**Diploma in Information Technology (IT)**

**Project Particulars**

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

**Project Team’s Particulars**

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

Description of Software Development Life Cycle and Software Development Models

**Zulhilmi**

The **SDLC** is the process required for every software project when creating a software. The process contains a detailed description on how to develop, maintain, replace, reform or update the software. This ensures that a good software is created. Examples of SDLC models are, Prototype model, Iterative model and RAD model.

The idea behind the **Prototype** model is that instead of finalizing the requirements of a software before proceeding to code it out, a throwaway mock-up of the software is built to further understand the actual requirements that are needed. The mock-up will be built based on the available requirements that are known. By interacting with the mock-up, the client will get a better understanding of the system itself so that they can improve on the requirements in order to achieve their desired system.

Instead of developing the full specification of the requirements, an **Iterative** model only implements some parts of the software, which will then be reviewed and improved on. This process will then be repeated, thus, creating a newer version of the software for each iteration until they achieve the desired system.

As for **RAD** model, all components or functions are divided into smaller, more easily managed modules and created in parallel of each other. The completed modules will then be delivered and assembled into a working mock-up of the software. This allows customers to interact with a working function and encourages them to provide feedback regarding them.

**Zafrulla**

**SDLC** is the process of developing a software from start to end. This includes the requirements of engineering, analysis and design, implementation, testing and deployment. Three such models are the Waterfall model, V-shaped model and Agile model.

The **Waterfall** model is a flow of stages which is perceived as going downwards throughout the development of the software. It is critical that the previous phase must be achieved before the next one starts. This model is mostly known to be used in development of softwares as it is the initial proposition. Each stage has its verification to find errors or conflicts but it is tough to go back to the previous stages after it has been completed.

The **V-Model** is similar to the waterfall model as it is known as an add-on to its predecessor. Instead of going down the “waterfall way”, the methods move upwards to appear as a V shape. The model is simple and easy to use but it is difficult to adjust the scope and it is expensive. This model also operates better when the demands are simply understood. The model also verifies and validates the product in the primary stages of its development. It also has early test planning which is the key contrast to the Waterfall model.

The **Agile** model is based on an incremental model. Incremental, fast cycles are used to develop the softwares. Customers, developers and testers are always interacting with one another. The design and technicality are constantly being monitored as to ensure quality results. However, to follow this model, it requires certain skills for the team. In its least possible time, the product software is of high quality and the customer would be satisfied.

**Myron**

The SDLC is a conceptual model used in [project management](http://searchcio-midmarket.techtarget.com/definition/project-management) that describes the stages involved in an information system development project, from an initial feasibility study through maintenance of the completed application. 3 examples of SDLC models would be the waterfall model, spiral model and agile model.

[**Waterfall**](https://www.roberthalf.com/blog/salaries-and-skills/waterfall-methodology-101-the-pros-and-cons) is the oldest and most straightforward of the SDLC models. It finishes one phase, then moves on to the next. However, there is no going back. Each stage relies on information from the previous stage and has its own project plan. It is easy to understand and simple to manage but early delays can throw off the entire project timeline. And since there is little to no room for revisions once a stage is completed, problems can’t be fixed until you get to the maintenance stage. This model doesn’t work well if flexibility is needed or if the project is long term and ongoing.

One of the most flexible SDLC models, the **Spiral** model is like the Iterative model and its repetition. The project passes through four phases over and over in a “spiral” until completed, allowing for multiple rounds of refinement. This model allows for the building of a highly customized product, and user feedback can be incorporated from the beginning of the project. However, the risk is that it might create a never-ending spiral for a project that goes on and on.

By breaking the product into cycles, the **Agile** model quickly delivers a working product and is considered a very realistic development approach. The model produces ongoing releases, each with small, incremental changes from the previous release. At each iteration, the product is tested. This model emphasizes interaction, as the customers, developers and testers work together throughout the project. But since this model depends heavily on customer interaction, the project can head the wrong way if the customer is not clear on the direction he or she wants to go.

**Terms of Reference**

**1. Introduction**

This project has been assigned by Pantheon Systems as part of our Student Internship Programme. The hotel owners Delonix Regia, Mr. and Mrs. Wang, has notified us that their hotel is not doing very well in spite of being a in relatively good district. The application that has been requested for us to create should be able to aid the hotel in upgrading to a better hotel management system than the one they have now. We believe that there are a lot of potential customers around the hotel’s district. However, due to the bad hotel management system that is in place, the business is not doing so well in terms of providing a good customer experience. Hence, we believe this system is needed in order to improve the management of the hotel.

**2. Objectives of the Project**

The objectives of this project is to provide a fully functional automated Hotel Management System that is reliable with the least possible error. The software is targeted towards the staff of the hotel. It should be able to keep records of customers and staff, manage the movement and storage of materials available, manage front desk requests, manage unpaid invoices and manage housekeeping duties. The software should require the least number of steps needed to perform each action so that the users will not have a difficult time learning and getting used to the software. The feedback from customer can be used to fix the problems that the customer mentioned so that the software can be improved via the Agile development model.

**3. Scope of the Project**

**Customer Record Module:** This module would store data of guest by assigning each guest a unique customer ID after they sign up. This ID would store their name, email, handphone number, room ID, checkin information etc.

**Staff Management Module:** This module would store data of the staff working there and have their information of their ID, name, age, attendance, salary and date they joined the company etc.

**Material Management Module:** This module administers the materials in association with the hotel. It involves the movement and storage of these materials. Without this module, it allows conflicts such as failure to supply goods which in turn will affect the business.

**Front Desk Management Module:** This module administers the customers needs and requests. Examples of this would be guest check in and out, early booking of event space or facilities. It also includes roles to make basic arrangements such adding an extra bed. Lastly the module handles the prices for different guests such as corporate prices and promotions.

**Account Receivable Module:** This module will help manage all the revenue transactions occurring within the hotel and also display detailed reports regarding those transactions. Receivables are unpaid invoices that the hotel has to collect after providing a service to the customer. This module can help to edit, delete, update all the unpaid invoices.

**Housekeeping Module:** This module will help manage the cleanliness of the hotel, such as, rooms, toilets, building, etc. This can help housekeeping staff view their cleaning schedule and which room requested for housekeeping.

**4. Distribution of Workload**

|  |  |
| --- | --- |
| **Objectives/Deliverables** | **Members** |
| Scope: Guest Record & Staff Management  Constraints  Roles and Responsibility | Myron |
| Scope: Account Receivable & Housekeeping  Resources  Approach and Methodology of the Project  Definitions and Acronyms  Assumptions  Budget Summary  Risk Management Plan | Zulhilmi |
| Introduction  Objectives  Scope: Material Management & Front Desk  Product Positioning in the Market/Company  Work Breakdown Structure  Project Schedule | Zafrulla |

**5. Constraints**

1. Issues regarding resources, such as, notebooks not functioning properly due to hardware or software malfunction.
2. Not enough staff to hire for the project.
3. Not enough software testers to prove the integrity of the software.
4. Not being able to meet the deadline of the project.
5. Not enough budget to support the whole development phase of the project.
6. Not being able to meet the requirements of the customer.

**6. Resources**

Hardware: 6 Notebooks (Asus Zenbook UX330US)

Operating Systems: Windows 10 Pro

Software: Android Studios, Notepad++, Xcode

Server: Amazon S3

Database: Amazon RDS

**7. Product Positioning in the Market/Company**

The application we are proposing is for the staff of Delonix Regia. The staff using the application will consists those from the back end to the front end. The goal of this application is to have an all-in-one service where the staff is able to efficiently make use of the application.

The application “roomMaster” by InnQuest provides a hotel management system for all sizes. It is recognised and has been in the market for a long time. However, it has a messy user interface. This makes their clients shy away as most would rather not use an application that confuses them. Also, their reporting function is not operating to its full potential. This makes their clients rely less on the application.

Our application will be simple to use so employees will be less confused. The buttons will be labelled clearly so that each department can use fluidly. Our application is also unique in the sense that it will generate graphs and pie charts for monthly and annual reports. These reports can be selected to be emailed to specific accounts and it can be opened in Microsoft Excel or other spreadsheet document.

Each module will have various functions such as scheduling and assigning an employee to name a few. Another unique feature that we are integrating is a column to add in remarks. This will greatly enhance the efficiency as the workload is not solely limited to the functions provided. As we have improved on the cons from the other application, it will have a standing in the market.

**8. Approach and Methodology of the Project**

For this project, our team will be adopting the Agile Model. With Agile Model, we will first plan on the features of the software and how to build it. We will then proceed to build and launch a fully working software. By getting feedback from customers, we then repeat the whole process again while tackling the issues mentioned in customer’s feedback.

A potential problem we might occur is the lack of necessary designing and documentation of the software during the beginning of the software development cycle. A way to overcome this is to have a proper plan and constant interaction with the customer to make sure the design and documentation are within their expectations.

Another potential problem is that the entire project can go haywire if the customer is not clear on what they want their final desired outcome to be. A way to overcome this is to help the customer understand what they really want so that they can be clear on their objectives.

Lastly, any decisions during the development phase can only be decided by senior/experienced programmers, thus, inexperienced programmers might make bad decision that will compromise the whole project. A way to overcome this is to assign senior/experienced programmers to make decisions or combine experienced and inexperienced programmers together.

**Project Plan**

1 **Introduction**

This project has been assigned by Pantheon Systems as part of our Student Internship Programme. The hotel owners Delonix Regia, Mr. and Mrs. Wang, has notified us that their hotel is not doing very well in spite of being a in relatively good district. The application that has been requested for us to create should be able to aid the hotel in upgrading to a better hotel management system than the one they have now. We believe that there are a lot of potential customers around the hotel’s district. However, due to the bad hotel management system that is in place, the business is not doing so well in terms of providing a good customer experience. Hence, we believe this system is needed in order to improve the management of the hotel.

1.1 **Objectives and scope of the project**

The objectives of this project is to provide a fully functional automated Hotel Management System that is reliable with the least possible error. The software is targeted towards the staff of the hotel. It should be able to keep records of customers and staff, manage the movement and storage of materials available, manage front desk requests, manage unpaid invoices and manage housekeeping duties. The software should require the least number of steps needed to perform each action so that the users will not have a difficult time learning and getting used to the software. The feedbacks from customer can be used to fix the problems that the customer mentioned so that the software can be improved via the Agile development model.

**Customer Record Module:** This module would store data of customer by assigning each guest a unique customer ID after they sign up. This ID would store their name, email, handphone number etc.

**Staff Management Module:** This module would store data of the staff working there and have their information of their ID, name, age, attendance, salary and date they joined the company etc.

**Material Management Module:** This module administers the materials in association with the hotel. It involves the movement and storage of these materials. Without this module, it allows conflicts such as failure to supply goods which in turn will affect the business.

**Front Desk Management Module:** This module administers the customers needs and requests. Examples of this would be guest check in and out, early booking of event space or facilities. It also includes roles to make basic arrangements such adding an extra bed. Lastly the module handles the prices for different guests such as corporate prices and promotions.

**Account Receivable Module:** This module will help manage all the revenue transactions occurring within the hotel and also display detailed reports regarding those transactions. Receivables are unpaid invoices that the hotel has to collect after providing a service to the customer. This module can help to edit, delete, update all the unpaid invoices.

**Housekeeping Module:** This module will help manage the cleanliness of the hotel, such as, rooms, toilets, building, etc. This can help housekeeping staff view their cleaning schedule and which room requested for housekeeping.

1.2 **Assumptions and constraints**

The hotel is located at a reasonably good district, however, it still isn’t doing well. We assume that there is sufficient customers around but due to the poor hotel management system that is in place, not many customers are satisfied with their services, hence, the lack of growth for their business.

We assume that the hotel allow most of their sales and services to be on credit. This gives the customers the ability to pay only after receiving/using their services/product. This helps to avoid any inconvenience when performing physical payment during each transaction.

The constraint we might face are:

1. Issues regarding resources, such as, notebooks not functioning properly due to hardware or software malfunction.
2. Not enough staff to hire for the project.
3. Not enough software testers to prove the integrity of the software.
4. Not being able to meet the deadline of the project.
5. Not enough budget to support the whole development phase of the project.
6. Not being able to meet the requirements of the customer.

1.3 **Definitions and acronyms**

1. RAD - Rapid Application Development
2. V-Model - Verification and Validation model
3. Amazon S3 - Amazon Simple Storage Service
4. Amazon RDS - Amazon Relational Database Service
5. UX Designers - User Experience Designer
6. DB - Database
7. PHP - Hypertext Preprocessor
8. SQL - Structured Query Language
9. SCCB - Software Change Control Board
10. SRS - Software Design Specification

2 **Roles and responsibilities**

|  |  |
| --- | --- |
| Role | Responsibility |
| Project Manager (Scrum Master)  (Zulhilmi) | * Facilitate requirements gathering. * Documenting of project requirements * Manage the team. |
| UX Designer (Zulhilmi) | * Analyze software requirement specification. * Convert the requirements into an easily understood user interface. |
| Application Developer (Zafrulla) | * Code the solution using Java, C#, Python, PHP, SQL language. |
| DB Administrator (Zafrulla) | * Create database following the standard steps of relational database designing. * Do modifications and maintain the database according to the requirements of the other developers. |
| Software Tester (Myron) | * Assure that business goals and objectives are satisfied. * Assure validations are implemented correctly. |

3 **Estimates and project schedule**

3.1 **Work breakdown structure**

Product Backlog - Product owners list all the features that are needed.(Requirements. Priority order)

1. Write SRS & User Stories
2. SRS & User Stories Review
3. SRS & User Stories Acceptance

Sprint Planning - Product owner, Scrum Master and Development team decide which user story has higher priority for next sprint(assign points to the story)

1. Assign points to User Stories
2. Rank User Stories based on points
3. Choose a set of User Stories to focus on

Sprint Backlog - The list of user stories that has been chosen to focus on. Everyone has a clear understanding about the “stories” based on discussion from Sprint planning.

1. Group the higher ranked stories together
2. Hold a stand up meeting to finalize the backlog

Sprint - 1 to 4 weeks cycle, worked till completion. But daily scrum occurs(what they did, what they’re doing and obstacles faced).

1. Scrum Master assigns tasks to team members
2. Each team member works on the tasks in relation to their role
3. Perform Daily Scrum

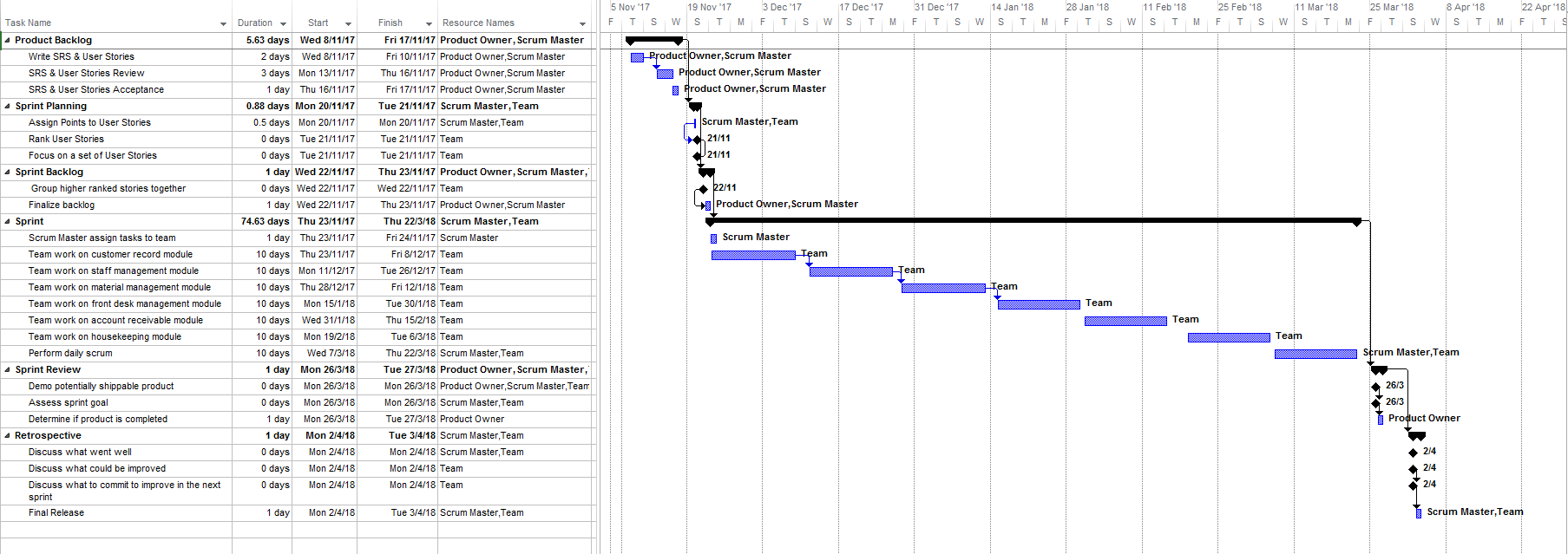
Sprint Review - Showcase the product to the Product Owner, can decide if want to add on anything or done.

1. Demonstrate potentially shippable product during meeting
2. Product Owner assess with the sprint goal
3. Product Owner determines if product is completed

Retrospective - Team brainstorms on what they can do to improve the process

1. Discuss what went well in the sprint
2. Discuss what could be improved
3. Discuss what to commit to improve in the next sprint

3.2 **Project Schedule**



3.3 **Budget Summary**

Project duration: 3 months

Project Manager & UX Designer Salary Per Month - SGD 6,977

6,977 \* 3 = SGD 20,931

Application Developer & DB Administrator Salary Per Month - SGD 3,692

3,692 \* 3 = SGD 11,076

Software Tester Salary Per Month - SGD 3,506

3,506 \* 3 = SGD 10,518

**Total manpower cost: SGD 42,525**

Asus Zenbook UX330US - SGD 1,698

1,698 \* 3 = SGD 5,094

Windows 10 Pro - SGD 399

399 \* 3 = SGD 1,197

**Total resource cost: SGD 6,291**

**Total cost: SGD 48,816**

**Estimated budget required: SGD 60,000**

4 **Risk Management Plan**

Point System

3 - High

2 - Medium

1- Low

|  |  |  |  |
| --- | --- | --- | --- |
| Type of Risk | Severity of Impact | Likelihood of Occurrence | Risk Reduction Strategies |
| Human error | 2 | 3 | Set up a team formation to do peer reviews so that they can check each other's work. |
| Unrealistic budget & deadline | 3 | 1 | Hire only the staff with the right technical skills needed and assign them to their expertise, this ensures that they are in their element. |
| Incompatible software/components | 3 | 1 | Review the software/component to make sure they are compatible with the tasks that needs to be performed |
| Developed functions does not match requirements | 3 | 2 | Development team has to understand what the customer really wants. Produce a mock-up of the function to show the customer before finalizing it. |
| User interface does not fit the needs | 2 | 2 | Create all the different type of scenarios that can happen when performing an action so that the designer can have a clear overview of what is needed. |
| Poor architecture, performance, quality | 3 | 2 | Perform a simulation of the architecture. Perform benchmarking on the performance. Tuning the quality of the function if needed. |
| Constant alteration of the requirements | 2 | 3 | Developers should increase the threshold for changes in their encapsulation (information hiding). The SCCB should be informed and decides whether or not the changes can be implemented. |

**References**

Pocket Your Shop Follow. (2013, April 16). 27 Examples Of Brilliant Mobile App Features For Hotels. Retrieved November 07, 2017, from <https://www.slideshare.net/PocketYourShop/27-examples-of-brilliant-mobile-app-features-for-hotels>

Six must-have features for a stellar hotel app. (2015, September 21). Retrieved November 07, 2017, from <https://ehotelier.com/insights/2015/09/21/six-must-have-features-for-a-stellar-hotel-app/>

Proposal of Hotel Management System.pdf. (n.d.). Retrieved November 07, 2017, from <https://www.scribd.com/doc/120029171/proposal-of-Hotel-Management-System-pdf>

Charitha Gamage, Associate Engineer Follow. (2014, September 14). Project Proposal document for Hotel Management System. Retrieved November 07, 2017, from <https://www.slideshare.net/thissagamage1/project-proposal-document-for-hotel-management-system>

Hotel management System Proposal - Documents. (2015, April 27). Retrieved November 07, 2017, from <https://docslide.us/documents/hotel-management-system-proposal.html>

The 5 Most Popular Hotel Management Software Solutions For Small Hotels Compared. (n.d.). Retrieved November 07, 2017, from <https://blog.capterra.com/the-5-most-popular-hotel-management-software-solutions-for-small-hotels-compared/>