Temasek Polytechnic
School of Informatics and IT

Diploma in Information Technology (IT)

Terms of Reference

Project Particulars

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

Project Team's Particulars

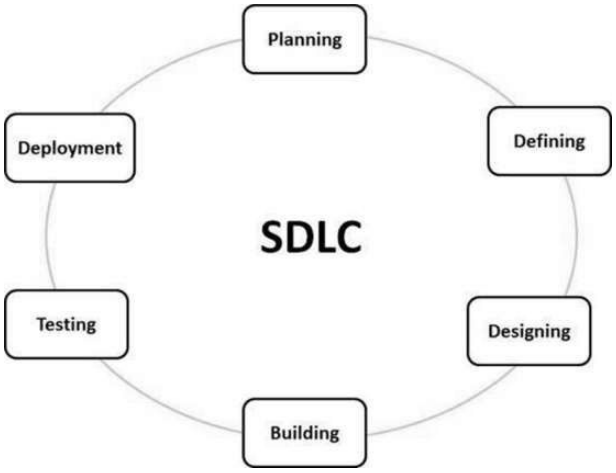
Matric Number	**Student Name**
1605028H	Ong Jia Hui
1600712B	Joanna Lim Min Le
1601992I	Cheong Ming Lun
1603305B	Toh Chien Yuan

Jia Hui's SDLC + 3 part model

A SDLC also known as the software development life cycle is a term used in systems engineering, information systems and software engineering to describe a process of creating a software through the 7 stages of SDLC. The steps for SDLC are identifying the problem, planning, designing, building, creating, testing, and deploying an information system.

SDLC works by lowering the cost of software development while simultaneously improving quality and shortening production time. SDLC achieve these goals by following a 7 stages plan that removes the typical pitfalls or mistakes that will be made to software development projects.

The plan starts by evaluating the existing systems that the company is using for any deficiencies which means to learn the strengths and weaknesses of the current system with any improvements as the goal. Next, would be the second stage, where the team will define the requirements of the new software and determine the cost and resources that will be required. It also details the risks involved and provides sub-plans for lightening those risks if applicable. A software requirement specification document is created. Then follow by the third stage of design specification. All the stakeholders will then review this plan and provide feedbacks and suggestions. In the fourth stage, the development stage of the SDLC, it develops the software by generating all the actual codes. Then in the fifth stage, the team will test for any defects and deficiencies. They will then fix those issues until the product meets the original specifications and their expectations. In the sixth stage, the deployment stage, it will be based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base and to check out if there are any defects here and there.



Waterfall Model:

Waterfall is the oldest and most straightforward of the structured SDLC methodologies — it is a sequential process, where one phase proceeds after another and cannot go back after each phase is done. Each phase also relies on information from the previous phase and has its own project plan. There are eight stages in this methodology which are conception, initiation, analysis, design, construction, testing, implementation and maintenance. This model doesn't work well if flexibility is needed or if the project is long term and ongoing because there are no room for any errors to be made in each step along the way.

Agile Model:

By breaking the product into cycles, the Agile model 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 hence it can proceed on without any unwanted errors.

This model emphasizes interaction, as the customers, developers and testers work together throughout the project. But since this model depends a lot on customer interaction hence might go wrong if the the customer is unsure of their expectations.

Iterative Model:

This SLDC model emphasizes on repetition. Instead of starting with fully known requirements, the developers implement a set of software requirements, then create a version of it quickly for a relatively little cost, test, then evaluate and pinpoint further requirements. A new version of the software will be produced with each phase, or iteration. The same cycle will rinse and repeat until the complete system is ready.

This model is able to give the developers a working version early in the process and makes it less expensive to implement changes. However, resources can quickly be eaten up by repeating the process again and again which may cost a lot.

sources:

https://www.roberthalf.com/blog/salaries-and-skills/6-basic-sdlc-methodologies-which-one-is-best

https://stackify.com/what-is-sdlc/

Ming Lun's SDLC + 3 part model

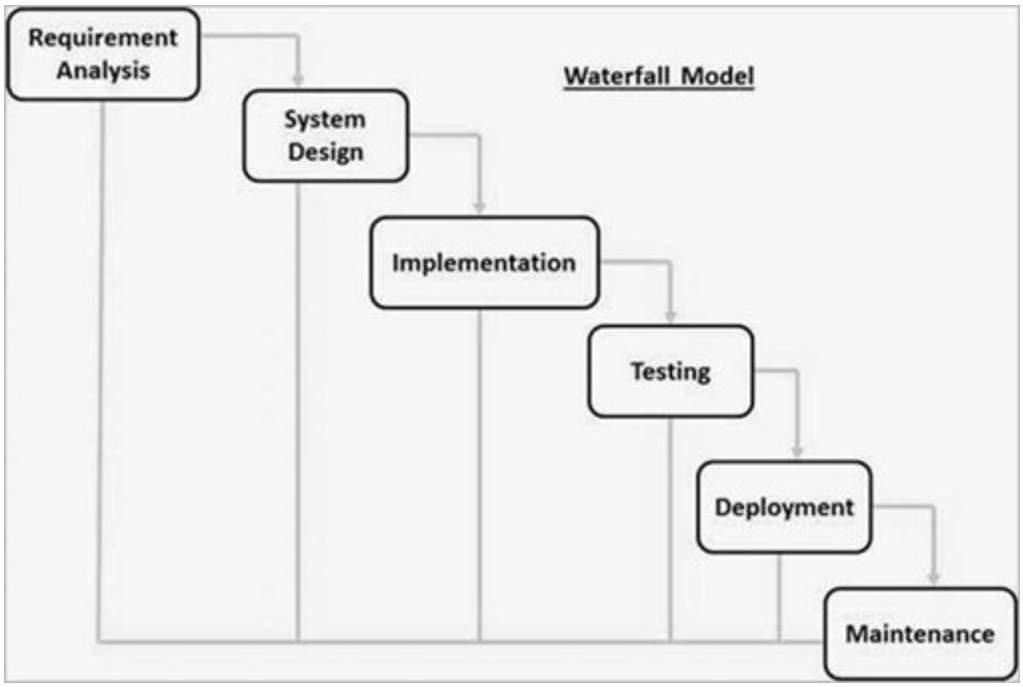
The SDLC (aka Software Development Lifecycle) is the process by which the software is conceptualized, developed and maintained. There are different types of SDLC models out there that serve different purposes.

SDLC Development Models

1) Waterfall Model

The Waterfall Model is one of the earliest SDLC approach that was used in software development. Just like a waterfall, this model depicts the software development process in a **systematic linear flow**, flowing steadily downwards through the phases. This meant that the development process can only continue if the previous phase is complete and do not overlap. This is so as the outcome of one phase acts as the input for the next phase.

There are 7 key phases to understanding the Waterfall SDLC Model:



7 Key Phases
1. Planning
2. Requirement Analysis
3. Design
4. Implementation / Coding
5. Testing
6. Deployment
7. Maintenance

How does this work?

0) Before Everything
External customers with their business ideas and funding will scout and reach out for various tech companies until they find the one that is he/she finds suitable for this project.

After reaching out to the company, the customer is going to meet the company's product owner or AKA the project manager. Which then, they will be **discussing terms of their agreement, sign a deal and accept the project.**

1) Planning Requirements
In this phase, both the customer and the product owner will outline the **requirements of the application product.** This may include defining key features of the application and their functionalities and uses. Multiple iterations of the SDLC will run through for possible new features & requirements that may be added in the future.

Example:
The Blackboard application requires a Subjects page.
The Blackboard application requires a Login page.
The Blackboard application requires a Dashboard Landing page.
… etc.

2) Requirements Analysis
After defining the **outlined requirements**, the team that is involved in the project will meet up for a meeting to discuss and analyse matters regarding the product features. The team may consist of operation managers, developers, product owners and testers etc. They will **further define into specifics of each outlined requirement and give more planning details**. Once everything is approved by the stakeholders, all information regarding the product requirements will be documented in this **SRS (Software Requirement Specification) document** which will serve as a guideline throughout the project life cycle.

Example:

The Blackboard application will require a login page for students to access their forums will require:
1) Username input field
2) Password field …etc
3) Submit Button
4) Read User from Database
5) Log User into the System

3) Design
In this phase, all the agreed requirements will be taken into consideration and planning of different aspects of the product will start. (Etc. what does the product look like?)

These are some of the more common aspects that will be considered:

a) Business Rules
b) User Interface Layout
c) Colour Schemes
d) Programming Languages
e) Frameworks
f) System Server Design
g) Database Relationships
h) Architect of Application
i) Mobile Aspects
j) Supported Browsers
… (many more)

4. Implementation / Coding
This phase is where the action and actual development starts, and the product is built.

The **Operation team** will **start setting up the physical hardware** for the servers.

The **Developers** will **start writing the codes** for the software. (using suitable programming languages that were agreed in earlier stages of the SDLC)

The **Designers** will **continue planning the user interface** of the system.

The **Testers** will **analyse the requirements** and build test cases for their **test plans**. These tests are used to measure the usability of the application so to ensure smooth operation of the project. They find out fundamental flaws in the early stages of the application and help redesign in those areas.

5. Testing
This stage is where there will be a **product testing for defects** which were reported, tracked, fixed and retested until the product satisfies the quality defined in the **SRS**

(Software Requirement Specification). Defects can range from many things **such as colour scheme error, scripting error, user interface error etc. Testers** will gather all the information of the current bug issues and **report them into a bug tracking system. Which then will be transferred to the developers to fix the issue.** This phase is quite important as it may cause implications if the errors are not checked and removed.

Example:
When a user logs out of his/her credit card account, but the session is not being cleared at all. A random perpetrator can walk up to the computer and click the refresh the page, gaining access to their and start making on various things using the account. These security flaw in the programming may cause monetary loss for customers and thus affecting the company's name. Thus, it is important to get everything tested, before releasing the product for the public to use.

6. Deployment
When the product is done and confirmed ready to be released formally in the market after several testing. The operations team will finish up final stages and get them ready for production meaning that they'll make everything scalable for production. Which includes setting up links and database for real users, working alongside developers and other departments. After the product has received confirmation and is ready, the product will then be deployed to the public.

7. Maintenance
This is the stage where the application is getting a little bit more popular and used by many public users. (Logging in, registering & using the application) As such, maintenance is highly required to maintain and monitor many aspects in the application environment, to ensure smooth running of the application.

Examples:
a) Server Issues
Due to the high amounts of users logging in, registering and using the application. The operation team will need to monitor the load of the stress that is being imposed onto the servers, this is important as it is to ensure that the high user population would not make the entire application system crash and become unusable. They will react by making larger server network, larger databases to support the user population.

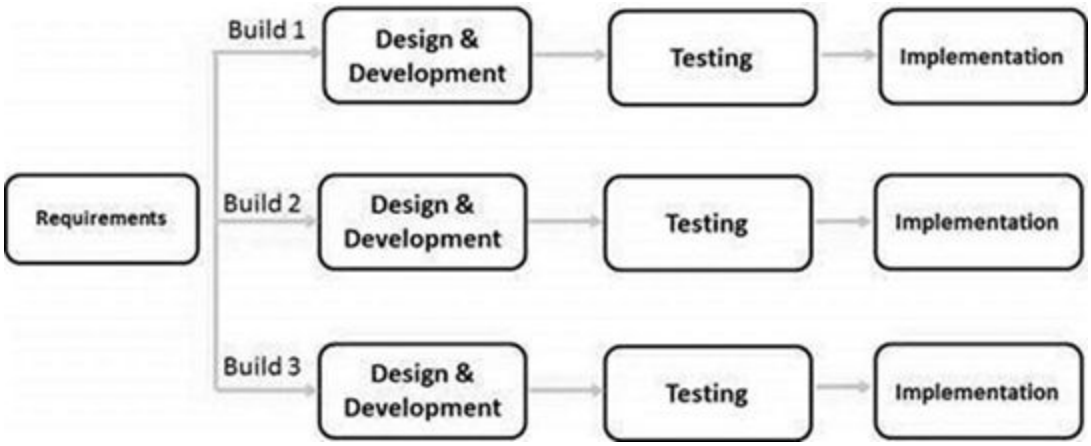
b) Production Support
There will bound to be bugs inside the systems even after release, customers will frequently send in emails regarding bugs or any issues regarding the application. Which then the developers will look into these issues and get them settled and redeployed, eliminating the bugs and issues.

Software Projects that work well under the Waterfall Model (When to use)

- Projects which software requirements are clearly defined, fixed and well documented.
- *Ample resources with the relevant professionals to support the development.*
- *Technology used for the project is not too complex and dynamic* ***(may cause complications, since this model depends on phase succession)***
- *The project is a short one.* ***(when an error surfaces in between the developmental stages, it may cause issues and possibly a restart of the project is needed)***

2) Iterative Model

The iterative process begins by implementing a simple prototype from a small set of identified software requirements, and repeatedly evolving the prototype software through various stages until it is fully complete and ready to be deployed to the market. The general idea is to develop the software through repeated cycles **(iterative)** and in small portions progressively over time **(incremental).**



In this model, the requirements will be divided into various segments, which each segment will be used in each iteration phase, **where the development module will go through the specified requirements, design, implementation and testing phases***. Which results in small improvements from each phase.

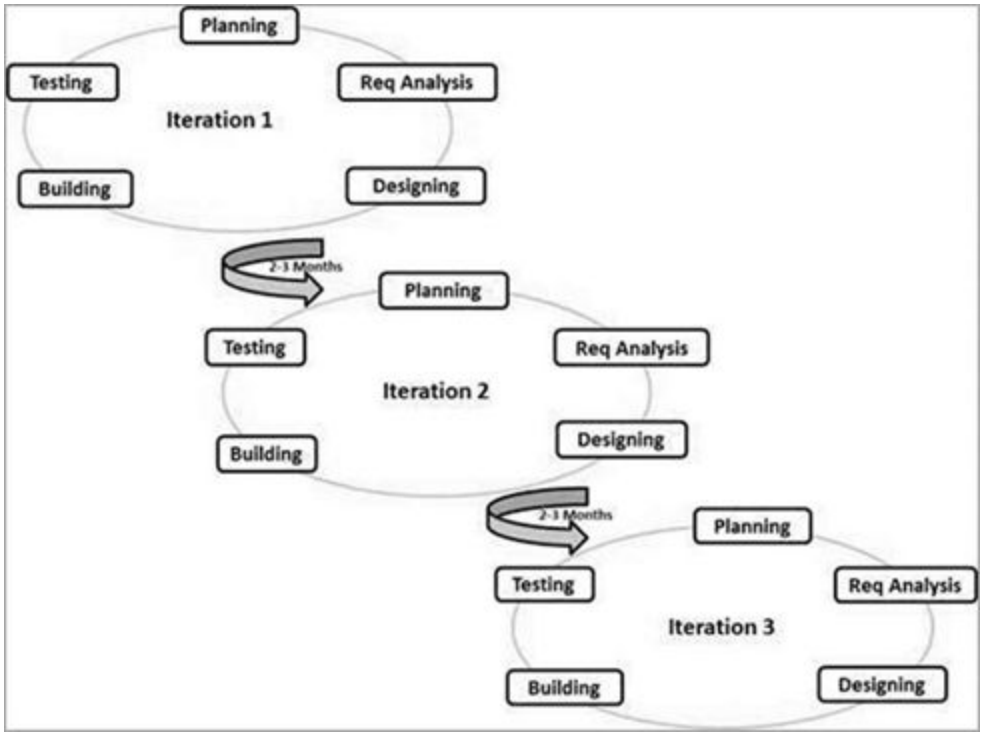
The general development idea of (specifying requirements, designs, implementation and testing phases) has been explained under the Waterfall Model part.

Software Projects that work well under the Iterative Model (When to use)

- Projects which software requirements are clearly defined, fixed and understood.
- Major requirements must be highlighted, however some functions might subject to change/evolve over the developmental phases.
- Introduction to new technology for the project is possible due to its slow pacing dynamic, the development team can learn the new skills while working on the project.

- Due to the slow dynamic of the model, there are certain risks which may affect the end goal in the future**. (lack of time)**
- More suitable for bigger scale projects. **(long term)**

3) Agile Model
Just like the iterative model, the Agile model also adopts the iterative approach when working on the software product. However, what differentiates the Agile model is that it's tasks are divided into separate time phases/periods to deliver specific features for release.



Agile uses an ***adaptive approach*** where detailed planning is not emphasized, only giving the general scope of what the software should have. This encourages the development team to adapt and explore new possibilities and ideas, due to the dynamic change in product requirements. These products will be tested frequently, through the release iterations, minimizing the risk of any major failures in future.

In addition, ***customer interaction*** is a vital factor to the Agile Methodology which has a more open communication development environment as compared to other SDLCs. This ensures that both the developers and customer parties have a common goal in sight.

Chien Yuan's SDLC + 3 part model

The SDLC is a framework describing the tasks performed in each step during the software development process. It is a structure that is followed by the development team within the software organization. The SDLC includes an in depth plan on how to develop, maintain and replace specific software.

Planning: The requirement analysis are done by software engineers, and after the requirements are gathered from the customers, a document is created and the project is determined and documented.

Implementation: The software engineers start writing the code according to the client's requirements.

Testing: This process of finds defects or bugs in the created software.

Documentation: The project is documented for future reference and for the improvement of the software in the development process.

Deployment and maintenance: The software is deployed after it has been approved for release.

Maintaining: Software updates as well as maintenance is done for future reference.

Waterfall Model: After each phase is completed, they are checked to see if there are any errors. If it is successful, then the next phase will be carried out. This process will then repeat until completion of the whole project.

V-Shaped Model: This model is similar to the waterfall model, in terms of phase-by-phase completion. However, it has more importance placed on testing. The testing procedures are written even before the commencement of writing code, and the system plan is generated before starting the development phase.

Incremental Model: This model has multiple development cycles. The cycles are divided up into smaller iterations. These iterations go through a set of phases including requirements, design, implementation and testing. Unlike the other two models, a working version of the software is produced during the first iteration.

V-Shaped Model: V-shaped model is used when the customer has a high confidence of their project, and wants to complete the project without any adjustments.
This is because the model is very rigid, as the software is developed already during the implementation phase, and there are very little to no prototypes produced. If any changes are required, extra work has to be done, such as updating of all the documents. On the other hand, if the customer is very sure of the project, it will be advantageous as tests will be carried out before the coding starts. This reduces waste of time, and increases the chances of success as compared to other models such as the waterfall model.
All in all, it is supposed to be for small to medium sized projects, where the customer is clear of the objectives of the project. Also, having enough resources as well as expertise for the project is crucial.

Joanna's SDLC + 3 part model

Software Development Life Cycle (SDLC)

The Software Development Life Cycle is also known as the application development life cycle. This term is used in systems engineering, information system and software engineering. The process of SDLC involves *planning*, *analysis*, *design*, *implementation*, *testing and integration,* and *maintenance* of an information system. SDLC applies to a range of hardware and software configurations. In which, a system can be composed of hardware only, software only, or a combination of both. SDLC is a process which produces software with the highest quality and lowest cost in the shortest period.

In SDLC, every stage is critical as it affects the result of the software. Planning is the first stage of SDLC. First, the identification of the organisation's objective, the requirements of the new software and determine the cost required are important. By identifying the key points, it will show the risks involved. This will lead to the second stage which is analysis.

Second, after all the findings in the first planning stage, now it's time to evaluate the findings. Analyzing the proposed system and provides an alternate solution for the shortcomings. At the end of this stage, a Software Requirement Specification is created.

Third, in designing, the Software Requirements Specification will be converted into a design plan called the Design Specification. The stakeholders will review the plan and give suggestions. It is compulsory to have a plan for collecting and incorporating stakeholders' contributions into the Design Specification. If any failure occurred at

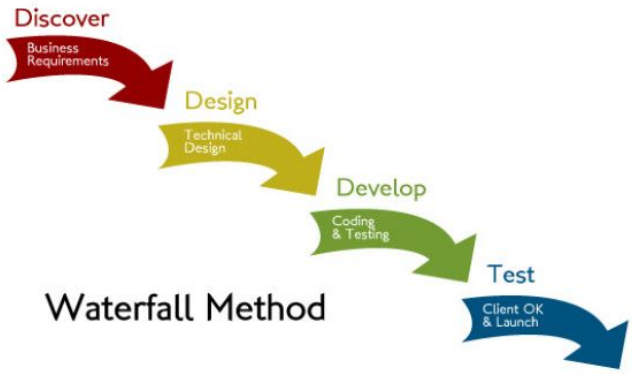
this stage, it will almost certainly result the overrun in cost or total collapse of the entire project.

Fourth, in implementation, software engineers will generate the codes for the system. Fifth, for the testing section, detection for defects and deficiencies is a must. Detecting defects can help in fixing the issues before the product meets the original specifications. Whereas the integration section enables user to try using the product and give necessary feedback for more improvements that can be made.

Finally, the maintenance of the software is vital as it can help users to save cost in the long run. If there is a major breakdown, it can cause huge amount of loss. In addition, new requirements may take a longer to complete.

3 Software Development Models

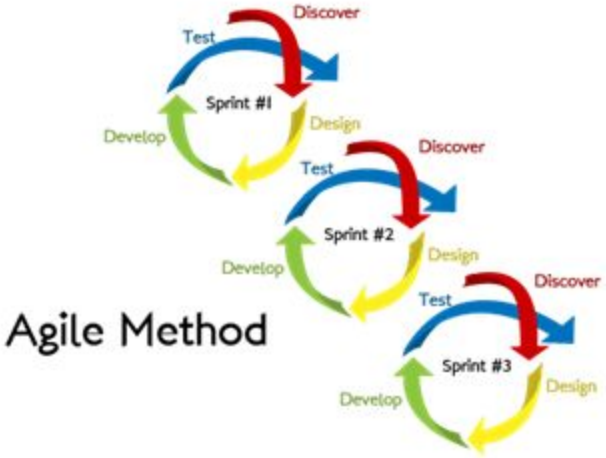
Waterfall Model



This is the oldest and most direct out of all the SDLC models. As can be seen from the diagram above, the process of Waterfall model starts with finding the business requirements. Followed by creating a design, then generating codes and test the system out. This entire process ends when the system succeed in the testing and is ready for launch.

When one phase is finished, it will then move on to the next. There is no going back in this SDLC model. In every stage, waterfall model depends on information from the previous stage and it has its very own project plan. This model is easy to understand as it is simple to manage. However, delays found in the preliminary stages could throw off the entire timeline of the project. There is a limited room for revisions and therefore once a stage is completed, the issues occurred cannot be fixed until it reaches the maintenance stage.

Agile Model



This model separates the product into cycles and delivers a working product very quickly. As can be seen from the diagram above, the process starts with discover, design, then develop and test. It seems like the Waterfall model in most of the processes, however the Agile model produces a succession of releases. This model breaks product into cycle and quickly delivers a working product. It is considered realistic development approach. The testing of each release feeds back information that is incorporated into the next version.

The Agile model emphasizes interaction to all who are involved in the entire process of this model. However, since this model depends on the interaction with customers, the project may head the wrong direction if customers are unsure which approach they want to take.

Iterative Model

This model is a repetition incarnate. It starts off with the implementation of a set of software requirements, followed by the test, evaluation and identify further requirements. In each phase, a latest version of the software is produced. Rinse and repeat until the complete system is ready. This model gives user a working version of the system in the early stage and it ensures making it less expensive to implement changes. However, resources can be eaten up by repeating the process continuously.

Sources:

- https://stackify.com/what-is-sdlc/
- https://www.roberthalf.com/blog/salaries-and-skills/6-basic-sdlc-methodologies-which-one-is-best

1. Introduction

The Delonix Regia hotel hasn't been doing well despite being located at a reasonably good district. This might have something to do with the management system that the hotel is currently adopting. As such the hotel owners both Mr. and Mrs. Wang have requested help from us, a group of students from Temasek Polytechnic to come together and work on the development of a new management system for the hotel.

We have came together and discussed with the hotel owners regarding some of the issues faced by their hotel.

Brief description of Existing System

a) Inefficient Technology
The OPERA property management system is being used for both the front desk and services side. It takes up too much time to perform manual work such as keying in details for the customer or searching up for room information. The loading time frames in between interfaces also takes too long and it cannot handle multiple hotel operations simultaneously. This resulted in dissatisfaction in customers as they would need to wait a longer time to book in or request for any form of services, especially when there is a high guests frequency ordering such services or checking in and out.

b) Cranky & Outdated User Interface
Many employees including the 2 owners of the hotel have complained that the system user interface was very cluttered, hard to read and barely intuitive. Making it really confusing to use, especially for the newer employees who will require more time to get familiarised with the application. In addition, as the hotel is being owned by 2 middle-aged couple, they might have difficulties reading and checking logs in the system due to the messy layout as they may have sight problems due to their age. As such it would help if the user-interface of the system is easy-to-read and convenient for both the owners or employees.

These issues will be a guideline for the development of our software. As requested, we will constantly update the owners of the progression of the project periodically.

Sources:

1. Hotel Management System (Issues)
https://www.quora.com/What-are-some-problem-in-hotel-management-system-software

2. Clutter Effects on User Interface
https://www.infragistics.com/community/blogs/ux/archive/2011/05/16/the-impact-of-screen-clutter-on-the-user-experience.aspx

2. Objectives of the Project

The main objective of the project is **to develop a new hotel management system for the Delonix Regia Hotel, which aims to provide better quality of service and inner operation for the owners and staffs to make use of.**

We have came up with 4 different modules for our new management system and these are the following modules:

1) Room Management Module
2) Employee Management Module
3) Events & Services Management Module
4) F&B Management Module

In order to implement these modules, we need to define our objectives which includes:

1) Room Management Module

- To provide a database to keep the followings:
- Availability of the rooms (includes event halls)
- Customers' details
- To provide tools to help employees manage the booking of the rooms. For example, when will a customer check out the room and when the next customer can check in the same room.

2) Employee Management Module

- To simplify the task of maintaining records of the employees working at the Hotel.
- To develop a simple yet effective database to store these employee information.
- To provide full functional reports to management of the Hotel.
- Provide effective tools to manage the inner operations relating to the Hotel employees. (such as medical leaves, employee scheduling, announcements, job assignment etc.)

3) Events & Services Management Module

- To provide a database to store relevant information regarding upcoming events or services required.
- To provide reports to management of the Hotel.
- To provide tools to help employees manage events and services. For example, event scheduling, so guests can sign up and staff can verify through the system.

4) F&B Management Module

- To provide a database to store the followings:
- Meal courses available. For example, what's on the menu for the Christmas period.
- Amount of stocks for the ingredients. For example, when an ingredient stock is running low, the employee in charge will know to order more stocks for this particular ingredient.
- To provide tools to help employees manage orders. For example, keeping track what time the order is taken and how long it will take to serve the food. This can ensure the quality in food service.

3. Scope of the Project

3.1 Room Management Module

The Room Management feature would allow the consumers to reserve any hotel rooms such as or suites, deluxe room or supreme for their desired duration of stay whether through online reservations or through the receptionist at the lobby of the hotel. Meeting rooms, function rooms or ballrooms can also be reserved as well for any business discussions, corporate meetings, wedding dinners or D&D events host by anyone. It will be convenient as the hotel staffs are able to check the reservations and availability to allow anymore bookings so the consumers and staffs do not need a very long time to process the booking. The consumers are also able to do their bookings online and check if there are any vacancies left for the room that they desire to book while doing the bookings at home. If the system is implemented successfully, it should:

1) Allow the customers to book hotel rooms for their duration of stay.
2) Allow the hotel management to check the availability and reservations made by any consumers for the rooms in the hotel.
3) Allows the customer to book any function rooms for events or hotel rooms for their stay.

3.2 Employee Management Module

The Employee Management module would be beneficial to both the owners and employees working in the hotel. It basically maintains the information about the personal and official details of the employees in the hotel.

Firstly, employees will be able to access their working schedules through the application with their respective ID accounts that they would need to sign up with the managers before joining the hotel. Through this application, there are many features that the employees can make use of to request specific scheduling timings, applying leaves, claim medical money and many other employee welfare features. They will also be given a punch card that they would use every time they come for work,

tapping on the punch card to register that the particular employee has came for work on that day, also recording into the system the duration that the employee has been working in the hotel.

Secondly, the owners and management higher-ups are able to keep track of their employees. This includes managing the schedules for employees, their leave requests, employees details and designations (which part of the hotel does each employee goes), and send pay rolls out to their employees. These are just some of the main features that this module can do.

If the system is implemented successfully, it should:

Employees
1) Allow Employees to request schedule timings.
2) Allow Employees to submit medical leaves.
3) Allow Employees to apply vacation leaves.
4) Allow Employees to check on their current salary & details online.
5) Allow Employees to check their designated schedule and job shift.

Owners
1) Allow Owners & Management Higher-Ups to schedule employee's shifts.
2) Allow Owners & Management Higher-Ups to approve leave requests.
3) Allow Owners & Management Higher-Ups to deliver payrolls/salary online.
4) Allow Owners & Management Higher-Ups to assign job designation.
5) Allow Owners & Management Higher-Ups to manage announcements
6) Allow Owners & Management Higher-Ups to send urgent staffing requests (for areas in the hotel that urgently requires somebody to go and help out.)

3.3 Event & Services Management Module

The Event and Services Management Module would be beneficial to both staff as well as customers. The application provides a platform for both sides to communicate with ease.

Customers may use the Event function of the application to book rooms of the hotel to carry out an event. Through the application, they will be able to enter the time, approximate number of people attending and state whether food is needed. Staff can access the database of the application to access information regarding upcoming events to get relevant information about it. Also, owners can monitor events to keep track of the latest events happening in their hotel.

For example, customers can use the service feature through the application to request for a particular service such as room cleaning or food ordering. The staff will then receive notifications on the application, and will carry the tasks out based on urgency and time of request. After that, they will carry out the services as quick as

possible. After completing services, the staff will then mark the task as complete on the application. The customers will also be able to rate the efficiency of the services carried out, and state any reasons if they are unsatisfied. Owners can see the progress of the tasks carried out and monitor whether the standards of the services are good or in need of adjustments.

Customers

1)Customers can book events when there are rooms that are available.
2)Customers can request for services, and rate the efficiency of the staff.

Owners/Staff

1)Owners and Employees can track upcoming events and find out relevant information.
2)Staff will receive notifications on customers requesting services and will be assigned to carry it out.

3.4 F&B Management Module

The F&B Management Module would benefit the owners, employees and customers. The direct beneficiaries are the owners and employees however, customers will be benefitted indirectly. It was because customers will be able to enjoy higher quality food service. Firstly, it will provide smooth communications. It provides a seamless line of communication between the service and kitchen staff. This lowers the chances of human error or mix-ups in orders.

Secondly, this module can improve the productivity and processes of all the restaurant workstations. It allows user to monitor the progress and workload of the kitchen team to ensure errors such as missing orders are avoided. Third, this module can organize the food service in which improves customers' satisfaction. The module will stay in touch with the orders and menu items front and back of the business in order to serve food in good time to customers. Lastly, this module keeps track of the ingredients. The inventory organization helps to ensure there are sufficient amount of ingredients to run the business. It saves time and wastage with accurate reporting of sales and supplies.

If the system is implemented successfully, it should:

Owners

1) Check the orders
2) Check the performance of the restaurant (in terms of how many or fast the food is served)
3) Check sales

Employees

1) Input/Cancel orders
2) Give instructions (service staff would not need to go to kitchen or vice versa)
3) Check upcoming orders
4) Check when to serve the food
5) Check inventory for the amount of stocks left

4. Distribution of Workload

Objectives/Deliverables	**Members**
Introduction of Project - Purpose of Project - Relevant Background Information - Issues to be Addressed	Cheong Ming Lun & Toh Chien Yuan
Objectives of Project	Discussed by Everyone
Room Management Module - Functionalities / Features	Ong Jia Hui
F&B Management Module - Functionalities / Features	Joanna Lim Min Le
Employee Management Module - Functionalities / Features	Cheong Ming Lun
Event & Service Management Module - Functionalities / Features	Toh Chien Yuan
Constraints	Discussed by Everyone
Resources	Discussed by Everyone
Product Positioning in the Market Company	Joanna Lim Min Le & Ong Jia Hui
Approach & Methodology of Project	Cheong Ming Lun & Toh Chien Yuan

5. Constraints

5.1 Room Management Module

5.1.1 Last Minute Room Cancellation

The Hotel Booking feature in the Room Management module doesn't allow last minute cancellation.

5.2 Employee Management Module

5.2.1 Employee Punch Card

The management application will not record employees present for work if the employee forgets to punch in with his/her employee card.

5.3 Event & Services Management Module

5.3.1 Rating inaccuracy

The ratings that customers leave after a service might not be accurate, as some customers might intentionally give bad or good ratings to particular staff. This will be a minority but it would affect the overall service results of a particular staff.

5.4 F&B Management Module

5.4.1 Orders Input

The module will not show orders to the kitchen staff if the service staff did not managed to input into it.

5.4.2 Stocktake

The module will not be able to show the accurate amount of stocks left if the employee did not input the correct amount in the first place.

6. Resources

6.1 Software Required

6.1.1 Front-End

a) Visual Studio Online

6.1.2 Back-End

Why?

+ Cloud Storage/ Git Sharing
+ Online Collaborative (Shared Work)
+ Project Tracking Tools (Agile Tools)
+ Supports multiple IDEs (Eclipse, ItelliJ, Android Studio, Visual Studio, Visual Studio Code, ...)
+ Free to use

a) MongoDB (Database Server System)

Why?

+ Supports Document Validation
+ Encrypted Storage Engine
+ Great for Content Management (Employee information)
+ Real-time apps with in-memory storage engine

- Doesn't fit applications needing complex transaction

b) Visual Paradigm

Tools to Help Us

+ To plan out our Use Case
+ To capture and emphasize further requirements
+ Process Analysis
+ System Design
+ Database Design

6.2 Hardware Required

6.2.1 Database Hardware

a) Lenovo ThinkServer TS140

- Getting: 4
- $299.95 each from ebay
- CPU: Intel Xeon E3-1226 v3
- RAM: 4GB
- Memory: 2 x 1TB HDD

Total Price: $1,199.80

6.2.2 Punch Card System (for employees)

a) Datamine 620D Punch Card Time Recorder (Punch card machine)

+ Affordable punch card machine
+ Timing Display
+ Backed with employee shift timings in database.
+ Card fairly light and small, easy storage.

Pricing: $52.80 each
Getting: 4 (for different locations)

Total Price: $211.20

7. Product Positioning in the Market/Company

Describe how your product differentiates itself from similar products in the market. Describe any interesting or unique features that your product has.

+ Our product will be interactive and very easy to use with its minimalist design
+ Functions that are fully functional
+ Highly Customizable to suit the user's likings.
+ Real Time System

8. Approach and Methodology of the Project

Describe the development model your team will be adopting for the project. Describe any potential risks or problems your team might have adopting the development model and what you can do to overcome them.

Our software will be based on the **Agile Method**

We chose this model as progression based on this model is very **customer satisfaction driven.** As this model allows us to get to understand more about the requirements that our clients wants and through many iterations we get closer to the desired product itself, as there will be many opportunities to communicate and

discuss with our clients.

In addition, last minute changes are also welcomed to alter any final changes and get the application ready and perfected before deployment.

Temasek Polytechnic
School of Informatics and IT

Diploma in Information Technology (IT)

Project Plan

Project Particulars

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

Project Team's Particulars

Matric Number	**Student Name**
1605028H	Ong Jia Hui
1600712B	Joanna Lim Min Le
1601992I	Cheong Ming Lun
1603305B	Toh Chien Yuan

Revision History

Date	Version	Description	Author
27/10/2017	1.0	Initial Project	All
29/10/2017	1.1	TOR	All
01/11/2017	1.2	Project Plan	All
07/11/2017	1.3	Finalisation	All

Table of Contents

1. Introduction
   1.1. Objectives and scope of the project
   1.2. Assumptions and constraints
   1.3. Definitions and acronyms

2. Roles and responsibilities

3. Estimates and project schedule
   3.1. Work breakdown structure
   3.2. Project schedule
   3.3. Budget summary

4. Risk management plan

Project Plan

1 Introduction

1.1 Objectives and scope of the project

The main objective of the project is **to develop a new hotel management system for the Delonix Regia Hotel, which aims to provide better quality of service and inner operation for the owners and staffs to make use of.**

We have came up with 4 different modules for our new management system and these are the following modules:

1) Room Management Module
2) Employee Management Module
3) Events & Services Management Module
4) F&B Management Module

By the end of this project we aim to create these modules with the following features:

1) Room Management Module

- To provide a database to keep the followings:

-Availability of the rooms (includes event halls)
-Customers' details

- To provide tools to help employees manage the booking of the rooms. For example, when will a customer check out the room and when the next customer can check in the same room.

2) Employee Management Module

- To simplify the task of maintaining records of the employees working at the Hotel.
- To develop a simple yet effective database to store these employee information
- To provide full functional reports to management of the Hotel.
- Provide effective tools to manage the inner operations relating to the Hotel employees. (such as medical leaves, employee scheduling, announcements, job assignment etc.)

3) Events & Services Management Module

- To provide a database to store relevant information regarding upcoming events or services required.
- To provide reports to management of the Hotel.
- To provide tools to help employees manage events and services. For example, event scheduling, so guests can sign up and staff can verify through the system.

4) F&B Management Module

- To provide a database to store the followings:
- Meal courses available. For example, what's on the menu for the Christmas period.
- Amount of stocks for the ingredients. For example, when an ingredient stock is running low, the employee in charge will know to order more stocks for this particular ingredient.
- To provide tools to help employees manage orders. For example, keeping track what time the order is taken and how long it will take to serve the food. This can ensure the quality in food service.

1.2 Assumptions and constraints

1.2.1 Purchasing of Servers & Punch Card Systems

We will purchase our servers and card systems from e-commerce websites and it will then be shipped to us and we can start assembling the server roll ups in the server room in the hotel, same goes with the punch card systems.

We will be expecting shipping to arrive in 2 weeks (12-14 days).

However, if the shipping arrives later than 14 days, it might affect our schedule and we might have to push the project back a little. Hence, the project finish date might be affected.

In order to mitigate such problem, we will plan out a risk management plan to reschedule or work patterns to maintain that everything will be done by the proposed date or earlier.

1.2.2 Software Availability

Both Visual Studio's Online, MongoDB & Visual Paradigm are readily available online. They will be downloaded into our computers on the first week and we will start working on the project itself.

1.2.3 Technical Constraints

The time period given to us to complete the work is 30 days. We hope to finish the front end and back end parts of the application by the end of 2 & ½ weeks (Approx 18 days.). By then the servers and all the hardware should be delivered here already. We just have to set up everything and get everything done within (2 days). Which leaves us only about 10 days to do some testing and perfect everything assuming the hardware will be here in time.

10 days is a little tight, but we will try our best to complete the system as soon as possible.

1.3 Definitions and acronyms

Acronym	Term
DB	Database
UML	Unified Model Language
MD	Module
SVR	Severs
SRS	Software Requirements Specifications

2 Roles and responsibilities

Person	Role
Ong Jia Hui	**Project Leader** - Oversees Project Development & Planning **Requirements Reviewer** - Interact with client to gather the requirements for both front-end & back-end needed for the software project. (for the entire team) **User Interface Designer** - Designs the User Interface & User Experience aspect of the app for the **Room Management Module** **Requirements Specifier** - Work on different aspects of the

	requirements that was provided by clients. (for **Room Management Module**)
Cheong Ming Lun	**Project Reviewer / Planner** - Helps with the planning & tidying up the entire project plan and TOR in General **Requirements Reviewer** - Interact with client to gather the requirements for both front-end & back-end needed for the software project. (for the entire team) **User Interface Designer** - Designs the User Interface & User Experience aspect of the app for the **Employee Management Module** **Requirements Designer** - Designs the flow of the application, the user experience(looks, navigation and it's functionality etc.) part of the application for **Employee Management Module**) **Resources Gatherer (Hardware)** - Orders required software/hardware and get it delivered for the team.
Toh Chien Yuan	**User Interface Designer** - Designs the User Interface & User Experience aspect of the app for the **Events & Services Management Module** **Requirements Designer** - Designs the flow of the application, the user experience(looks, navigation and

	it's functionality etc.) part of the application for **Events & Services Management Module**) **Tester** - Tests for bugs or any errors once a prototype of the application is being made. **Software Architect** - Makes decision for the behind the scenes architectural formating for the application. **Resources Gatherer (Software)** - Orders required software/hardware and get it delivered for the team.
Joanna Lim Min Le	**User Interface Designer** - Designs the User Interface & User Experience aspect of the app for the **F&B Management Module** **Requirements Designer** - Designs the flow of the application, the looks, navigation and it's functionality etc. **Tester** - Tests for bugs or any errors once a prototype of the application is being made. **Software Architect** - Makes decision for the behind the scenes architectural formating for the application.

3 Estimates and project schedule

3.1 Work breakdown structure

[Iteration 1 / Sprint 1]

***Approximately 30 Days to Complete 1st Iteration**

Task Name	Task Classification	Duration (hrs/days)
1. Business Analysis Phase	**Large Task**	**6 Days in Total**
1.1 Understanding Client's Issue	**Sub-Task**	**1 Hour of Discussion**
2. Write SRS	**Large-Task**	**6 Days in Total**
2.1.1 Discuss Software Requirements w Team	**Sub-Task**	**3 Hours of Discussion**
2.1.2 Create Use Cases	**Sub-Task**	**3 Days**
2.1.3 Create UML Sequence Diagram	**Sub-Task**	**3 Days**
2.1.4 Create DB Schema	**Sub-Task**	**3 Days**
2.1.5 Discuss Design Theme	**Sub-Task**	**3 Days**
2.1.6 Draw Application Layout	**Sub-Task**	**3 Days**
2.1.7 Rough Sketch of Each Screen	**Sub-Task**	**3 Days**
2.1.8 Finalise Software Planning	**Sub-Task**	**1 Day**
2.2.1 SRS Reviewed by client	**Sub-Task**	**1 Day**

2.2.2 SRS Approved	**Sub-Task**	**1 Day**
3. Development Phase 1	**Large Task**	**11 Days in Total**
3.1 System Foundations	**Sub-Task**	**5 Days**
3.2 Room Management Module	**Large Task**	**5 Days in Total**
3.2.1 Work on Back-End	**Sub-Task**	**4 Days**
3.2.2 Set-up Databases & Link	**Sub-Task**	**1 Day**
3.2.3 Work on Front-End	**Sub-Task**	**4 Days**
3.2.4 Module Testing	**Sub-Task**	**1 Day**
3.3 Employee Management Module	**Large Task**	**5 Days in Total**
3.2.1 Work on Back-End	**Sub-Task**	**4 Days**
3.2.2 Set-up Databases & Link	**Sub-Task**	**1 Day**
3.2.3 Work on Front-End	**Sub-Task**	**4 Days**
3.2.4 Module Testing	**Sub-Task**	**1 Day**
3.4 Events & Services Management Module	**Large Task**	**5 Days in Total**
3.2.1 Work on Back-End	**Sub-Task**	**4 Days**
3.2.2 Set-up Databases & Link	**Sub-Task**	**1 Day**
3.2.3 Work on Front-End	**Sub-Task**	**4 Days**
3.2.4 Module Testing	**Sub-Task**	**1 Day**
3.5 F&B Management	**Large Task**	**5 Days in Total**

Module		
3.2.1 Work on Back-End	**Sub-Task**	**4 Days**
3.2.2 Set-up Databases & Link	**Sub-Task**	**1 Day**
3.2.3 Work on Front-End	**Sub-Task**	**4 Days**
3.2.4 Module Testing	**Sub-Task**	**1 Day**
3.6 Connect All Modules	**Large-Task**	**2 Days**
3.7 Customer Review 1	**-**	**5 Days**

***Will Undergo Iteration 2 if first Iteration was not satisfied.**

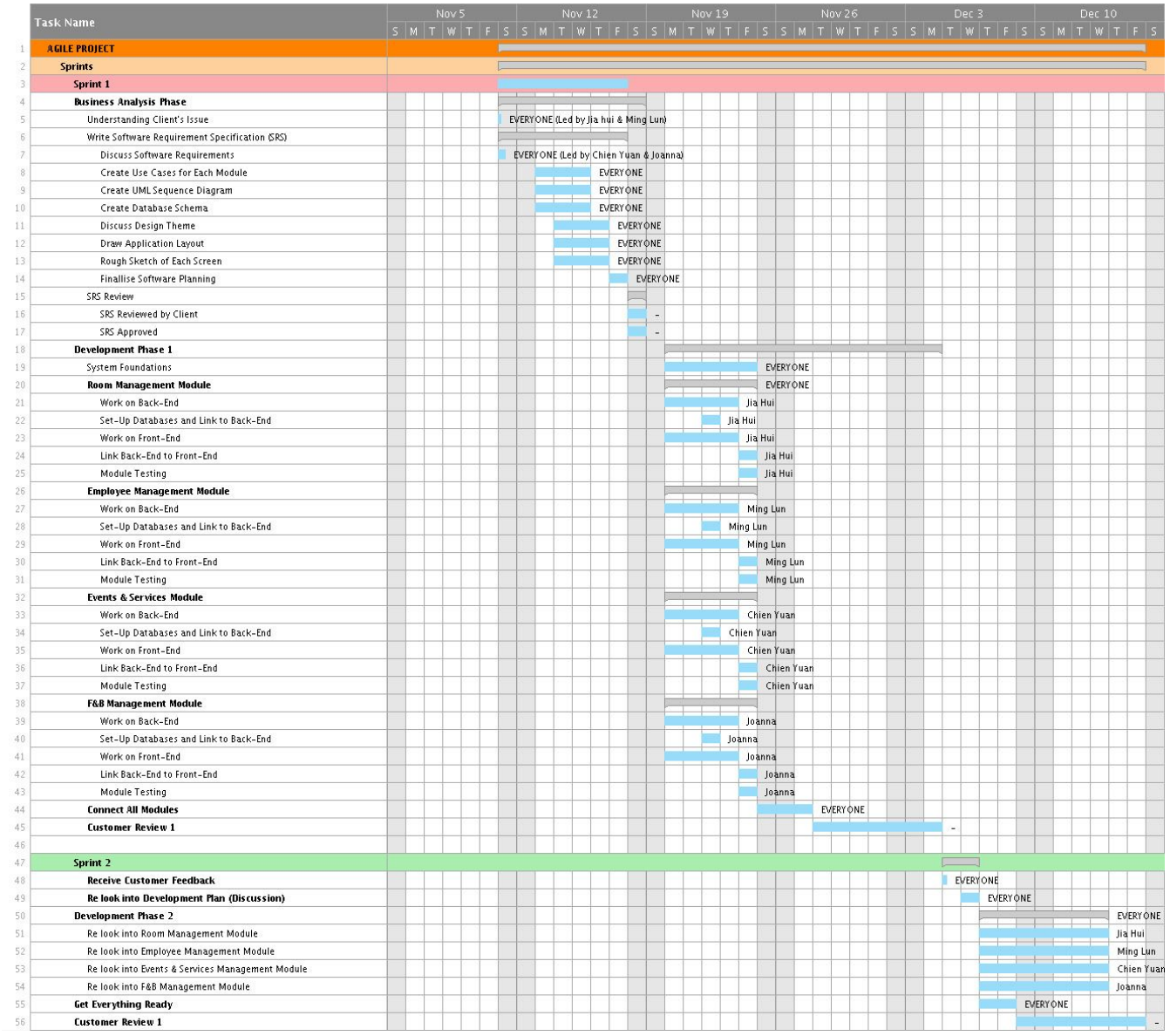
[Iteration 2 / Sprint 2]
***Approximately 8 Days to Complete 2nd Iteration**

Task Name	Task Classification	Duration (hrs/days)
1. Receive Customer Feedback	**-**	**2hrs**
2. Re Look into Development Plan	**Large Task**	**1 day**
3. Development Phase 2	**Large Task**	**5 days in total**
3.1 Re look into Room Management Module	**Sub-Task**	**5 days**
3.1.1 Re look into Employee Management Module	**Sub-Task**	**5 days**
3.1.2Re look into Events & Services Management Module	**Sub-Task**	**5 days**

3.1.3 Re look into F&B Management Module	**Sub-Task**	**5 days**
4. Get Everything Ready	**Large Task**	**2 days**
5. Customer Review 2	**-**	**6 days**

***If customer is satisfied, the project is a success.**
***If customer still wants some changes, it will go through another round of iteration just like iteration 2.**

3.2 Project Schedule



3.3 Budget Summary

Category	Cost
1) Manpower	**Basic Rate of a Software Engineer in Singapore is:** $4k **Duration of Project:** Approx. 1 month ++ **No. Of people Hired:** 4 people (4 of us students) **Total Amount:** $12k- $13k
2) Hardware	a) **Lenovo ThinkServer TS140** - **Getting: 4** - **$299.95 each from ebay** - **CPU: Intel Xeon E3-1226 v3** - **RAM: 4GB** - **Memory: 2 x 1TB HDD** ***Total Price: $1,199.80*** **b) Datamine 620D Punch Card Time Recorder (Punch card machine)** **Pricing: $52.80 each** **Getting: 4 (for different locations)** **Total Price: $211.20**
3) Software	- **NIL**
4) Total	**$13k + $1,199.80 + $211.20 = $14,411.10**

4 Risk Management Plan

We will try to adhere to the time schedule that we have set for ourselves. However, let's say if we really do run out some time, we will reallocate our jobs to fit a more dynamic and efficient way to conserve time to meet to the needs of the deadline.

Sources:

1) Database Software
https://blog.capterra.com/free-database-software/

2) Employee Punch Cards
http://www.datamine.com.sg/dm620D.php

3) Hardware Requirements
http://help.sana-commerce.com/sana-commerce-83/installation/setup-web-and-database-server/hardware-requirements-for-web-and-database-servers

4) Hotel & Kitchen Management System
https://www.tigernix.com/home/software/hotel-management-system/table-kitchen

5) Visual Studios Online
https://www.youtube.com/watch?v=Xqj6pwAuErs
https://www.visualstudio.com/vso/

6) MongoDB
https://www.google.com.sg/search?q=Mongodb&ie=utf-8&oe=utf-8&client=firefox-b-ab&gfe_rd=cr&dcr=0&ei=UT8BWszvJomh8wfTuaOIDg