account number, all home address details and phone number.

**Duty types of staff:** General maintenance, room maintenance, estate maintenance and security.

**Reporting module**

1. Room status, list all rooms and their room status; vacant, occupied or vacant and scheduled for cleaning
2. List all guest in a room, example 2 adult 1 child
3. List all the guest in all the rooms in any point of time
4. Room occupancy report, statistics indicating room occupancy daily weekly monthly or yearly. This statistic can be shown in pie chart or graph format. (Only available to admin and management)
5. Housekeeping report, list the 4 duties of the staff has been allocated, generate housekeeping schedule based on a daily, weekly or monthly (available to the admin and management)

## **2.2** **User Characteristics**

**Reception staff**

The reception staff are the staff located mostly at the front desk where the guests can walk up to them for enquiries.

* They also have full access to the room availability and booking module.
* The reception staff can view certain parts of the reporting module. These parts include ⅗ of the whole reporting module.

1. They can view the room status. This allows them to list all rooms and their status to see if it is vacant, occupied or vacant and scheduled for cleaning.
2. They can view all the guests in a room.
3. They can also view all the guest in all the rooms at any given time.

**Management staff**

The management staff includes the hotel manager and other executive positions.

* Full access to 3 modules

1. Room availability and booking module: Viewing guest details and booking dates. They are also able to edit guest records
2. Housekeeping and staff management module: Viewing the staff details and duty type
3. Reporting module: All 5 parts of the module can be accessed (Refer to 2.1)

**Administrative staff**

The administrative staff include Mr and Mrs Wang who are the higher ups.

* Full access to all modules

1. Room availability and booking module: Viewing guest details and booking dates. They are also able to edit guest records
2. Housekeeping management module: Viewing the staff details and duty type
3. Reporting module : All 5 parts of the module can be accessed
4. User account and login creation module

## **2.3** **General Constraints**

**Hardware/software constraint**

System can only be used/installed on a single computer at the front desk.

* Windows XP SP2
* Intel Pentium 4
* 1GB Ram
* 160GB HDD

With this hardware/software restriction, the developed software must be able to function with such low specifications.

**Backup constraint**

The data (reports) should be backed up to a server instead of the internal HDD so that the backup data can be retrieved from a separate machine if the main computer were to fail. The server that we chose is a Network Attached Storage (NAS) device called, “Western Digital My Cloud DL2100” which has 4TB of storage capacity.

**Budget constraint**

Budget should comprise of the NAS device we chose which cost around $900. Also, a typical software costs $40,000 to $250,000 to develop. With the final budget of $70,000, we should compromise on hiring the software developers and other extra costs that might come up in the future while developing.

## **2.4** **Functional Requirements**

For room availability and booking, the system must accept and store data such as the room number, room type, room status, remarks, guest ID, check-in/check-out date and the number of adults and children per booking. When checking for a room availability, the room status will indicate whether the room is vacant or occupied. A guest ID will be automatically generated when the system saves the guest’s booking details.

For payment, the system must accept and store data such as the payment ID, guest ID, credit card number, full name and the expiry date of the card. The payment ID will be automatically generated when the payment details are saved in the system.

For housekeeping, the system must keep record of staff details, name, date of birth, bank account number, all home address details, phone number, duty types the staff can do.

1. general maintenance
2. room maintenance
3. estate maintenance
4. security

For reporting, the system should list all rooms and their room status, vacant, occupied or vacant and scheduled for cleaning. Next it should list all guest in a room - example 2 adult 1 child. Then it should list all the guest - in all the rooms in any point of time.

The system should generate a room occupancy report - statistics indicating room occupancy daily weekly monthly or yearly. It should also generate a housekeeping report which includes the 4 duties (General maintenance, Room maintenance, Estate maintenance and Security).

The system should also include additional input, “Remarks” for users if they want to have a smoking room or not, a king size bed or not. The system should also allow the admin/management to preview the reports before printing them.

## **2.5** **Data Requirements**

**Staff Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Field Size | Data Format | Description | Example |
| Staff\_ID | integer | 4 | NNNN | Unique number ID for every staff | 1001 |
| Staff\_FirstName | text | 20 | -NIL- | First name of Staff | Jones |
| Staff\_LastName | text | 20 | -NIL- | Last name of Staff | Arnold |
| Staff\_DOB | date | 10 | DD/MM/YYYY | Date of Birth for Staff | 12/04/1964 |
| Staff\_NRIC | text | 9 | SNNNNNNNH | NRIC of Staff | S1234567H |
| Staff\_Address | text | 100 | -NIL- | Address of Staff | 14 Tanjong Penjuru Crescent, 608976, Singapore |
| Staff\_BankAcctNo | integer | 10 | NNNNNNNNNN | Bank Account Number of Staff which is used to pay their salary | 1234567891 |
| Staff\_PhoneNo | integer | 8 | NNNNNNNN | Phone Number of Staff | 12345678 |

**Staff Account Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Field Size | Data Format | Description | Example |
| Account\_ID | integer | 4 | NNNN | Unique number ID for the Staff account | 1001 |
| Staff\_ID | integer | 4 | NNNN | Unique number ID for every staff | 2001 |
| Staff\_Username | text | 20 | -NIL- | Username of Staff account | Jones |
| Staff\_Password | text | 10 | -NIL- | Password of Staff account | 123jones |
| Staff\_Type | text | 10 | -NIL- | Type of Staff | Reception |

**Guest Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Field Size | Data Format | Description | Example |
| Guest\_ID | integer | 4 | NNNN | Unique number ID for Guest | 1001 |
| Guest\_FirstName | text | 20 | -NIL- | First Name of Guest | Jones |
| Guest\_LastName | text | 20 | -NIL- | Last Name of Guest | Arnold |
| Guest\_Address | text | 100 | -NIL- | Address of Guest | 14 Tanjong Penjuru Crescent, 608976, Singapore |
| Guest\_Email | text | 50 | -NIL- | Email address of Guest | Jones@gmail.com |
| Guest\_PhoneNumber | integer | 8 | NNNNNNNN | Phone Number of Guest | 12345678 |

**Room Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Field Size | Data Format | Description | Example |
| Room\_No | integer | 4 | NNNN | Unique number ID for the Rooms in the hotel | 2001 |
| Room\_Type | text | 20 | -NIL- | The type of Rooms in the hotel | Deluxe |
| Room\_Status | text | 50 | -NIL- | Status of the room | Vacant and scheduled for cleaning |

**Booking Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Field Size | Data Format | Description | Example |
| Booking\_ID | integer | 4 | NNNN | Unique number ID for Booking details | 2001 |
| Room\_No | integer | 4 | NNNN | Unique number ID for the Rooms in the hotel | 2001 |
| Room\_Remark | text | 100 | -NIL- | Remarks that the Guest inputs | Change to King size bed |
| Guest\_ID | integer | 4 | NNNN | Unique number ID of Guest which is used to link with Booking details | 2001 |
| CheckIn\_Date | datetime | 15 | DD/MM/YYYY HH:MM:SS | Check in date and time of the Guest | 20/11/2017 10:30:00 |
| CheckOut\_Date | datetime | 15 | DD/MM/YYYY HH:MM:SS | Check out date and time of the Guest | 25/11/2017 10:30:00 |
| Number\_Of\_Adult | integer | 1 | N | Number of adults per booking | 2 |
| Number\_Of\_Children | integer | 1 | N | Number of children per booking | 3 |

**Payment Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Field Size | Data Format | Description | Example |
| Payment\_ID | integer | 4 | NNNN | Unique number ID for Payment details | 1001 |
| Guest\_ID | integer | 4 | NNNN | Unique number ID of Guest which is used to link with their payment details | 1001 |
| Credit\_Card\_No | integer | 16 | NNNNNNNNNNNNNNNN | Credit card number of Guest | 1234567812345678 |
| Card\_FullName | text | 50 | -NIL- | Full name of the Guest | Jones Arnold |
| Card\_ExpiryDate | date | 4 | MM/YY | Expiration date of credit card | 04/21 |

**Housekeeping Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Data Type | Field Size | Data Format | Description | Example |
| Duty\_ID | integer | 4 | NNNN | Unique number ID for housekeeping duty | 1001 |
| Duty\_Type | text | 40 | -NIL- | Type of Duty for the housekeeping | Room maintenance |
| Duty\_Remark | text | 150 | -NIL- | Remark about the Duty | Room 1001, 1003, 1006 |
| Staff\_ID | integer | 4 | NNNN | Unique number ID for Staff which is used to link with housekeeping details | 2001 |
| Schedule\_StartDate | datetime | 15 | DD/MM/YYYY HH:MM:SS | Schedule start date for the housekeeping to perform their Duty | 20/11/2017 10:30:00 |
| Schedule\_EndDate | datetime | 15 | DD/MM/YYYY HH:MM:SS | Schedule end date for the housekeeping to end their Duty | 21/11/2017 19:00:00 |

## **2.6** **User Interface Requirements**

A mock-up draft design has been made to show what will be developed. The mock-up consists of navigation and a user interface of the system.

|  |
| --- |
|  |
| This is the login page before any operation is taken place. This is needed to differentiate the different types of users with different access control. |

|  |  |
| --- | --- |
|  |  |
| This screen will show up when the **receptionist** logs into their account. The receptionist only has 3 different navigation headers.  The blue bar at the top of the page shows the system status and importance notices. | This is shown when the **receptionist** selects the “Booking” navigation. The user can select the month and the range of dates to see the booking table for reference. The management and administrative users can modify if necessary. This table will be taken from the MySQL database. The payment button is in red for the users to easily locate. |
|  | |
| When the “Payment” button is pressed, this pop up screen is shown. The **user** can choose between Card Payment or Cash Payment. | |

|  |
| --- |
|  |
| This is shown when the **user** selects the “Card Payment” button. By keying in the Booking ID and the Guest ID, the address and city would be generated and the Guest is able to pay. The **guest’s** credit card information will be hidden from the user’s interface. |
|  |
| This is shown when the user selects the “Cash Payment” button. The **user** will have to key in the the amount of cash handed to them by the guest and the balance will be automatically generated and shown in the textbox below it. |

|  |  |
| --- | --- |
|  |  |
| This screen is shown when the **receptionist** selects “Room availability”. This page allows the user to select a date and check the room status. | This will be shown when the **receptionist** selects “Report”. The user can select the date to generate the report for booking and room availability. Alternatively, the user can select to generate a general report which does not take the date selection in consideration. |

|  |
| --- |
|  |
| This is an example of what is shown when a **user** selects to generate a report. The content of the report varies on the different user level logged in. If a **management** or **administrative** users generates, statistics and graphs will be included in the report. They can either export the data, which will be saved in the computer hard drive, or send it to a printer for it to be printed. |

|  |
| --- |
|  |
| This is shown when the **receptionist** selects “Guest Record”, they can view all the details of the guest. Only the **management** and **administrative** users can create, edit or delete these records. |

|  |  |
| --- | --- |
|  |  |
| This is shown when the **management** users log into the system. They have 5 navigation headers at the top. The additional headers are “Staff management” and “Housekeeping” compared to the reception users. | This screen comes up when the **management** users selects on “Staff management”. They are able to create, view, edit and delete the records of the staff. They can also search for the staff. |

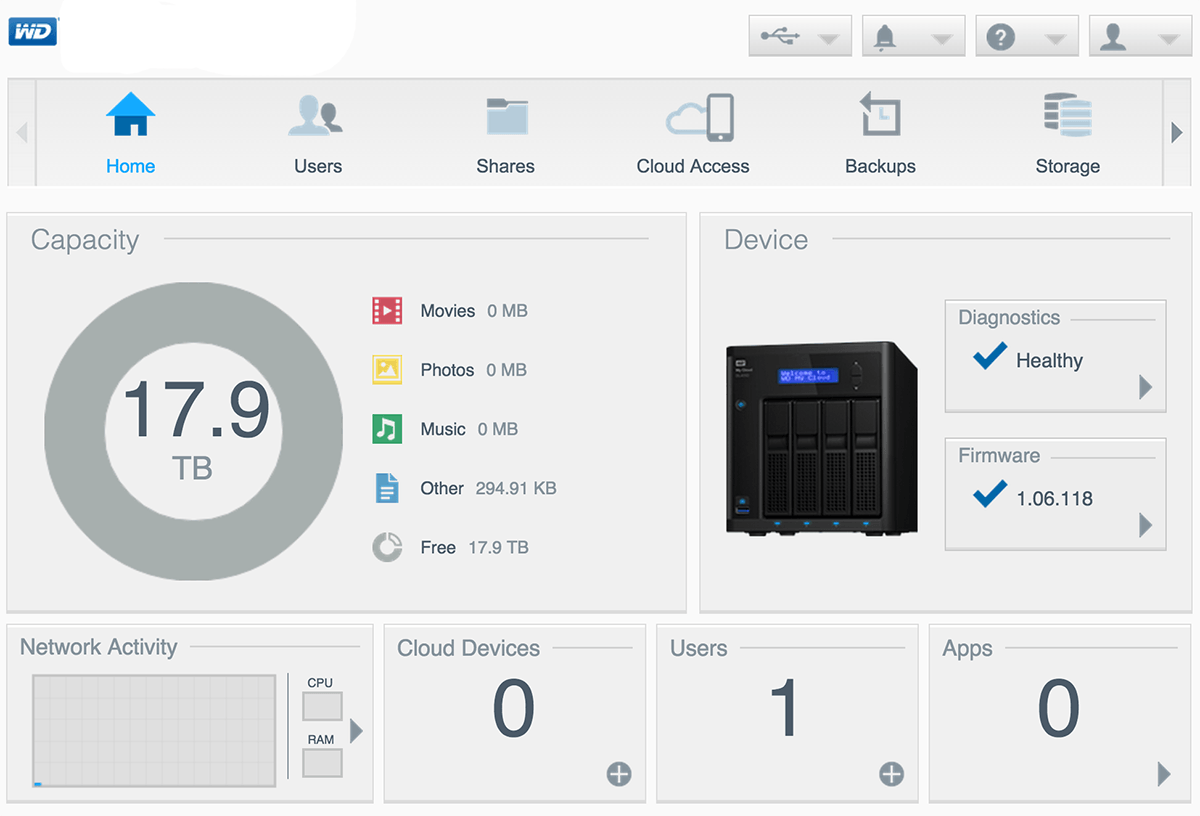
|  |
| --- |
|  |
| When the **management** users select “Housekeeping”, they can view the records for the duty types, staff ID and their schedule. The user can also search the records in the search box as well as create, edit and delete them. |

|  |  |
| --- | --- |
|  |  |
| When the **administrator** logs in, he views this page. The administrator can edit the System Status at the top left corner. Also, an additional navigation header, “Manage accounts” exists only for the administrator comparing to the management user accounts. | After selecting “Manage accounts”, this page is shown. Here, the **administrator** can search for records. Also, creating, editing and deleting of records are made available for the administrator. |

|  |  |
| --- | --- |
|  |  |
| When the **administrator** selects “Create Here”, this pop up shows up. Here, user account creation is done. After clicking “Create Account”, this new data is added to the database. | When the **administrator** selects “Edit”, the current record changes to textboxes. Editing of the records are made available then. The button “Edit” changes to “Save” for the user to click again to confirm the changes. |

|  |
| --- |
|  |
| The above is shown when the **administrator** selects “Delete”. A pop up is shown to confirm the deletion of the record. |

## **2.7** **Interface with Other Systems**



**NAS Interface:** The NAS device is used to perform backup and restoring of data (reports). During the process of backing up and data restoration, the user will have to interact with a different built-in interface of the NAS device called, “My Cloud OS”.

For security, the NAS provides a robust data protection by having multiple Redundant Array of Independent Disk (RAID) configuration which helps creates multiple copies of the data that is being backed up, into multiple hard drive in the RAID configuration. This means that if one hard drive were to fail, the backup data can still be retrieve from the other hard drives in the RAID. Volume encryption are also added for each hard drive, thus, increasing the security level of all data stored.

## **2.8** **Assumptions**

Delonix Regia reserves the right to re-evaluate the assumptions stated below.

* We assume that every booking must be scheduled by an adult that is of age 18 and above.
* We assume that the whole system will be in the English language.
* We assume that only the management and administrative staff can edit the guest records.
* We assume that the final software will not one hundred percent look like the mock-up.
* We assume that there will not be glitches and flaws while and after developing the software.
* We assume that the budget for this system will be enough to cover all costs.
* We assume that the client’s needs have been finalised and no more additions to it.
* We assume that every person involved in this project has the same end goal.

# **3** **OPERATIONAL AND QUALITY REQUIREMENTS**

## **3.1** **Operating Environment**

The system will be operating on a computer that resides in the reception area of the hotel. However, due to the low specifications of the computer they own, we highly suggest Mr Wang to upgrade to a computer that match with the specifications that the system will be running on.

The specifications are as such:

Operating System: **Windows 10**

Central Processing Unit: **7th Generation Intel Core i5-7400**

Random Access Memory: **8GB DDR4**

Hard Disk Drive: **1TB 7200 RPM**

Graphic Processing Unit: **Intel HD Graphic 630**

An example of an existing computer that matches with these specification is the Inspiron 3268 by Dell.

## **3.2** **Development Constraints**

**Hardware constraint**

The computers that the development team uses might malfunction during the development phase. This will increase project budget/expenses and also delay the completion of any project objectives.

**Software constraints**

The software used during the development phase might not be compatible with the data type/functions of the software that the team is developing.

**Schedule constraint**

Some objectives might not be completed in time, thus, potentially pushing back the deadline and release date of the software, which in turn, increase the project budget/expenses.

## **3.3** **Performance**

When setting a reasonable response time, we have to take into account what the user think is happening when they perform an action. For example, when the user click a button to generate a report that has a year worth of data, it is obvious to them that it will take a longer time to response as compared to searching for an item through the search bar. With this, we can roughly estimate what is an acceptable response time needed for each action.

|  |  |  |  |
| --- | --- | --- | --- |
| **Action** | **Description** | **On average** (seconds) | **Peak hours** (seconds) |
| Login | Response time of a successful login. | 2 ~ 2.5 | 3 ~ 3.5 |
| Booking Details | Response time of retrieving booking details. | 1 ~ 1.5 | 2 ~ 2.5 |
| Payment Details | Response time of retrieving payment details. | 1 ~ 1.5 | 2 ~ 2.5 |
| Room Availability | Response time of retrieving room status. | 1 ~ 1.5 | 2 ~ 2.5 |
| Room Availability Report | Response time of generating room status report. | 5 ~ 6 | 7 ~ 8 |
| Staff Details | Response time of retrieving staff details. | 1 ~ 1.5 | 2 ~ 2.5 |
| Housekeeping Details | Response time of retrieving housekeeping details. | 1 ~ 1.5 | 2 ~ 2.5 |
| Staff Account Creation | Response time of a successful account creation of staff. | 2 ~ 2.5 | 3 ~ 3.5 |
| Updating of Records | Response time of updating records. | 1 ~ 1.5 | 2 ~ 2.5 |
| Deleting of Records | Response time of deleting records | 1 ~ 1.5 | 2 ~ 2.5 |

## **3.4** **Availability**

System should be required to run 24 by 7 throughout the year. Backups should be done within 2am to 3am as it is the off-peak hours during the first Sunday of every month. As for maintenance, a downtime of 1.5 hours a week should be set aside. Downtime should not take too much time as it may badly affect business operation which in turn affect sales revenue. The maintenance should also be done during off-peak hours (2am - 4am) in order to avoid any clashes with normal operation of the system.

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

**Security Features**

1. The system will be protected by a user login screen that requires a name and password
2. The system can use Advanced Encryption Standard (AES) to provide confidentiality when transferring data.
3. The system can also use Elliptic Curve Cryptography to ensure authentication, integrity and non-repudiation service.
4. The System uses Industry standard 256-bit wildcard SSL booking page to protect sensitive customer data.
5. The system also includes tokenization to ensure card data is never able to be retrieved if the server is compromised.
6. The system will ensure pages that are members only, should be hidden from public, setting exclusiveness.

**Access Control**

The table shows which staff type has access to which features of the system.

|  |  |  |  |
| --- | --- | --- | --- |
| **Features** | **Reception** | **Management** | **Administration** |
| Booking reservation |  |  |  |
| Housekeeping |  |  |  |
| Room status report |  |  |  |
| Room occupancy report |  |  |  |
| Housekeeping report |  |  |  |
| User accounts |  |  |  |

# **4** **SPECIAL REQUIREMENTS**

* We require Mr and Mrs Wang to upgrade the computer on their reception area to the one that we recommended in the Operating Environment.
* We require constant communication with the stakeholders (system users and client) of this project.
* A working internet connection is needed 24/7 while developing and after it is developed for the users.

# **5** **REFERENCES**

Paras Pahari, Freelancer Programmer at Freelancing Pvt Ltd Follow. (2017, March 16). Hotel Management System SRS. Retrieved November 19, 2017, from <https://www.slideshare.net/ParasPahari/hotel-management-system-srs>

Best NAS Devices for Small Businesses. (2017, May 19). Retrieved November 19, 2017, from <https://www.storagecraft.com/blog/best-nas-servers-for-small-business/>

Security in an Online Reservation System. (2016, April 28). Retrieved November 19, 2017, from <http://www.otrams.com/blog/online/security-in-an-online-reservation-system>

Enzenhofer, K. (2017, July 26). What is the expected response time to deliver good user experience? Retrieved from <https://www.dynatrace.com/blog/what-is-the-expected-response-time-for-a-good-user-experience/>

How Much Does Custom Software Cost? (2017, September 07). Retrieved from <http://soltech.net/how-much-does-custom-software-cost/>

Borysowich, C. (2008, July 18). Lists of Assumptions & Constraints with a Sample. Retrieved from <http://it.toolbox.com/blogs/enterprise-solutions/lists-of-assumptions-constraints-with-a-sample-26073>

Temasek Polytechnic

School of Informatics and IT

**Diploma in Information Technology (IT)**

Software Design Specifications (DS)

**Project Particulars**

|  |  |
| --- | --- |
| **Tutor** | Qi Yutao |
| **Class** | P01 |
| **Project Title** | Delonix Regia Hotel Management System |

**Project Team’s Particulars**

|  |  |
| --- | --- |
| **Matric Number** | **Student Name** |
| **1602691F** | **Myron Low** |
| **1605809A** | **Zafrulla** |
| **1605894C** | **Zulhilmi** |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 18/11/2017 | 1.0 | Started and finalized the Architecture Design. | Zafrulla |
| 19/11/2017 | 1.5 | Started and Finalized the Database Design. | Zulhilmi |
| 20/11/2017 | 2.0 | Started and finalized User Interface Design. Started Program Design. | Zafrulla |
| 21/11/2017 | 2.5 | Finalized Program Design and formatting of the report.. | Zafrulla  Zulhilmi |

**Table of Contents**

1. DISTRIBUTION OF WORKLOAD. 4

2. ARCHITECTURE DESIGN. 4

3. USER INTERFACE (UI) DESIGN. 4

4. PROGRAM DESIGN. 4

5. DATABASE DESIGN. 4

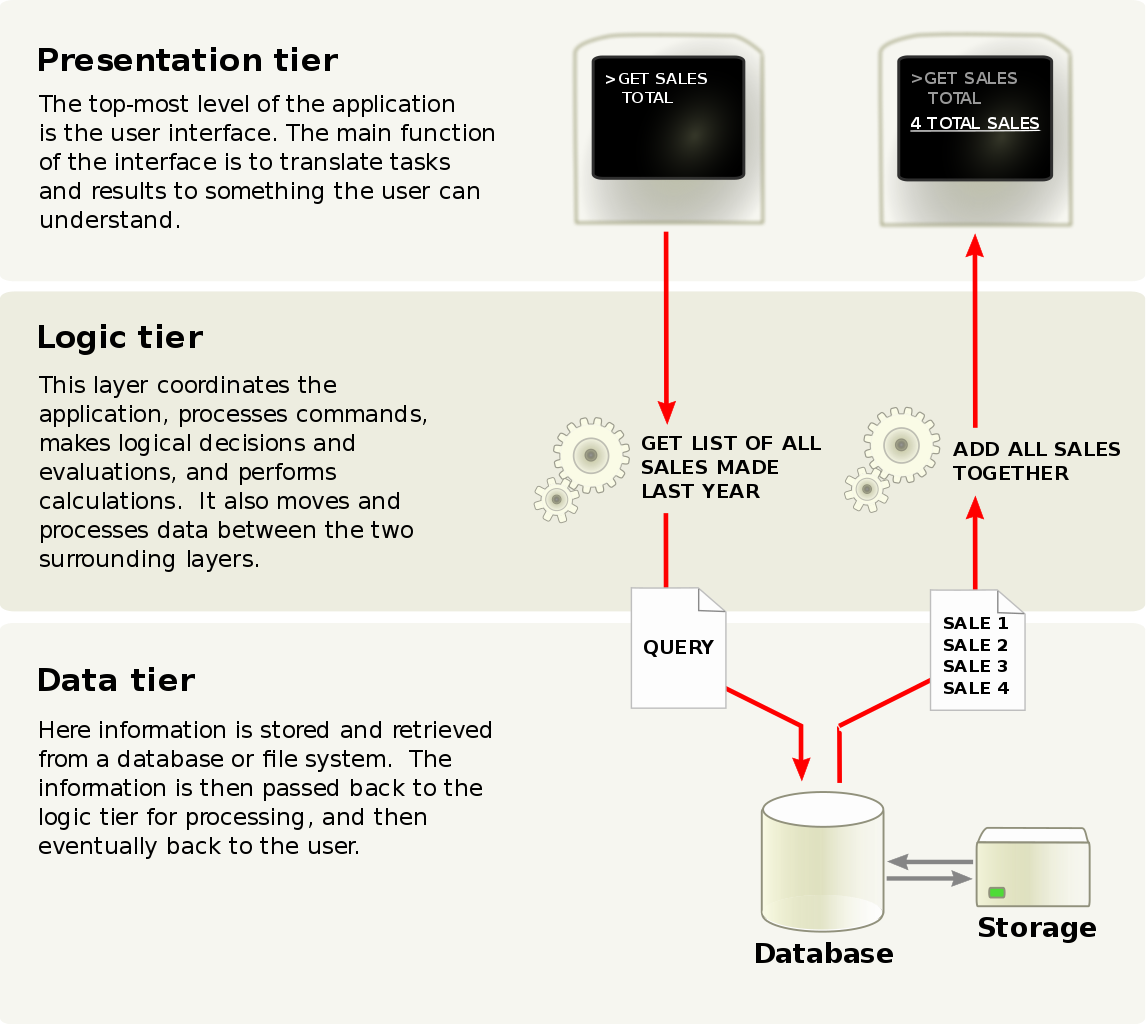
# **1.** **DISTRIBUTION OF WORKLOAD**

|  |  |
| --- | --- |
| **Design** | **Members** |
| * Architecture Design * User Interface Design * Program Design | Zafrulla |
| * Database Design | Zulhilmi |
| * NIL | Myron |

# 

# **2.** **ARCHITECTURE DESIGN**

The proposed system architecture design will be a 3-tier comprising of client-business-data.



**Client**

When using a software, whatever is shown in the screen is known as the presentation layer. As the computer will be at the front desk, the receptionist will be the one mainly using this computer. For example, the receptionist will be able to interact with the system through the user interface.

**Business**

This layer serves as the logic layer where it interacts with the database to retrieve the data. The business layer is needed as our system will have a login module. There needs to be a logic for the system to interact as there are different access controls for the different users; reception, management and administrative staff.

**Data**

This layer is known as the layer where the data resides. Operations are performed here such as insert, update and delete. Most of the data will be stored on a MySql database which can be accessed from a website called “000webhost”. The data stored in this database can be taken from Section 2.5 of the SRS.

**Conclusion**

The proposed 3 tier client-business-data system architecture design is needed compared to the 2 tiers. The 2 tiers only incorporate the clind-data model where there is no logic involved. This model would most probably work if there is no logic involved. For example, where the receptionist checks with the booking of the guests through the single computer provided. The single computer would not have conflicts when communicating with the database. However, as stated above, the login module is required for the access control. Thus, logic is needed and the 3 tier is highly suitable for it.

# **3.** **USER INTERFACE (UI) DESIGN**

We have chosen to develop a web application instead of a windows software because it requires an internet connection 24/7 to connect to the database. We have taken Jake Nielsen’s “Heuristics Evaluation” and LEMES (Learnability, Efficiency, Memorability, Error free and Satisfaction) while designing this system. The design will be improved consistently when developing.

|  |  |
| --- | --- |
|  | A simple login and minimalistic user interface when the user first opens up the application. This is applied so that there will be less confusion for the users with only two textboxes to focus on. Every textbox in this application also comprises of a placeholder too. This adds on to having the least confusion while using the system.  The button is in a separate colour to differentiate between the other elements in the page. A flat button design is applied for all the buttons in the system as it is the most commonly used type. |
|  | Once logged in, the system status and other important notices are made visible for all the users. Only the **administrator** can edit the system status as with the red button located at the top left.  The navigation is located at the top as most users are accustomed to navigating through pages from the top. |
|  | The “LOGOUT” button can be found at the top left of the page. This button will stay throughout the navigation. Every user will logout the same way. |
|  | When any user selects a page from the navigation header, the button will be highlighted in blue. This is to let the user know what page they are viewing at any point of time while using the system.  In the “Booking” page, the **receptionist** can view the booking table taken from the database. A “PAYMENT” button is also located in this page at the top right. The button is red in color to highlight the user’s attention.  The receptionist can also select the range of dates to view in the records. |
|  | Once the payment button is clicked, a pop up window will appear. The user can enter the Booking and Guest ID. Afterwards, they can select either “Card Payment” or “Cash Payment”. The two buttons are separated for clear indication.  The background will then be greyed-out. This implies that the buttons at the back are currently disabled. |
|  | If the user selected “Card Payment”, the price of the booking will be generated. Using the payment machine, the data will be generated into the other textboxes for the card details. The user of the system cannot view this data as it is private.  The “Confirm” button is in red at the bottom as with most forms have their buttons at the bottom. |
|  | If the user selected “Cash Payment”, two text boxes will be shown for the user to enter. The price will also be generated. Like the card payment form, the “Confirm” button is in red at the bottom. |
|  | In the “Room Availability” page, the **receptionist** can select a date to view the room status. The headers in this page, “Choose A Date:” and “Check Room :” acts as a guide for the user, instead of just adding the datetimepicker and data table into the page.  Only the **management** and **administrative** users can edit the room status. |
|  | In the “Reports” page, the **receptionist** can view only 3 types of reports to generate. The datetimepicker is used for the selection of date. The buttons are grouped together to represent its similarities.  The **management** and **administrative** users will be able to view more options for the reports such as the “Room Occupancy Report” and “Housekeeping Report”. |
|  | When the user selects to generate the report, a preview will be shown as stated “Preview” for the title. Also, the name and staff ID of the person generating the report will be included.  At the top, two buttons which are labelled as “Export” and “Print” is shown. This “Preview” title and button will be fixed at the top even though the report is scrolled down. |
|  | When the **receptionist** selects “Guest Record”, they can view all the guest’s record shown in a table format. A magnifying glass icon is used assuming that most of the users recognise the icon from different applications that they use.  Only the **management** and **administrative** users can create, edit and delete these records. |
|  | In the **management** user accounts, they have access to two more navigation headers compared to the reception staff accounts, “Staff Management” and “Housekeeping”.  The user can search for the staff record in the search box. They can also create staff records at the top right. Editing and deleting are also made available per row. This enhance the workflow precision. |
|  | Like the “Staff Management” page, “Housekeeping” shows a similar layout. This is applied so that the layout remains consistent. |
| The **administrative** users have an extra navigation which is “Manage accounts”. In this page, the administrator can create, view, edit and delete user accounts.  The buttons to edit and delete are displayed at every record to specify clearly.  At the top, the administrator can search in the search box for records or create user accounts. These two different features are separated with the word “OR” in bold. We have made it bold to capture the attention of the user informing of the separation between the features. | |
|  | When the **administrator** selects “Create Here”, a registration window pops up to create the account. Similar to the other forms, the button to create is located below and the textboxes include placeholders. |
|  | After selecting “Edit”, the button changes to “Save”. The button also has a different colour for the administrator to note the change of labels in the button.  Each record then changes to a textbox field for the user to key in. The user must select “Save” to update as to aid in making less mistakes |
|  | When the button “Delete” is selected, a message box comes up to confirm the deletion of record. This is put in place as to avoid any errors when dealing with sensitive information. |

# 

# **4.** **PROGRAM DESIGN**

**Use case: Login account**

**Description:** The user can select the textbox and key in their username and password. The user cannot create, edit, or delete any accounts in this use case. The user can attempt to sign into their account 3 times, afterwards, a notice gets sent to the admin.

**Actors:** Administrative users, Management users and Reception users.

Main flow  
1. This use case starts when the application is opened.

2. System displays login form.

3. Staff enters username and password.

3. Staff selects “Sign In” button.

4. System checks account information with database.

5. If login unsuccessful, go to unsuccessful login alternate flow.

7. System displays “Sign in Successful” end the use case ends here.

Unsuccessful login alternate flow  
5.1. System displays “Sign in Unsuccessful”.

5.2. Go back to step 3 in main flow.

**Use case: Payment of fees**

**Description:** The user can choose to select payment when the guest chooses to checkout. The user cannot edit the guest’s card details as well as the price of the booking. The guest will not be able to view the system too.

**Actors:** Administrative users, Management users and Reception users.

Main flow  
1. This use case starts after the user logs into the system.

2. System displays the home page.

3. User selects “Booking” tab.

4. System displays Booking page.

5. User selects “PAYMENT” button.

6. System displays form for payment.

7. User enters booking and guest ID.

8. If Guest chooses to pay by card, go to card payment alternate flow.

9. User selects “Cash payment”.

10. System displays price and form for booking.

11. User enters “Cash handed” based on amount guest gave.

12. System displays balance for cash.

13. User selects “Confirm” and the use case ends here

Card payment alternate flow

8.1. User selects “Card Payment”.

8.2. System displays form for card payment.  
8.2. Guest scans card.

8.3. System displays filled text boxes.

8.4. User selects “Confirm” and the use case ends here.

**Use case: Check Booking**

**Description:** The reception staff can use this booking page to view the different booking table. They are not allowed to edit any booking. The management and administrative staff however can edit the bookings.

**Actors:** Reception users

Main flow  
1. This use case starts when the user selects “Booking”.

2. User selects drop down box.

3. System displays the list of months.

4. User selects month.

5. System displays results for selected month.

6. User keys in range of dates.

7. System displays booking table for selected dates and month and the use case ends here.

**Use case: Check Room Availability**

**Description:** The reception staff can use this room availability page to view the status of the room. Only the management and administrative accounts can edit the records.

**Actors:** Reception users

Main flow  
1. This use case starts when the user selects “Room Availability”.

2. User selects date from datetimepicker.

3. System displays room availability table for selected date and the use case ends here.

**Use case: View Report**

**Description:** The reception staff can view and generate only 3 different types of report. They can also select the dates. However, the datetimepicker cannot be edited or deleted. The management and administrative staff have access to more than the 3 reports. They can also view statistics in graph or pie chart format.

**Actors:** Reception users

Main flow  
1. This use case starts when the user selects “Reports”.

2. If the user chooses to select “Generate Booking Report” or “Generate Room Availability Report”, go to other reports alternate flow.

3. User selects “Generate General Report”.

4. System displays preview of report.

5. User selects “Print Report” and the use case ends here.

Other reports alternate flow

2.1. User selects month or day to view report.

2.2. User selects “Generate Booking Report” or “Generate Room Availability Report”.

2.3. Go back to step 4 in main flow.

**Use case: Manage Staff Records  
Description:** The management and administrative staff can access this page. The reception staff has no access. The management and administrative staff can create, view, edit and delete the records.

**Actors:** Administrative users, Management users

Main flow  
1. This use case starts when the user selects “Staff management”.

2. If user chooses to create a staff record, go to create staff alternate flow.

3. If user chooses to edit a staff record, go to edit staff alternate flow.

4. If user chooses to delete a staff record, go to delete staff alternate flow.

5. User selects search box.

5. User keys in staff data in search box.

7. System displays the record and the use case ends here.

Create staff alternate flow  
2.1. User selects “Create Staff Record”  
2.2. System displays a pop up form.  
2.3. User keys in information of new staff.

2.4. User selects “Create” button and the use case ends here.

Edit staff alternate flow  
3.1. User selects “Edit” button in the record.

3.2. System displays text boxes to edit the data.

3.3. User edits the data in text boxes.

3.4. User selects “Save” button.

3.5. System displays new data and the use case ends here.

Delete staff alternate flow  
3.1. User selects “Delete” button in the record.

3.2. System displays message box to confirm deletion.

3.3. User selects on “Confirm”.

3.4. System deletes record, displays every other record and the use case ends here.

**Use case: Manage Housekeeping  
Description:** The reception staff has no access to this page. The management and administrative staff can view and edit the records.

**Actors:** Administrative users, Management users

Main flow  
1. This use case starts when the user selects on “Housekeeping”.

2. If user chooses to search for record, go to search housekeeping alternate flow.

3. User selects “Edit” button in the record.

4. System displays drop down list for duty type.

5. User updates the room status.

6. User selects “Save” button.

7. System displays updated room status and the use case ends here.

Search housekeeping alternate flow  
2.2. User selects search box.

2.3. User keys in room information.

2.4. System displays searched information.

2.5. Go back to step 3 in the main flow.

**Use case: Manage account  
Description:** The reception and management users has no access to this page. Only the administrative staff can create, view, edit and delete accounts.

**Actors:** Administrative users

Main flow  
1. This use case starts after the user selects “Manage accounts”.

2. If the user chooses to create an account, go to create account alternate flow.

3. If the user chooses to edit an account, go to edit account alternate flow.

4. If the user choses to delete an account, go to delete account alternate flow.

5. User selects search box.

6. User keys in account information.

7. System displays searched record and the use case ends here.

Create account alternate flow  
2.1. User selects “Create Here”  
2.2. System displays a pop up form.  
2.3. User keys in information of new account.

2.4. User selects “Create” button and the use case ends here.

Edit account alternate flow  
3.1. User selects “Edit” button in the record.

3.2. System displays text boxes to edit the data.

3.3. User edits the data in text boxes.

3.4. User selects “Save” button.

3.5. System displays new account data and the use case ends here.

Delete account alternate flow  
3.1. User selects “Delete” button in the record.

3.2. System displays message box to confirm deletion.

3.3. User selects on “Confirm”.

3.4. System deletes selected record, displays every other record and the use case ends here.

# **5.** **DATABASE DESIGN**

When designing the database, we decided to separate all the data involving the Staff of the hotel and the Guest of the hotel. The tables that involves the Staff, are the Staff, Housekeeping and Staff Account. Whereas for Guest, the Guest, Payment, Booking and Room tables are involved.

**Staff** (Staff\_ID, Staff\_FirstName, Staff\_LastName, Staff\_DOB, Staff\_NRIC,

Staff\_Address, Staff\_BankAcctNo, Staff\_PhoneNo)

**Staff Account** (Account\_ID, Staff\_ID, Staff\_Username, Staff\_Password, Staff\_Type)

**Housekeeping** (Duty\_ID, Duty\_Type, Staff\_ID, Schedule\_StartDate, Schedule\_EndDate)

**Guest** (Guest\_ID, Guest\_FirstName, Guest\_LastName, Guest\_Address,

Guest\_Email, Guest\_PhoneNo)

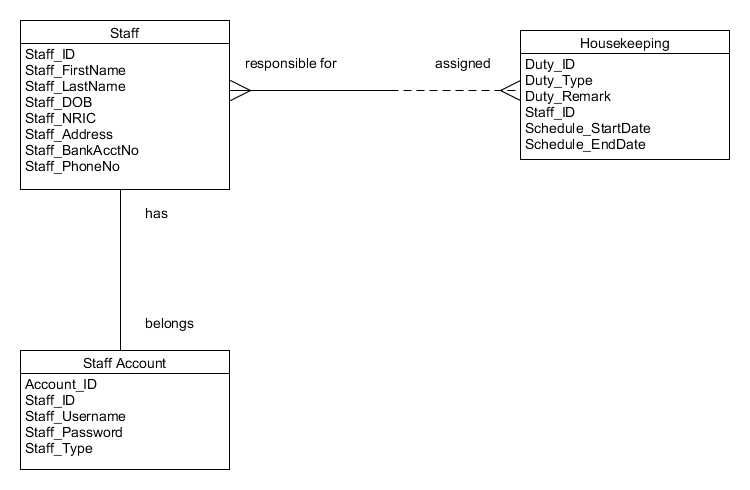
**Payment** (Payment\_ID, Guest\_ID, Credit\_Card\_No, Card\_FullName, Card\_ExpireDate)

**Booking** (Booking\_ID, Room\_No, Room\_Remark, Guest\_ID, CheckIn\_Date,

CheckOut\_Date, Number\_Of\_Adults, Number\_Of\_Children)

**Room** (Room\_No, Room\_Type, Room\_Status)

**Staff**



**Guest**

