Temasek Polytechnic
School of Informatics and IT

Diploma in Information Technology (IT)

Terms of Reference

Project Particulars

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

Project Team's Particulars

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

1. Introduction (Xin Yi)

Mr. and Mrs. Wang have been managing the hotel, Delonix Regia, for 20 years. Their business was good until the recent years. Today, most of the hotels are available for online booking which brings convenience to the customer by booking rooms online with a few clicks. However, Mr. and Mrs. Wang are unsure of how to handle the computer system and therefore, their hotel is only available for walk-in guests. As all the data, such as guest information, check-in & checkout records, etc. are manually recorded in the files, it is less secure and easily mess up with the records. Mr Wang tends to record the data to the wrong files, and sometimes, they make a loss for mistook guest who has made the payment. In addition, there are many competitors in the market and they would like to improve on their business growth and revenue. Therefore, they decided to implement a software system for their hotel to improve efficiency and reduced human errors.

2. Objectives of the Project (Joseph)

The objective of this project is to provide an efficient Hotel Management System (HMS) for Delonix Regia which is being managed by Mr. and Mrs. Wang. The goal is to develop a management system which contains an easy-to-understand Graphical User Interface (GUI) which will make it effortless for the hotel employees to carry out daily booking operations. Furthermore, the system must be able to display real-time accessibility of room, rates, shifts and many more. Also, the system must be robust and any form of bugs or glitches must be strictly resolved before the actual deployment of the system.

The summary of the objective for this project are as follows:

- Develop an efficient Hotel Management System (HMS)
- Develop and implement an uncomplicated GUI for the HMS
- Ensure system produce real-time information (E.g. Price rates)
- Ensure the system is robust
- Resolve any bugs or glitches before actual deployment

3. Scope of the Project (Xin Yi)

Guest Management: Record of guest information and stay
Reservation: Allows for online and on-site booking, showing the available room types and stay package
Payment Management: This works as payment from guest using credit/debit card.
Housekeeping Management: records of staff details, streamline responsibility distribution of staff while providing real time visibility to staff
Review: Allow for customer feedback to improve customer satisfaction
Report Management: Generation of weekly/monthly reports which offer a performance update for analysis
Inventory Management: Records of inventory, ensuring sufficient inventory is accessible at all times

4. Distribution of Workload

Determine which members of the team will be responsible for what areas of project work. Individual's responsibilities should be clearly spelt out.

Objectives/Deliverables	**Members**
TOR (Terms of Reference): Constraints Resources PP (Project Plan): Assumptions and Constraints Budget Summary	Nicholas
TOR: Introduction Scope of the Project PP: Objectives and scope of the project Risk Management Plan	Xin Yi
TOR: Objectives of the Project Product/Positioning in the Market/Company PP: Project Schedule	Joseph
TOR: Distribution of Workload Approach and Methodology of the Project PP: Roles and Responsibilities Work Breakdown Structure	Ron

5. Constraints (Nicholas)

The definition of constraint is a limitation or restriction

- limiting factor: something that limits freedom of action
- state of restriction: a state in which freedom of action is severely restricted

There are primarily 5 types of constraints, they are deadline, time, funds, manpower and the software technical issues.

Time

- Different timetable
- No common time to meet for everyone
- Insufficient time during each meet up
- Lab opening hours

Deadline

- Fail to meet deadline
- Other assignments or projects to do as well
- Project takes longer than expected
- New requirements added at the end
- There is only 8 weeks

Funds

- Insufficient funds
- Highly skill set needed
- Project dragged over deadline causing a surge in funds needed
- Hard to give a rough estimate of funds needed

Manpower

- Not enough manpower to cover all the requirements
- Project is very complex and need more personnel
- There is only a group of 4 people

Technical

- Sabotaging due to political agenda
- Hard to change requirements
- Difficult to pass on to the next group of teams
- No indentation difficult to understand the software

6. Resources (Nicholas)

The project is not really finalized and may change in future due to new requirements or a constraint requiring the project to change its software/hardware. Assuming that the team will be using Windows to develop the software. If it is a software for the phone, we will only be targeting the iOS and Android OS as there are the most popular ones in the market.

For now some of the software and hardware will be, (at the bare minimum)
For Computers running Windows

- Dual-core 64-bit processor
- 8 GB of memory
- Up to 24 GB of internal storage (Kony Visualizer: 4GB, Android SDK: 2GB, Windows SDK: 4GB, BlackBerry NDK: 4GB, plus ample space for multiple complex projects)
- Network interface card
- Windows 10, Windows 8.1 Update, Windows 8, and Windows 7

IOS
Developing apps for iOS includes the following requirements.

- iOS-related code and applications can only be developed on a Mac OS computer
- The latest version of Xcode, the Apple SDK for creating iOS apps

Android OS
Developing apps for Android includes the following requirements.

- The Android SDK
- Apache Ant (Another Neat Tool), an open-source tool that automates aspects of the Android build process.
- Gradle (An advanced build toolkit that manages dependencies and allows to define custom build logic.)

Other softwares to use

- Google Docs (Capture requirements)
- Microsoft Office (Capture requirements)
- Visual Source Safe (Base line Capture requirements)
- Share Point (Baseline Capture requirements)
- Structure 101 (Creating enforceable architecture)
- Sonargraph (Creating enforceable architecture)
- Maven (tool for basic project structure)
- Eclipse (Start coding)
- JUnit (Start coding)
- Emma (Start coding)
- Jetty (Start coding)
- Tomcat (Start coding)
- Sonar (Start coding)
- Quality Centre (testing)

Hardware

- Minimum of one Windows PC to each member
- Printer
- Scanner
- Server

7. Product Positioning in the Market/Company (Joseph)

The Guest Management System (GMS) keeps all the information of the guest and stay. Compared to other GMS which requires manually input, this system enables the staff to keep track of real-time information. For example when the guest checks out, the system will automatically free up the room that has previously been occupied by the guest. This GMS have a search function that are able to search for specific guest swiftly.

The Hotel Reservation System (HRS) consists of 2 approach which is online booking and on-site booking. As most guests are tourists, some of them might make an online reservation back in their own country or some tourists may perform an on-site reservation due to the nick of time. This serves more efficiency and flexibility for the guest compared to other HRS which only provides on-site booking. This HRS are able to display real-time available room types (E.g. deluxe) and stay package on Delonix Regia's website.

The House Keeping Management System (HKMS) provide records of staff details and streamline distribution of staff's responsibility while providing real time visibility to staff. This ensures that room service will always be provided to the guest without any time-delay. Comparing to other HKMS which do not provide distribution of staff's responsibility, this system provides real time staff information such as staff's working shift and indicates the staff's roles and duty.

Delonix Regia's website also provides reviews for feedback purposes. This is to ensure that Delonix Regia's service are always top in mind. Reviews are crucial to the hotel business as it provides an assurance and reliability to the guest that are looking for a hotel to stay in. With tons of good reviews of the hotel will ultimately increase the revenue and the popularity of the hotel. However, some hotels do not provide reviews which will affect the business revenue in the long run.

Report Management System (RMS) is crucial for the hotel business as it provides a summary of the hotel's performance on a daily or monthly basis. This RMS are able to filter report based on different categories such as expenses report, revenue report, number of guest report and many more while other hotel's RMS only provides a summary report which does not provide much visibility. The RMS report can be customizable and filtered depending on the staff's needs. For instance, the staff can combine the expenses and revenue report as one single report.

8. Approach and Methodology of the Project (Ron)

The development model we have chosen to adopt would be the Prototype Model, more specifically Evolutionary/Breadboard Prototyping.

Evolutionary Prototyping, also known as breadboard prototyping, is the development of a minimal functionality software prototype with the initially understood requirements, adding more functionalities when the other requirements are understood. The process of this prototyping type is as such: Basic Requirement Identification > Initial Prototype Development > Prototype Review > Prototype Revision and Enhancement, allowing for the development of a completed, high-quality software which meets or exceeds the requirements and expectations of customers as a high user involvement level is present, allowing for ease of requirement understanding via frequent feedback.

This is further beneficial as it is unclear as to whether Mr. and Mrs. Wang will be able to effectively portray and communicate their requirements during requirement analysis and design, allowing for us to receive reviews to better understand and possibly uncover more/change requirements and further develop the software incrementally. Furthermore, as each prototype is a working but limited version of the software, it will be easier for them to have an understanding of each prototype.

However, there are risks, with any other models, that come with this approach. Firstly, it holds a risk of increasing system complexity as the scope may widen/change during process incrementations. This can pose problems as the cost and difficulty increases in relation to the amount of incrementations, making changes in the later parts of the life cycle undesirable. For this risk, the setting of a cut-off date/iteration must be set to mark the end of the Prototype Revision and Enhancement, requiring the agreement of both the clients and development team.

By doing so, both parties will be aware of when enhancements/changes can and cannot be further added. This date can be set according to the duration of the project, 8 weeks, and the duration it would take to implement all functionality and enhancements into a complete and deployable software application. Given that only 5 weeks remain, an estimation of 1 week would be required to finalize the application software, setting the official cut-off date/iteration in the 6th week of the 8 week project span while setting the 7th week as a contingency, allowing for the pushing of enhancement requests by the end of the 6th week, also allowing for 3 weeks of iterations, and setting the 7th week as buffer time should any critical changes need be made.

Additionally, if the development team is willing, a change request form system can be provided to the client to list down any further requests they might have after the cut-off date/iteration, which will allow for further enhancement of the application after deployment after consideration and prioritization of the changes listed.

Secondly, estimation, planning, and managing prototyping projects might serve as another problem due to the lack of regular deliverables. This problem can be managed by setting cut-off dates, similar to that for the first problem. Furthermore, the duration set for each deliverable can be broadened into weeks rather than days, requiring the completion of a deliverable to be done within a range rather than a fixed date, simplifying estimations, planning, and management. Additionally, for management, a defect list can be used to manage any defects in the application found during development, allowing for easier communication and resolving.

Lastly, group and client meetings should be planned and facilitated weekly for discussion on the project to allow for further communication, allowing for clarification on progress, defects, and changes related to the project.

9. References

(Ron)
https://www.tutorialspoint.com/sdlc/sdlc_overview.htm