***PROJECT PLAN***

***AND***

***INITIAL SYSTEM ANALYSIS***

***FOR***

***DINE TOGRTHER SYSTEM***

Version: *1.0*

***AUG 2016***

Tang Sze Man (15207323)

TABLE OF CONTENTS

1 PROJECT PLAN 1

1.1 BRIFE INTRODUCTION TO PROPOSED SYSTEM 1

1.2 RESOURCE IMPLICATIONS 1

1.2.1 HARDWARE 1

1.2.2 SOFTWARE 1

1.3 DEVELOPMENT AND OPERATING COSTS 1

1.4 TANGIBLE AND INTANGIBLE BENEFITS 2

1.5 COST-BENEFITS ANALYSIS 3

1.6 DEVELOPMENT SCHEDULE 4

1.7 RECOMMENDATION 7

2 INITIAL SYSTEM ANALYSIS 8

2.1 INITIAL USE CASE MODEL 8

2.2 INITIAL USER INTERFACE PROTOTYPING 13

2.2.1 Member Registration 13

2.2.2 Login 14

2.2.3 Modify Member Information 15

2.2.4 Modify Friend List 16

2.2.5 Chat Room 17

# PROJECT PLAN

## BRIFE INTRODUCTION TO PROPOSED SYSTEM

Eating with a group of people is happier. Or somebody may say, always eating with the same group of people is boring. So, we may find someone to dine with us.

Sometimes, trying some set menus, like Chinese set menu for 16 people or Valentine Day’s menus, which requires specific number of people. Therefore, it would be perfect if I could find some people who also want to try the same set.

And people may want to find friend share the same taste and interests. For instance, cat lovers may like to go to cat cafes with cat lovers met on the Internet to try to go to maid cafe with others.

Also, someone may want to find a listener. Like talking secrets to strangers.

Maybe it is useful if we can search for some restaurants by restaurants’ general information, keywords in the menus, comments, photos, and videos.

Now, “Dine Together System” can help us find “dinning partners” and solve the situations above.

The “Dine Together System” has three types of users: Member, Restaurant Owner, and Administrator. And there are three subsystems: Membership Management Subsystem, Dining Event Subsystem, and Restaurant Comment Subsystem.

## RESOURCE IMPLICATIONS

### HARDWARE

* Virtual Machine
  + 1 core CPU (Virtual)
  + 4 GB RAM (Virtual)
  + 100G Disk Space (Virtual)
  + 1 Network Card (Virtual)

### SOFTWARE

* Operating System: Ubuntu Server 16.04.1 LTS
* Web Server: Apache 2.4
* Database: MySQL Database 5.7
* IDE: PHP Storm, Sublime text, Atom
* Processing software: Microsoft Office

## DEVELOPMENT AND OPERATING COSTS

Hosting the system is nearly the only cost for developing and operating the system.

For development costs, since hardware are using existing item and most of them are on cloud, the cost is assume to be zero. For software, the plan is going to use freeware or open source software, so, the cost is assume to be zero.

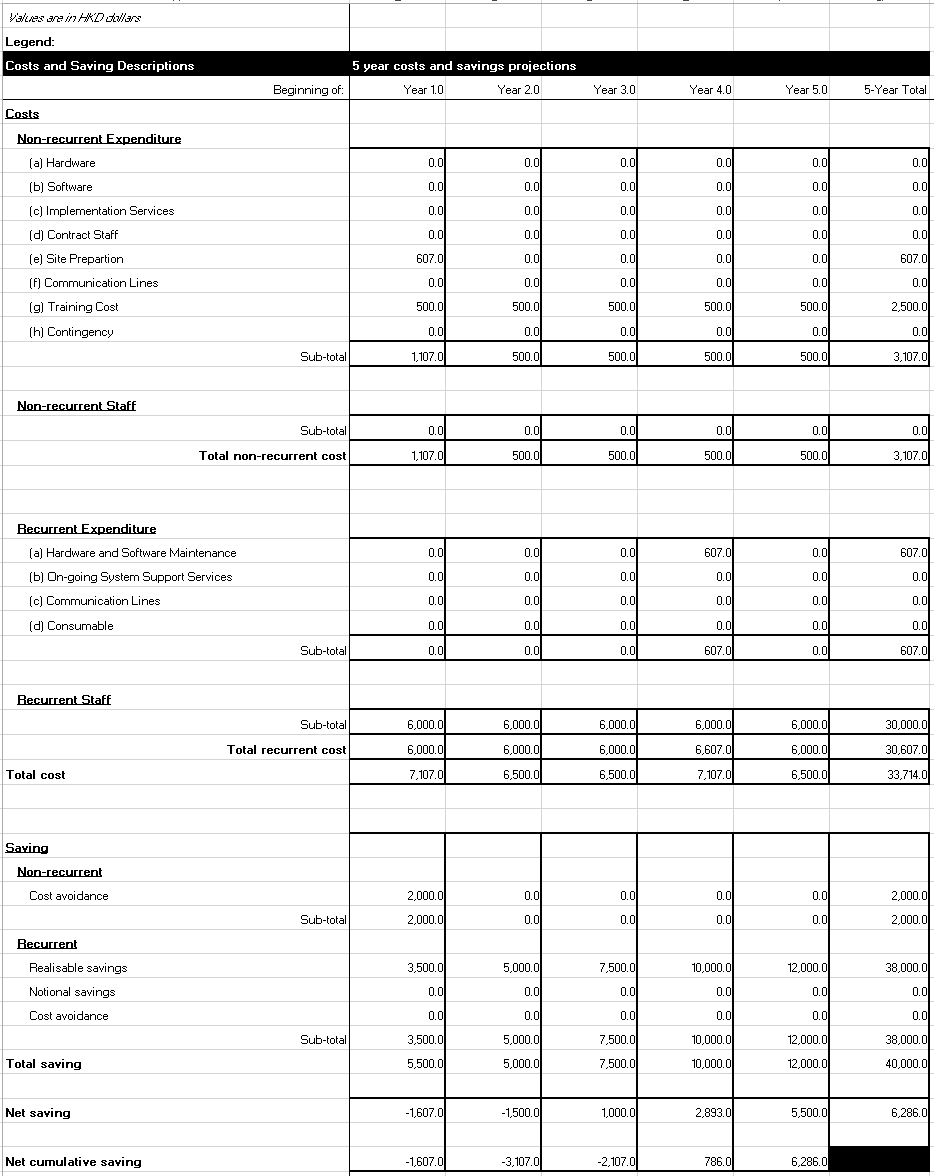
For operating cost, administrator, programmer and editor have to be hired for maintaining and supporting the system. This incurs training cost and wage expenditure.

## TANGIBLE AND INTANGIBLE BENEFITS

For tangible benefits, since most “hardware” are virtual, servers are not needed to be bought. The costs for hardware are saved. Moreover, by using open source software, the costs for software are saved.

For intangible benefits, improve employees’ working environment since home office is recommended.

## COST-BENEFITS ANALYSIS



The break-even point will be in the 4th year. Expense of site preparation is the cost of hosting the server on Amazon Web Services. The recurring staff is hired for doing administration work and maintaining the system. The training cost is for the staff.

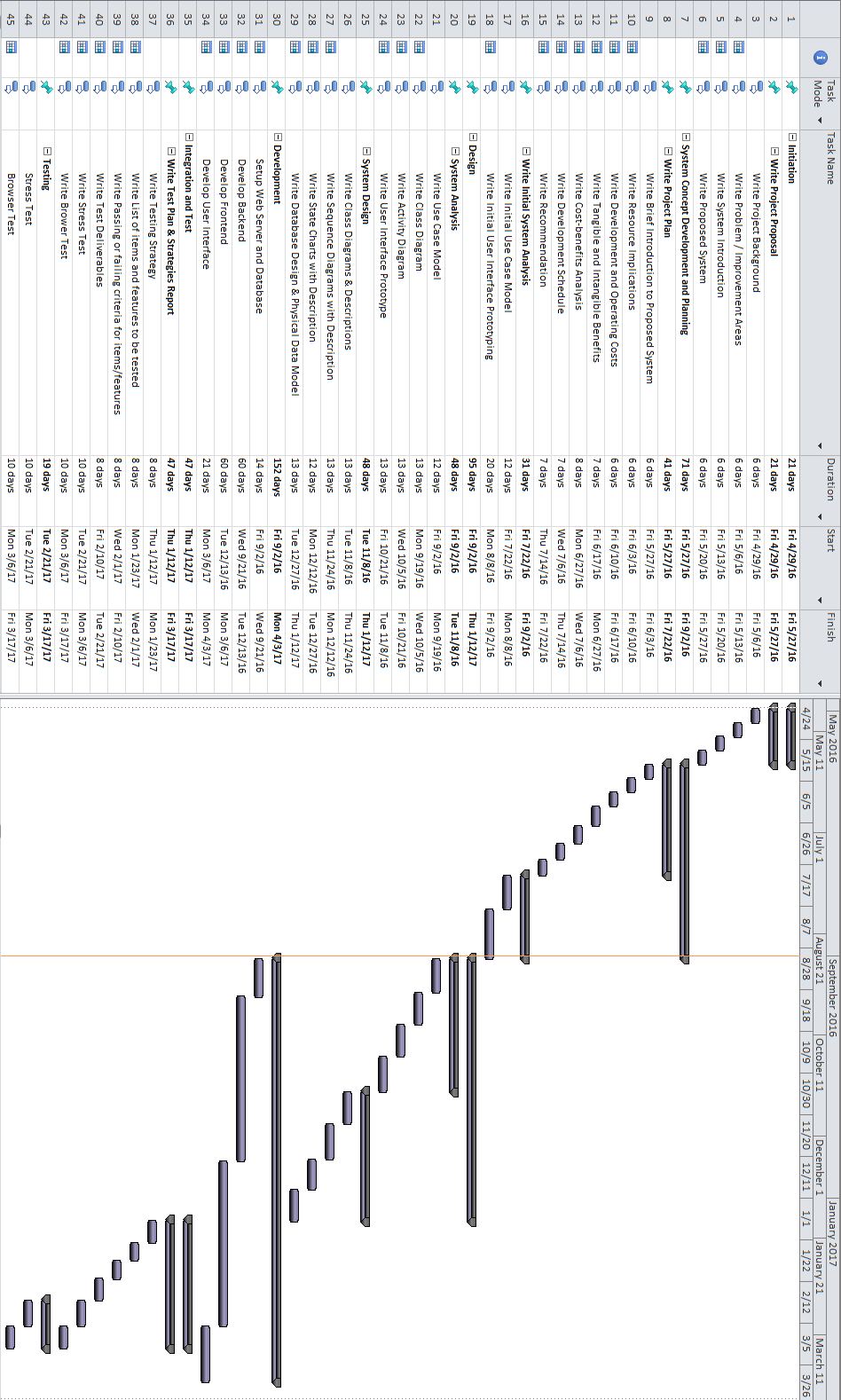
Return on Investment  
ROI = ($40,000.0 - $33,714.0) / $33,714.0 = 0.19

According to the return on investment, the return is slow but finally can gain.

## DEVELOPMENT SCHEDULE

Schedule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Duration | Start | Finish |
| 1 | Initiation | 21 days | 4/29/16 8:00 AM | 5/27/16 5:00 PM |
| 2 | Write Project Proposal | 21 days | 4/29/16 8:00 AM | 5/27/16 5:00 PM |
| 3 | Write Project Background | 6 days | 4/29/16 8:00 AM | 5/6/16 5:00 PM |
| 4 | Write Problem / Improvement Areas | 6 days | 5/6/16 8:00 AM | 5/13/16 5:00 PM |
| 5 | Write System Introduction | 6 days | 5/13/16 8:00 AM | 5/20/16 5:00 PM |
| 6 | Write Proposed System | 6 days | 5/20/16 8:00 AM | 5/27/16 5:00 PM |
| 7 | System Concept Development and Planning | 71 days | 5/27/16 8:00 AM | 9/2/16 5:00 PM |
| 8 | Write Project Plan | 41 days | 5/27/16 8:00 AM | 7/22/16 5:00 PM |
| 9 | Write Brief Introduction to Proposed System | 6 days | 5/27/16 8:00 AM | 6/3/16 5:00 PM |
| 10 | Write Resource Implications | 6 days | 6/3/16 8:00 AM | 6/10/16 5:00 PM |
| 11 | Write Development and Operating Costs | 6 days | 6/10/16 8:00 AM | 6/17/16 5:00 PM |
| 12 | Write Tangible and Intangible Benefits | 7 days | 6/17/16 8:00 AM | 6/27/16 5:00 PM |
| 13 | Write Cost-benefits Analysis | 8 days | 6/27/16 8:00 AM | 7/6/16 5:00 PM |
| 14 | Write Development Schedule | 7 days | 7/6/16 8:00 AM | 7/14/16 5:00 PM |
| 15 | Write Recommendation | 7 days | 7/14/16 8:00 AM | 7/22/16 5:00 PM |
| 16 | Write Initial System Analysis | 31 days | 7/22/16 8:00 AM | 9/2/16 5:00 PM |
| 17 | Write Initial Use Case Model | 12 days | 7/22/16 8:00 AM | 8/8/16 5:00 PM |
| 18 | Write Initial User Interface Prototyping | 20 days | 8/8/16 8:00 AM | 9/2/16 5:00 PM |
| 19 | Design | 95 days | 9/2/16 8:00 AM | 1/12/17 5:00 PM |
| 20 | System Analysis | 48 days | 9/2/16 8:00 AM | 11/8/16 5:00 PM |
| 21 | Write Use Case Model | 12 days | 9/2/16 8:00 AM | 9/19/16 5:00 PM |
| 22 | Write Class Diagram | 13 days | 9/19/16 8:00 AM | 10/5/16 5:00 PM |
| 23 | Write Activity Diagram | 13 days | 10/5/16 8:00 AM | 10/21/16 5:00 PM |
| 24 | Write User Interface Prototype | 13 days | 10/21/16 8:00 AM | 11/8/16 5:00 PM |
| 25 | System Design | 48 days | 11/8/16 8:00 AM | 1/12/17 5:00 PM |
| 26 | Write Class Diagrams & Descriptions | 13 days | 11/8/16 8:00 AM | 11/24/16 5:00 PM |
| 27 | Write Sequence Diagrams with Description | 13 days | 11/24/16 8:00 AM | 12/12/16 5:00 PM |
| 28 | Write State Charts with Description | 12 days | 12/12/16 8:00 AM | 12/27/16 5:00 PM |
| 29 | Write Database Design & Physical Data Model | 13 days | 12/27/16 8:00 AM | 1/12/17 5:00 PM |
| 30 | Development | 152 days | 9/2/16 8:00 AM | 4/3/17 5:00 PM |
| 31 | Setup Web Server and Database | 14 days | 9/2/16 8:00 AM | 9/21/16 5:00 PM |
| 32 | Develop Backend | 60 days | 9/21/16 8:00 AM | 12/13/16 5:00 PM |
| 33 | Develop Frontend | 60 days | 12/13/16 8:00 AM | 3/6/17 5:00 PM |
| 34 | Develop User Interface | 21 days | 3/6/17 8:00 AM | 4/3/17 5:00 PM |
| 35 | Integration and Test | 47 days | 1/12/17 8:00 AM | 3/17/17 5:00 PM |
| 36 | Write Test Plan & Strategies Report | 47 days | 1/12/17 8:00 AM | 3/17/17 5:00 PM |
| 37 | Write Testing Strategy | 8 days | 1/12/17 8:00 AM | 1/23/17 5:00 PM |
| 38 | Write List of items and features to be tested | 8 days | 1/23/17 8:00 AM | 2/1/17 5:00 PM |
| 39 | Write Passing or failing criteria for items/features | 8 days | 2/1/17 8:00 AM | 2/10/17 5:00 PM |
| 40 | Write Test Deliverables | 8 days | 2/10/17 8:00 AM | 2/21/17 5:00 PM |
| 41 | Write Stress Test | 10 days | 2/21/17 8:00 AM | 3/6/17 5:00 PM |
| 42 | Write Brower Test | 10 days | 3/6/17 8:00 AM | 3/17/17 5:00 PM |
| 43 | Testing | 19 days | 2/21/17 8:00 AM | 3/17/17 5:00 PM |
| 44 | Stress Test | 10 days | 2/21/17 8:00 AM | 3/6/17 5:00 PM |
| 45 | Browser Test | 10 days | 3/6/17 8:00 AM | 3/17/17 5:00 PM |
| 46 | Implementation | 18 days | 3/17/17 8:00 AM | 4/11/17 5:00 PM |

Gantt Chart  


## RECOMMENDATION

Since the benefits (in terms of money) is not high, to ensure having a positive net cash flow, Google AdSense should be deployed. In addition, promotion med for restaurant should be charged.

Another problem is that, this system is similar to “Openrice”, therefore, the selling point of this system should not be so similar. For example, change the selling point to making friend or speed dating.

# INITIAL SYSTEM ANALYSIS

## INITIAL USE CASE MODEL

The initial system mainly has four types of users: Visitor, Member, Restaurant Owner, and Administrator. Only visitor is not required to login. And there are three subsystems: Membership Management Subsystem, Dining Event Subsystem, and Restaurant Comment Subsystem.



Figure 2.1 Use Case Diagram of Membership Management Subsystem

Membership Management Subsystem allows visitor can register as member; member can login, modify interest, check member point, use member point, send friend request, accept friend request, delete friend request, personal message delivery; at the same time, administrators can warn, ban, and delete account.



Figure 2.2 Use Case Diagram of Dining Event Subsystem

Dining Event Subsystem create dining event, invite specific member to join the dining event, modify created dining event, delete created dining event, join dining event and search dining event, give comment to dining event creator, and give comment to dining event participants, and dining event alert, set requirement (e.g. location, time, minimum and maximum number, members’ interest, and private or public, etc.); at the same time, administrators can warn, ban, delete member accounts, and delete event.



Figure 2.3 Use Case Diagram of Restaurant Comment Subsystem

Restaurant Comment Subsystem allows all people search restaurant, member give comments, upload photos and videos; restaurant owners register restaurant information, management restaurant information, make discount, and advertise; administrators can warn, ban, delete member accounts, delete restaurant, delete comment, delete photos and video.

## INITIAL USER INTERFACE PROTOTYPING

### Member Registration

Function definition:

|  |  |
| --- | --- |
| **Item** | **Description** |
| Function Name | Member Registration |
| Category | Presentation Layer |
| Function Description | Visitor register as member |
| Mode | Online |
| Frequency | Everyday |
| Metadata | Username, gender, E-mail, password, interests, agreement |
| User Input Screens and Forms |  |
| Security Requirements | All password must not be displayed on screen and fulfil the password rule set |

### Login

Function definition:

|  |  |
| --- | --- |
| **Item** | **Description** |
| Function Name | Login |
| Category | Presentation Layer |
| Function Description | Member login to the system |
| Mode | Online |
| Frequency | Everyday |
| Metadata | E-mail, password |
| User Input Screens and Forms |  |
| Security Requirements | All user password must not be displayed on screen during input |

### Modify Member Information

Function definition:

|  |  |
| --- | --- |
| **Item** | **Description** |
| Function Name | Modify Member Information |
| Category | Presentation Layer |
| Function Description | Change personal information |
| Mode | Online |
| Frequency | Everyday |
| Metadata | E-mail, password |
| User Input Screens and Forms |  |
| Security Requirements | All user password must not be displayed on screen during input |

### Modify Friend List

Function definition:

|  |  |
| --- | --- |
| **Item** | **Description** |
| Function Name | Modify Friend List |
| Category | Presentation Layer |
| Function Description | Add and delete friend list |
| Mode | Online |
| Frequency | Everyday |
| Metadata | Add, delete |
| User Input Screens and Forms |  |
| Security Requirements | - |

### Chat Room

Function definition:

|  |  |
| --- | --- |
| **Item** | **Description** |
| Function Name | Chat Room |
| Category | Presentation Layer |
| Function Description | Instant messaging for member |
| Mode | Online |
| Frequency | Everyday |
| Metadata | Message |
| User Input Screens and Forms |  |
| Security Requirements | Message are encrypt |