Message Capsule –

Conditional-based Messaging System

by

NG Wai Tung, Terence

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science (Honours)

in Computing Studies

Hong Kong Baptist University

September, 2012

# Declaration

I hereby declare that all the work done in this Final Year Project is of my independent effort. I also certify that I have never submitted the idea and product of this Final Year Project for academic or employment credits.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

NG Wai Tung, Terence

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Message in a Paper Plane –

Conditional-based Messaging System

by

NG Wai Tung, Terence

Second Progress Report

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science (Honours)

in Computing Studies

Hong Kong Baptist University

January, 2013

內容

[Declaration 2](#_Toc349918111)

[1 Introduction 4](#_Toc349918112)

[1.1 Deliverables in this report 4](#_Toc349918113)

[1.2 The changes since first progress report 4](#_Toc349918114)

[1.2.1 Changes of System Objectives 4](#_Toc349918115)

[1.2.2 Changes of System Audience 4](#_Toc349918116)

[1.2.3 Changes of proposed hardware and software configuration 4](#_Toc349918117)

[1.3 New Deliverables in Second progress report 4](#_Toc349918118)

[1.3.1 Scenarios 4](#_Toc349918119)

[1.3.2 Prototyping 4](#_Toc349918120)

[1.3.3 Technical Diagrams 4](#_Toc349918121)

[1.3.4 Interface Design 4](#_Toc349918122)

[2 Revised Project Proposal 5](#_Toc349918123)

[2.1 Project Objectives 5](#_Toc349918124)

[2.1.1 Problem Identification 5](#_Toc349918125)

[2.1.1.1 Sending message is Instantaneous 5](#_Toc349918126)

[2.1.1.2 Route of message transferring is not spread according to geolocation 5](#_Toc349918127)

[2.1.1.3 User wants to control how receiver receive the message 5](#_Toc349918128)

[2.2 Proposed Solution 5](#_Toc349918129)

[2.2.1 Send different type of messages by user selection 5](#_Toc349918130)

[2.2.2 Receive different type of messages by user selection 6](#_Toc349918131)

[2.2.3 Business User specified functions 6](#_Toc349918132)

[2.2.4 Administration monitoring & logging functions 6](#_Toc349918133)

[2.3 Scope Definition 6](#_Toc349918134)

[2.3.1 Project Name 6](#_Toc349918135)

[2.3.2 Project Overview 6](#_Toc349918136)

[2.3.2.1 Provide a messaging system that is usable in mobile and web. 6](#_Toc349918137)

[2.3.2.2 Support new scenarios by enabling scheduling messaging based on mix of conditions 6](#_Toc349918138)

[2.3.3 Project Scope 6](#_Toc349918139)

[2.4 Project Plan 7](#_Toc349918140)

[2.4.1 Feasibility Study 7](#_Toc349918141)

[2.4.1.1 Operational Feasibility 7](#_Toc349918142)

[It is whether the user can use the supporting system to solve the problems in the supported system. In this case, it is whether the messaging system can solve the communication problem of user in the real world. 7](#_Toc349918143)

[2.4.1.2 Cultural Feasibility 7](#_Toc349918144)

[2.4.1.3 Technical Feasibility 7](#_Toc349918145)

[2.4.1.4 Economic Feasibility 7](#_Toc349918146)

[2.4.1.5 Schedule Feasibility 7](#_Toc349918147)

[2.4.1.6 Legal Feasibility 7](#_Toc349918148)

[2.4.2 Candidate Systems Matrix 7](#_Toc349918149)

[2.4.3 Feasibility Analysis Matrix 8](#_Toc349918150)

[2.4.4 Project Development Schedule 10](#_Toc349918151)

[2.4.5 Cost Estimation 11](#_Toc349918152)

[3 Revised First Progress Report and Second Progress Report 12](#_Toc349918153)

[3.1 System Analysis 12](#_Toc349918154)

[3.1.1 Source of Information 12](#_Toc349918155)

[3.1.1.1 Observation from App store 12](#_Toc349918156)

[3.1.1.2 Psychology Research 12](#_Toc349918157)

[3.1.2 Modeling Method Comparisons 12](#_Toc349918158)

[3.1.2.1 Structured analysis and design method 12](#_Toc349918159)

[3.1.2.2 Object-oriented analysis and design method 12](#_Toc349918160)

[3.1.3 Chosen Modeling Method 12](#_Toc349918161)

[3.1.4 Data Modeling 13](#_Toc349918162)

[3.1.4.1 Class Diagram 13](#_Toc349918163)

[3.1.4.2 Entity Relationship Diagram 13](#_Toc349918164)

[3.1.4.3 Entity Relationship Diagram Dictionary 14](#_Toc349918165)

[3.1.4.4 Entity Relationship Diagram (Normalized) 15](#_Toc349918166)

[3.1.4.5 Entity Relationship Diagram Dictionary (Normalized) 15](#_Toc349918167)

[3.1.5 Process Modeling 15](#_Toc349918168)

[3.1.5.1 Event Table 15](#_Toc349918169)

[3.1.5.2 Event Table Compact List 16](#_Toc349918170)

[3.1.5.2.1 User registers account 16](#_Toc349918171)

[3.1.5.2.2 User invites friends 16](#_Toc349918172)

[3.1.5.2.3 User sends message 16](#_Toc349918173)

[3.1.5.2.4 User captures message 16](#_Toc349918174)

[3.1.5.2.5 User shares message 16](#_Toc349918175)

[3.1.5.2.6 User checks statistics 16](#_Toc349918176)

[3.1.5.2.7 System changes message directions 16](#_Toc349918177)

[3.1.5.2.8 System notifies message being captured 16](#_Toc349918178)

[3.1.5.2.9 Admin handles report of spam message 16](#_Toc349918179)

[3.1.5.2.10 Business People sends message 16](#_Toc349918180)

[3.1.5.2.11 Business People pays for advertisement message 16](#_Toc349918181)

[3.1.5.3 Use case diagram 16](#_Toc349918182)

[3.1.5.4 Activity Diagram 17](#_Toc349918183)

[3.1.5.5 Sequence Diagram 20](#_Toc349918184)

[3.1.5.6 Functional Decomposition Diagram 22](#_Toc349918185)

[3.1.5.7 Context Data Flow Diagram 23](#_Toc349918186)

[3.2 System Design 23](#_Toc349918187)

[3.2.1 Finalized hardware and software configurations 23](#_Toc349918188)

[3.2.1.1 Detail of software technology use 24](#_Toc349918189)

[3.2.2 Scenarios 25](#_Toc349918190)

[3.2.3 Input Design 25](#_Toc349918191)

[3.2.3.1 User log into the Message Capsule System 25](#_Toc349918192)

[3.2.3.2 User registering account 26](#_Toc349918193)

[3.2.3.3 User sending message in web 27](#_Toc349918194)

[3.2.3.4 User sending message in mobile 27](#_Toc349918195)

[3.2.3.5 User changing settings 32](#_Toc349918196)

[3.2.4 Output Design 33](#_Toc349918197)

[3.2.4.1 User receiving message in web 33](#_Toc349918198)

[3.2.4.2 User receiving message in mobile 33](#_Toc349918199)

[3.2.4.3 User checking own usage statistics 33](#_Toc349918200)

[3.2.4.4 Administrator monitoring overall user usage 33](#_Toc349918201)

[3.2.5 Interface Design 33](#_Toc349918202)

[3.2.5.1 Logo Design 33](#_Toc349918203)

[3.2.5.2 Color scheme 33](#_Toc349918204)

[3.2.5.3 Prototyping in Mobile 34](#_Toc349918205)

[3.2.5.4 Prototyping in Web 35](#_Toc349918206)

[3.2.6 Data Modeling 35](#_Toc349918207)

[3.2.6.1 Detail ERD Diagram 35](#_Toc349918208)

[3.2.6.2 System Sequence Diagram 36](#_Toc349918209)

[3.2.7 Process Modeling 36](#_Toc349918210)

[3.2.7.1 Use Case narrative (Requirement specification) 36](#_Toc349918211)

[3.2.7.2 register account 37](#_Toc349918212)

[3.2.7.3 send message 38](#_Toc349918213)

[3.2.7.4 send message(business people) 39](#_Toc349918214)

[3.2.7.5 capture message 41](#_Toc349918215)

[3.2.7.6 check statistics 43](#_Toc349918216)

[3.2.7.7 change message location 44](#_Toc349918217)

[3.2.7.8 notify message being seen 45](#_Toc349918218)

[3.2.7.9 handle problematic message 46](#_Toc349918219)

[3.2.7.10 pays for advertisement message 48](#_Toc349918220)

# Introduction

## Deliverables in this report

Much has been changed after the first progress report. The real world problems being identified, the target audience of the proposed system, the primary objectives and scope of the system and the proposed solution are all become clearer. Therefore, it is essential to include a revised project proposal and revised first progress report in this report too to make documentation present in a consistent manner.

* Second progress report
* The revised project proposal
* The revised first progress report

Since first and second progress report share same format, they would be merged as one refine report.

## The changes since first progress report

The below is a summary paragraph from revised project proposal and revised first progress report. For detail, please read the revised project proposal and revised first progress report section.

### Changes of System Objectives

The system changes from a social network system backed by geolocation technology to a messaging system that is conditionally based. The system allows scheduling messaging based on conditions like geolocation, time and weather.

### Changes of System Audience

A clear target audience is identified. It would be a general-purpose messaging system for all walks of life. Beside of that, it would support scenarios for business to promote their products.

### Changes of proposed hardware and software configuration

Libraries and backend softwares are added or changed to support the implementation of the system.

## New Deliverables in Second progress report

### Scenarios

### Prototyping

### Technical Diagrams

Activity Diagrams for present the flow of different use cases in user perspective in analysis stage.

Sequence Diagrams for present the structure of different functions in system perspective in design stage.

### Interface Design

The color scheme use, the design language would be discussed.

# Revised Project Proposal

## Project Objectives

To identify the project objectives and related problems, we carry out CATWOE analysis.

|  |  |  |
| --- | --- | --- |
| **C**lients | General public | Business user |
| **A**ctors | Administrators | |
| **T**ransformation | allow send scheduling messages based on different set of user-defined conditions | |
| **W**orldview | general public: useful for relationship building | business user: useful for promoting products |
| **O**wner | Application software company | |
| **E**nvironmental constraints | Other social and messaging tools. Similar functions may appear in competitors' product very quick. | |

### Problem Identification

### Sending message is Instantaneous

There are already many applications in Internet that can send messages instantly. For example, SNS, Email, Whatsapp. While this message is essential to communication, people often find themselves feel overwhelming with messages[[1]](#footnote-1).New ways should be discovered so that people does not need to be distracted about sending or receiving messages. The new way should be complemented to existing Social Network and user does not spend double time for such way.

### Route of message transferring is not spread according to geolocation

Nowadays, messages in the Internet application are mainly spread from one point to another point directly. In contrast, in the old days, messages were spread by verbal; therefore, the old way is spread in a more continuous fashion, which is good for spreading public message that is meaningful to a neighborhood.

### User wants to control how receiver receive the message

Message is more meaningful if it is related to certain time, certain weather conditions, certain locations and/or with some people. But there are not many social tools cater the needs of user which let them to set in what situations the body of the message is revealed to the receiver freely.

## Proposed Solution

### Send different type of messages by user selection

The user can send messages based on conditions like weather, time and location. For location, the messages can be movable or fixed. The system tracks the location where user sends out the location message. Also, the system records the initial direction and speed that the movable message spreads by user input and possibly sensors. Real physical factors may (e.g Wind speed) affect the spread of the message afterwards. A message body may contain text, image or sound.User controls the exposure of their identity in the message they sent. That is, they determine who may receive the message and the amount of personal information they can obtain from that message.

### Receive different type of messages by user selection

The system will allow user to receive message that is near the user’s location or reveal a message by going to specific location according to sender setting. The system should provide a user-friendly and attractive way for user to receive message nearby. User may choose to add more content to a message and resend the message. Also, message is revealed according to weather or date and time according to sender settings.

The above are the two essential functions that the system needs to provide. They have different implementation than what is usually find in conventional social network system. Some sent messages have actual locations in the map in a certain time. The system records the route, type (public or to certain type of people) and property of the message. By these two functions, the system can mimic the physical way of sending messages to unknown recipients. This support, for example, some person can lay messages in a sightseeing place and later his friends rediscover the messages when they visit that place later on.

### Business User specified functions

Business User can send promoting based on conditional selection. They can broadcast message near their store.

### Administration monitoring & logging functions

Administrators can response to spam message report by user and obtain statistics of the use of different types of messages by users.

## Scope Definition

### Project Name

Message Capsule – Conditional-based Messaging System

### Project Overview

This project aims to:

### Provide a messaging system that is usable in mobile and web.

### Support new scenarios by enabling scheduling messaging based on mix of conditions

### Project Scope

This project will create a system which allows user to send messages by mainly three conditions- geolocation, time and weather. The sender can set different conditions based on their preferences. The sender can decide when, where and who, what the weather condition is to allow the reveal of the message body to the receiver. The sender can decide whether the message is a movable or immovable message. The system has to figure the user’s location. The movable messages are moving along the map according to wind speed and user’s input from sensors data. Another user may receive movable message by searching around his/her location with camera and decide to reply it. An achievement system will be made to display usage statistics to user. The system can allow both mobile and web client. The project will be finished in around six months.

## Project Plan

### Feasibility Study

Feasibility Study is consisted of six areas.

### Operational Feasibility

### It is whether the user can use the supporting system to solve the problems in the supported system. In this case, it is whether the messaging system can solve the communication problem of user in the real world.

### Cultural Feasibility

Different stakeholders of the system may have different worldview and what the aims of the system they think vary. Their worldview may also make them reluctant to adopt the use of the new system being implemented. In this case, it is whether the user would accept using this new messaging system to send and receive conditional-based messages.

### Technical Feasibility

It is the study of whether current technology and ability of the developers can support the implement of system. In this case, it is whether the system can support user location detection, data synchronization between server and clients.

### Economic Feasibility

It studies the cost to develop, the break-even point and the business opportunity of the system. In this case, it is whether the messaging system can allow potential business values and generate profit in some specific time frame.

### Schedule Feasibility

It studies whether the implementation can be finished on time. It is crucial because a delayed system may make a company loses time advantage.

### Legal Feasibility

It studies whether the system has legal issues or may arouse potential controversy because of modules the system built on or the purpose the system supports.

### Candidate Systems Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| Characteristics | Candidate 1 | Candidate 2 | Candidate 3 |
| Portion of System Computerized | A Web-based solution | A solution using native mobile apps | Same as candidate 2 |
| Benefits | Cross-platform, no need to install before using | Support all functions required | Cross-platform,  Support all functions required |
| Servers and Workstations | Server:  Windows PC  Client:  Any modern web browser | Server:  Windows PC  Client:  Mobile Devices – Android | Server:  Windows PC  Client:  Mobile Devices – Android and IOS |
| Software Tools Needed | PHP,  Mysql,  HTML5,  Javascript&Jquery,  Facebook Javascript SDK | PHP  Mysql,  Eclipse,  Android SDK,  Facebook apps SDK | PHP,  Mysql,  Corona SDK – Lua,  Facebook apps SDK |
| Application Software | Notepad++ (any text editor),  Photopshop, illustrator | Notepad++ (any text editor),  Photopshop, illustrator | Notepad++ (any text editor),  Photopshop, illustrator |
| Method of Data Processing | Client/Server | Client/Server | Client/Server |
| Output Devices and Implications | Output as web pages in mobile web browser | Output in application screen | Output in application screen |
| Input Devices and Implications | Mobile Device Touch Screen,  Location detection via mobile network | Mobile Device Touch Screen,  Location detection via mobile network,  Sensors of gyroscope and accelerometer | Mobile Device Touch Screen,  Location detection via mobile network,  Sensors of gyroscope and accelerometer |
| Storage Devices and Implications | MySQL Server | MySQL Server | MySQL Server |

### Feasibility Analysis Matrix

The following table is the Feasibility Analysis Matrix according to the above 6 aspects of feasibility analysis of three candidates.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Weight | | Candidate 1 | Candidate 2 | Candidate 3 |
| Description | / | | Write new web application for both front-end and back-end | Write new web application for back-end, native apps for front-end | Write new web application for back-end, native apps for front-end, use paid SDK to support multi platform apps |
| Operational Feasibility | 15% | | Some functions may not fully support.E.g receive sensors’ data, fluid user interface and notifcation  Score: 70 | Support all the required functions  Score: 100 | Support all the required functions.  Score: 100 |
| Cultural Feasibility | 15% | | The user experience may not be optimal due to the use of web app interface  Score: 90 | No foreseeable problems  Score: 100 | No foreseeable problems  Score: 100 |
| Technical Feasibility | 20% | | Multiple platforms in both PC and mobile devices can be support with same code.  Limited support to access mobile device location API and sensors.  Score: 70 | Native app provides better performance and visual effects.  Require more time to support new platform  Score: 80 | Multiple mobile platforms can be support with same code. Performance and visual effects are in between candidate 1 and 2.  Score: 75 |
| Economic Feasibility | 30% | | No special expense  Score: 80 | Require more expense to if support of multiple platforms is need  Score: 70 | Require purchase of the SDK for business usage  Score: 60 |
| Schedule Feasibility | 10% | | Five months  Score:80 | Six months  Score:70 | Five months  Score:80 |
| Legal Feasibility | 10% | | No foreseeable problems  Score:100 | No foreseeable problems  Score:100 | No foreseeable problems  Score:100 |
| Weighted Score |  | 80 | | **84**  **Candidate 2 is chosen.** | 81 |

Table 1Feasibility Analysis Matrix

### Development Schedule Project Development Schedule

### Cost Estimation

|  |  |  |
| --- | --- | --- |
| Development Costs | |  |
|  |  |  |
| Personnel: |  |  |
| 1 | System Analyst | $ 60,000 |
| 1 | Programmer | $ 36,000 |
| 1 | Graphic Designer | $ 36,000 |
|  |  |  |
| New hardware & software: | |  |
| 1 | Development Server | $ 3,000 |
| 1 | Server software(OS, application software) | $ 3,000 |
|  |  |  |
| Total Development Costs  $138,000 | | |
|  |  |  |
| Projected Operation Costs | |  |
| 1 | System Administrator | $ 12,000 |
| 1 | Programmer | $ 9,000 |
|  |  |  |
| Monthly Operation Costs  $ 21,000 | | |

# Revised First Progress Report and Second Progress Report

## System Analysis

### Source of Information

### Observation from App store

In the App store, in the Top Free in Social section, it may observe what social tools companies have released and tried to attract user to use them. It is being observed that no popular social tools are having scheduling conditional-based messaging functions. While there are many messaging scheduling apps in the app store, they only provide scheduling messaging by date and time only but not geolocation or weather. On the contrary, there are many Location-based social tools too. However, they do not include the scheduling messaging option. Therefore, these current apps cannot support some real scenarios that user may want to use. For example, tell receiver to receive the message in specific location in specific time and date.

### Psychology Research

Some Research [[2]](#endnote-1)has found that instant text messaging may harm our productivity. While supporting new scenarios for messaging, it should provide flexibility for user to set what degree of automatic notification they would like to have.

### Modeling Method Comparisons

There are different modeling and analysis and design methods to choose from. It can be divided into two main types:

### Structured analysis and design method

* Model-Driven
* Process-oriented

### Object-oriented analysis and design method

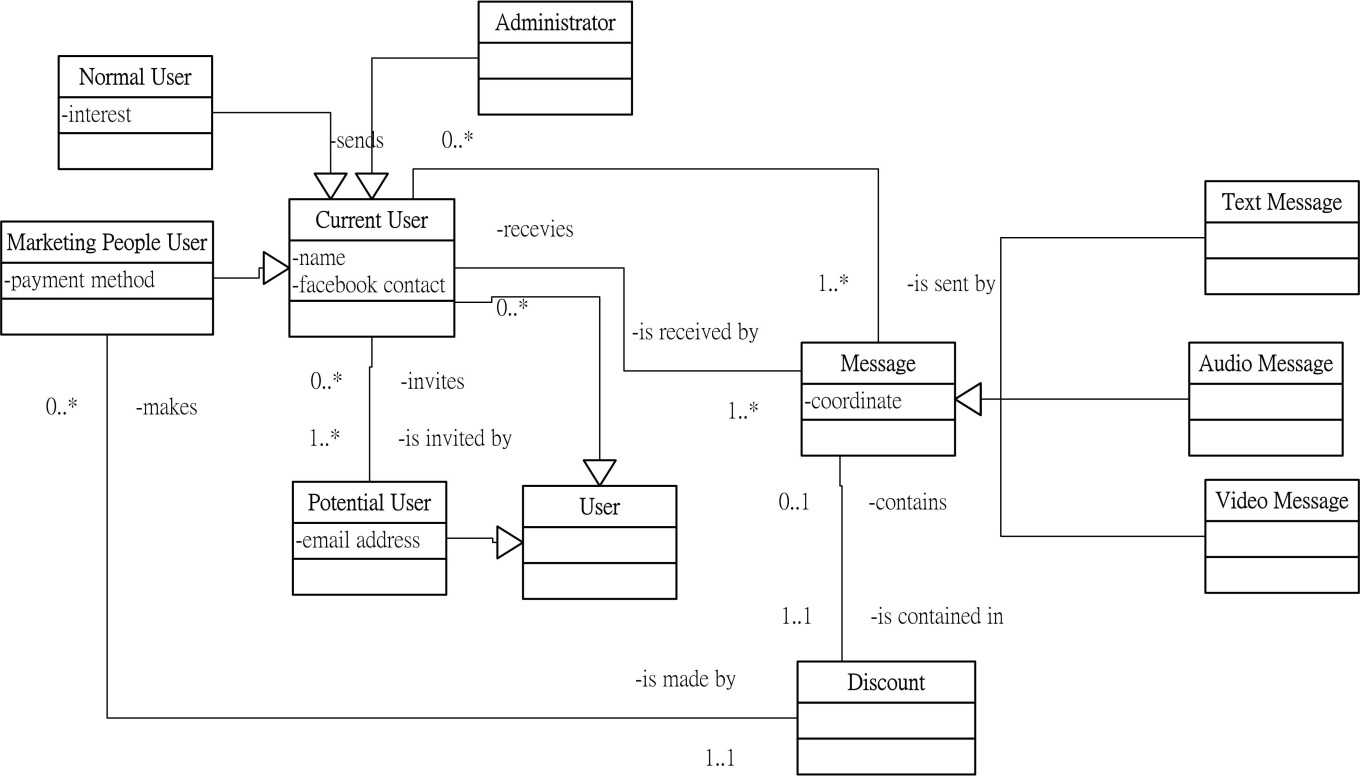
* Model-Driven
* UML, a standard language for all modeling work
* Modeling represent by class and objects

### Chosen Modeling Method

In the project proposal section, the candidate chose use Android SDK for development, which underlying programming language is JAVA. Java is an object-oriented language. Because of this, modeling by OOAD gives more convenience for implementation. Therefore, Object-oriented analysis and design method is chosen for modeling.

### Data Modeling

### Class Diagram



### Entity Relationship Diagram

The database is a traditional RDBMS. It is necessary to translate the class diagram as entity relationship diagram for SQL manipulation.



### Entity Relationship Diagram Dictionary

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Entity Name: | **User** | | | | | |
| Entity Description: | To store user's information | | | | | |
| **Name** | **Key** | **Data Type** | **Description** | | | |
| userid | primary | integer | User ID | | | |
| email | N/A | text | email of user | | | |
| firstname | N/A | text | First Name of user | | | |
| lastname | N/A | text | Last Name of user | | | |
| usertype | N/A | integer | Type of user | 1=normal user | 2=business user | 3=administrator |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Entity Name: | **Message** | | | | | | | | | | | |
| Entity Description: | To store messages to be sent or received | | | | | | | | | | | |
| **Name** | **Key** | **Data Type** | **Description** | | | | | | | | | |
| messageid | primary key | integer | Message ID | | | | | | | | | |
| senderid | forigen key | integer | User ID of sender | |  | | | | | | | |
| receiverid | forigen key | integer | User ID of receiver | |  | | | | | | | |
| messagebody | N/A | text | The content of the message | |  | | | | | | | |
| messagetype | N/A | integer | Type of message | | 1=text | | | 2=textWithImage | | | | |
| coordinate | N/A | float | the latitude of longitude of message | | | | | | | | | |
| weather\_condition | N/A | text | the weather condition message being revealed | 1=sunny | | 2=rain | 3=fog | | 4=snow | 5=storm | 6=humid | 7=dry |
| date\_to\_notify | N/A | date with time zone | the date notify the message to receiver | | | | | | | | | |
| date\_to\_hidden | N/A | date with time zone | the date to hidden the message | | | | | | | | | |
| movable | N/A | Boolean | whether the message is movable | | | | | | | | | |
| temperature | N/A | integer | the temperature for the message to be revealed | | | | | | | | | |
| temperature\_lower\_or\_higher | N/A | Boolean | whether the message reveal in a temperature higher or lower to the current temperature | | | | | | | | | |
| createtime | N/A | date with time zone | the date the message is created | | | | | | | | | |

### Entity Relationship Diagram (Refined)

The ERD in 3.1.4.2 is not flexible for further modification. It is better to separate weather condition, coordinate, image to a new entity so it supports new weather conditions in future and multiples coordinates to record the location of movable messages in different period. Also, multiple and different kinds of multimedia files can be supported in a message.

### Entity Relationship Diagram Dictionary (Refined)

### Process Modeling

### Event Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Event Number | Event Description | System Input | Actor Providing Input | System Output | Actor Receiving Output |
| 1 | User registers account | Personal information, interests | User | Registration result | User |
| 2 | User invites friends | friends's email in contact list | User | Invitation | Potential Users |
| 3 | User sends message. | Message body (text/image), send conditions(weather, date, location-related) | User(Sender) | New POI (Point of Interest) in Map | User(s)(Receiver) |
| 4 | User captures message | Request for surrounding message or specific date or weather fulfill | User(Receiver) | POI in surrounding | User(Receiver), User(Sender) |
| 5 | User shares message | Message that user wants to share, users to be shared | User | POI | User |
| 6 | User checks statistics | Request for user's statistics | User | user's statistics | User |
| 7 | System changes message directions | wind direction and wind speed (weather web services) | Time | update location of POI | User(s)(Receiver) |
| 8 | System notifies message being captured | Message being captured | System | Notification | User(Sender) |
| 9 | Admin handles report of spam message | Request for user report | Admin | update list of POI | User(s)(Receiver) |
| 10 | Business People sends messgae | Advertisement method, advertise message | Business People | New POI (Point of Interest) in Map | User(s)(Receiver) |
| 11 | Business People pays for advertisement message | Payment information | Business People | Payment result | Business People |

### Event Table Compact List

### User registers account

### User invites friends

### User sends message

### User captures message

### User shares message

### User checks statistics

### System changes message directions

### System notifies message being captured

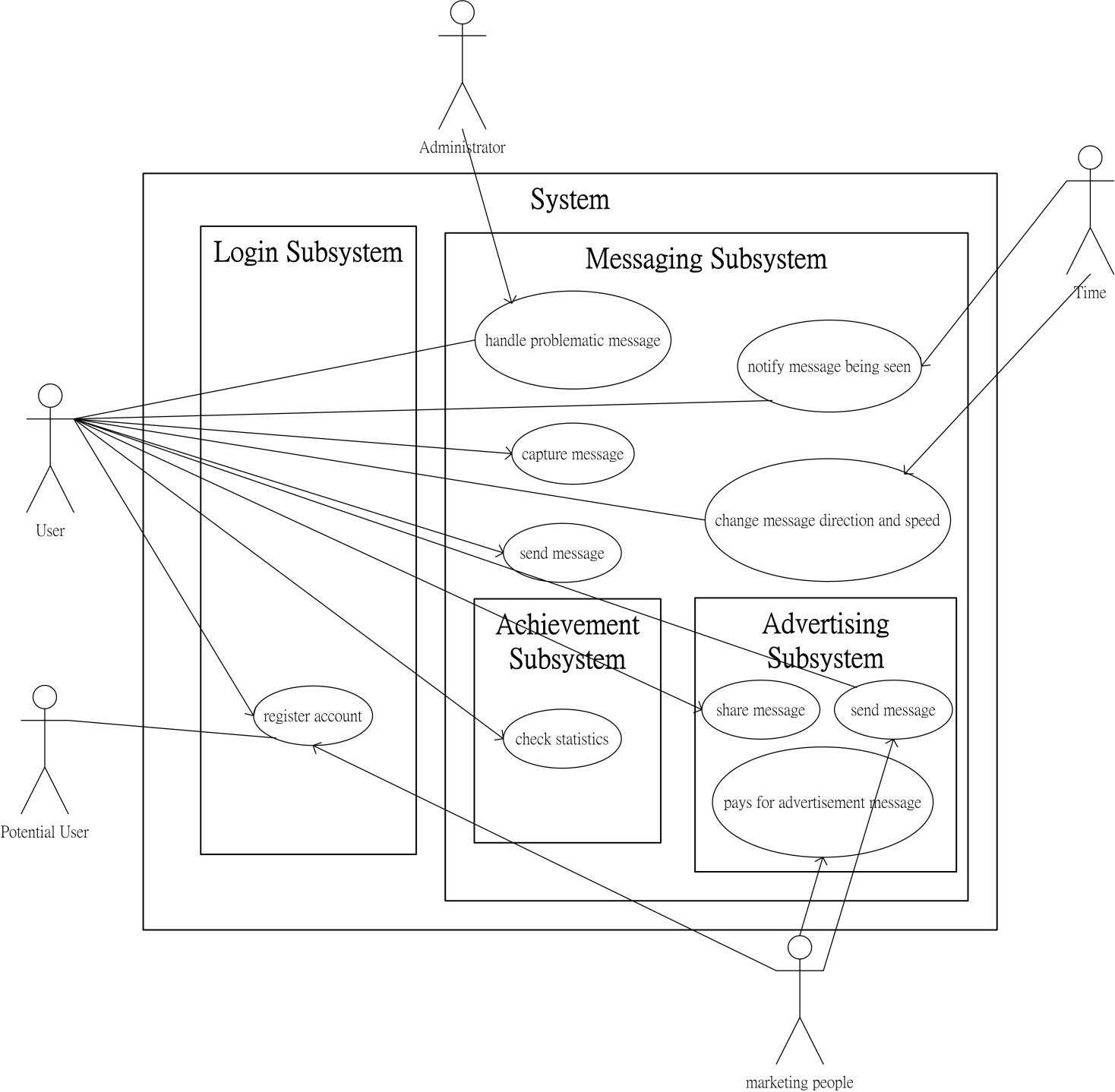
### Admin handles report of spam message

### Business People sends message

### Business People pays for advertisement message

### Use case diagram

This also shows the functional decomposition by dividing use cases into different subsystems.



### Activity Diagram

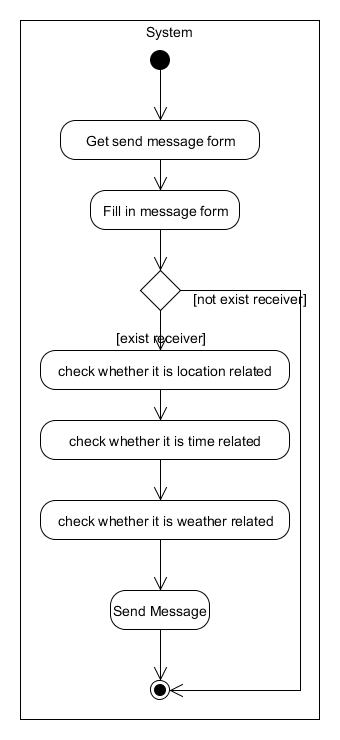


Figure 1Send Message Activity Diagram

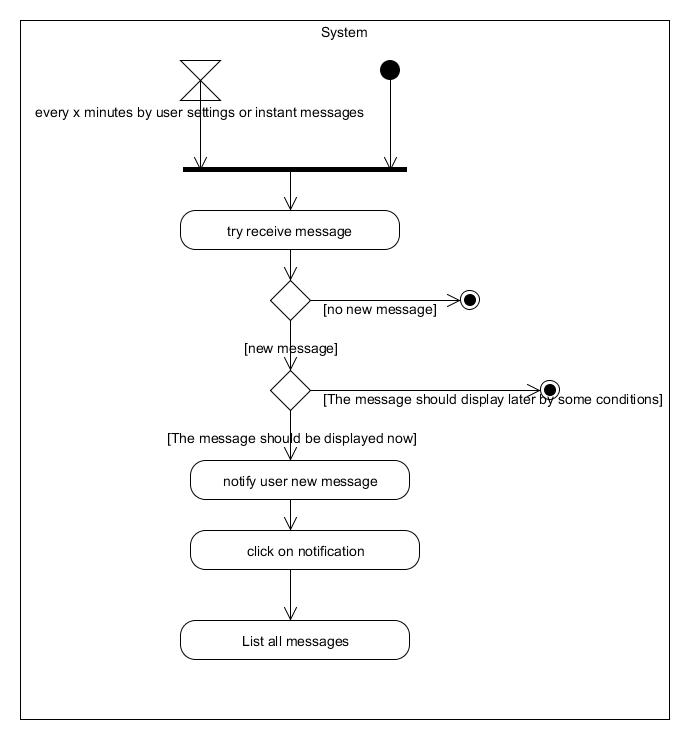


Figure 2Receive Message Activity Diagram

### Sequence Diagram

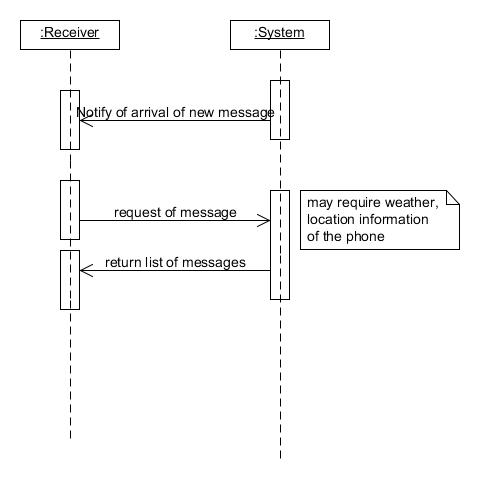


Figure 3Send Message Sequence Diagram

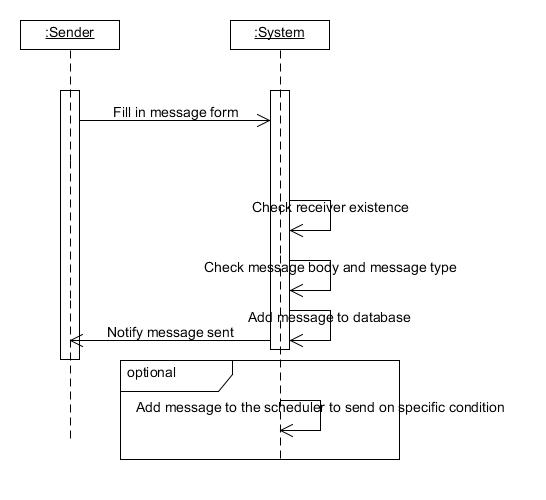


Figure 4Receive Message Sequence Diagram

### Functional Decomposition Diagram

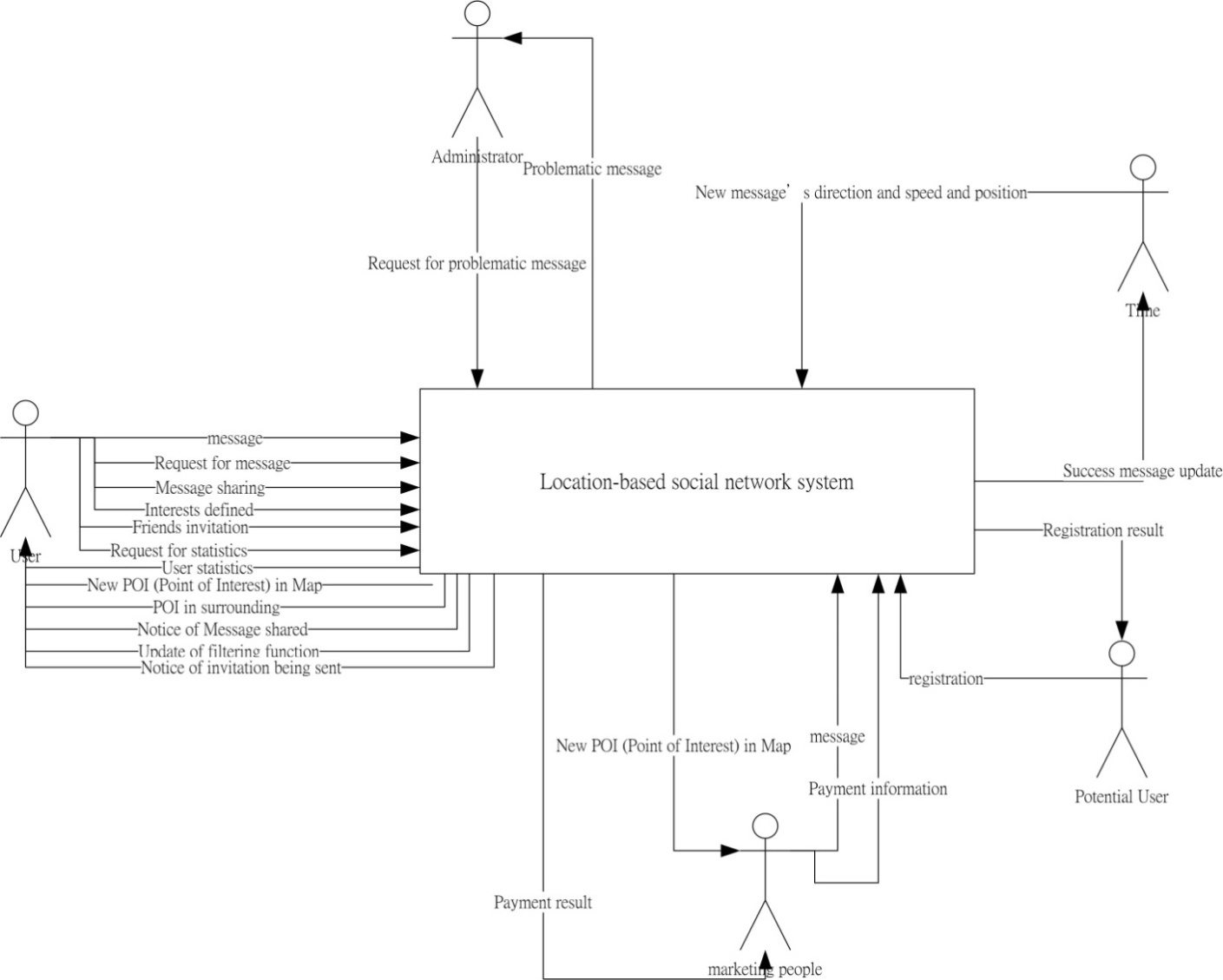
1. Login Subsystem

* 1. Login and Logout
  2. Find friends by contact list
  3. Invite Friends
  4. Register Account

1. Message Subsystem
   1. Send Message
   2. Reply Message
   3. Share Message
   4. Report Spam Message
   5. Notify user to read new message
   6. Update movable message location
   7. Receive Message
2. Business Subsystem
   1. Pays for advertising message
   2. Sends advertising message
3. Achievement Subsystem
   1. Check Statistics
4. Administrator Web Subsystem
   1. Monitor overall user usage
   2. Handle spam message report
   3. Get Frequent Topics of user
   4. Get Messages’Geographical Distribution

### 

### Context Data Flow Diagram



## System Design

### Finalized hardware and software configurations

|  |  |
| --- | --- |
| Portion of System Computerized | A solution using native mobile apps |
| Benefits | Support all functions required |
| Servers and Workstations | Server:  Windows PC with JavaEE  Client:  web browser, Mobile Devices – Android  connect with the server using HTTP |
| Software Tools Needed | postgresql,  Android SDK,  Javascript&Jquery,  Wikitude |
| Application Software | Notepad++ (any text editor),  Photopshop, illustrator |
| Method of Data Processing | Client/Server |
| Output Devices and Implications | Output as web pages in web browser, output as web pages and native screen in mobile application |
| Input Devices and Implications | Touch Screen,  Location detection via mobile network,  Sensors of accelerometer, compass |
| Storage Devices and Implications | Postgresql |

### Detail of software technology use

|  |  |  |  |
| --- | --- | --- | --- |
|  | Software/Technology Name | Description | Home Link |
| Backend | Jave EE | The backend sever |  |
| Tomcat | The JAVE EE server container |  |
| Postgresql | The DBMS being used | <http://www.postgresql.org> |
| Postgis | The plug-in to enable spatial data storage in DBMS | <http://postgis.refractions.net> |
| Frontend | Android | Serve the client in mobile |  |
| JSP | Serve the client in web |  |
| Development Tools | Eclipse | The IDE for development in server and client-side |  |
| Services and Libraries and Other Resources Usage | Openstreet Map API | The API for location finding | <http://wiki.openstreetmap.org/wiki/Nominatim> |
| Yahoo! Weather API | The API for weather finding | <http://developer.yahoo.com/weather/> |
| Google Cloud Messaging | The library for data synchronization | <http://developer.android.com/google/gcm/index.html> |
| Basic HTTP Client | The library for data exchange | <http://code.google.com/p/basic-http-client/> |
| Google Map API | The API for map showing | <https://developers.google.com/maps/> |
| Flickr API | The API for location finding | <http://www.flickr.com/services/api/flickr.places.findByLatLon.html> |
| Android Icon Templates Pack | The free icon set for android user interface | [http://developer.android.com/guide/practices/ui\_guidelines/icon\_design.html#templatespack](http://developer.android.com/guide/practices/ui_guidelines/icon_design.html%23templatespack) |
| GLYPHICONS | The free icon set for user interface | <http://glyphicons.com/> |
| Bootstrap | The template for web user interface | <http://twitter.github.com/bootstrap/> |
| Wikitude | Augmented Reality SDK | <http://www.wikitude.com> |
| Iconfinder | An icon finder | <http://www.iconfinder.com> |
| Foundation | The Front-end framework for web user interface | <http://foundation.zurb.com> |
| Leaflet | Javascript Library for mobile-friendly interactive maps | <http://leafletjs.com> |
| Android SDK | The SDK for android apps development |  |

### Scenarios

The following are some scenarios for Message Capsule system to be applied in the real world.

* For City Hunt
* For Relationship Building
* For Business Promotion

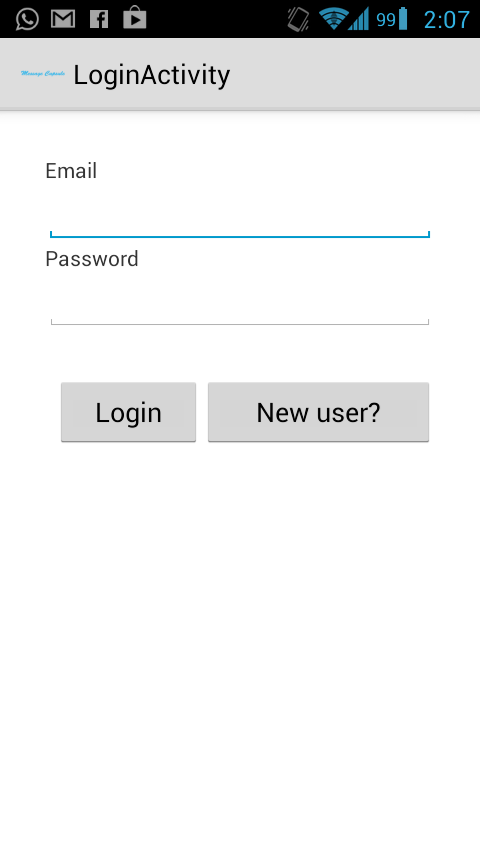
### Input Design

### User log into the Message Capsule System

Description

The login screen requires user to input their email and password in order to login into the system.

|  |  |  |
| --- | --- | --- |
| Details of input control | | |
| Name | Type of control | Description |
| Email | Text Box | User input email |
| Password | Text Box | User input password |
| Login | Button | Click the button to submit the login form |
| New User? | Button | Click the button to go to register screen |

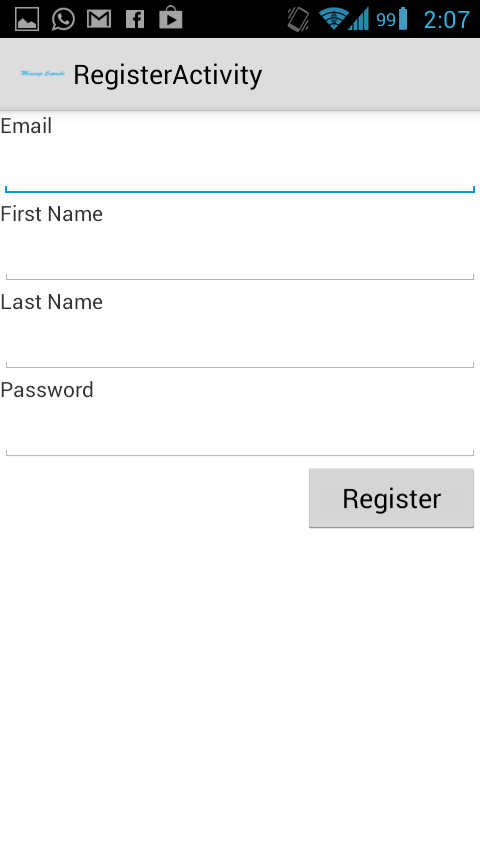


### User registering account

Description

The register screen requires new user to input basic information to register the account.

|  |  |  |
| --- | --- | --- |
| Details of input control | | |
| Name | Type of control | Description |
| Email | Text Box | New User input email |
| First Name | Text Box | New User input email |
| Last Name | Text Box | New User input email |
| Password | Text Box | New User input email |
| Register | Button | Click the button to submit the register form |



### User sending message in web

### User sending message in mobile

Description

The send message screen requires user to input message body and choose to enable certain conditional-based options.

|  |  |  |
| --- | --- | --- |
| Details of input control | | |
| Name | Type of control | Description |
| Recipient Email | Text Box | Sender inputs the receiver email |
| Message body | Text Box | The content of the message |
| Message reveal by Date? | Check Box | Choose whether the message is revealed by Date |
| Message reveal by Location? | Check Box | Choose whether the message is revealed by Location |
| Message reveal by Weather? | Check Box | Choose whether the message is revealed by Weather |
| Date selection box | Date selection box | Choose the day to reveal the message |
| Weather Condition | Drop down list | Choose the weather condition to reveal the message |
| Temperature higher or lower | Radio button | Choose the temperature higher than or equal or lower than to reveal the message |
| Temperature selection | Slider | Choose the temperature to reveal the message |

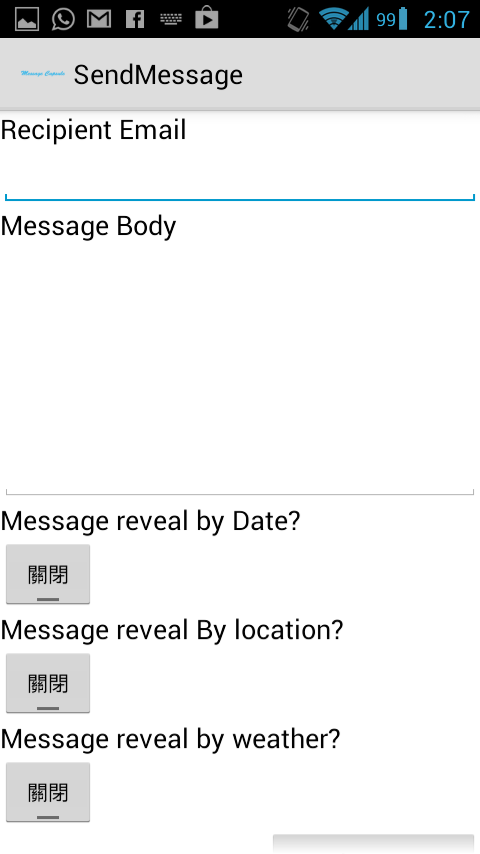
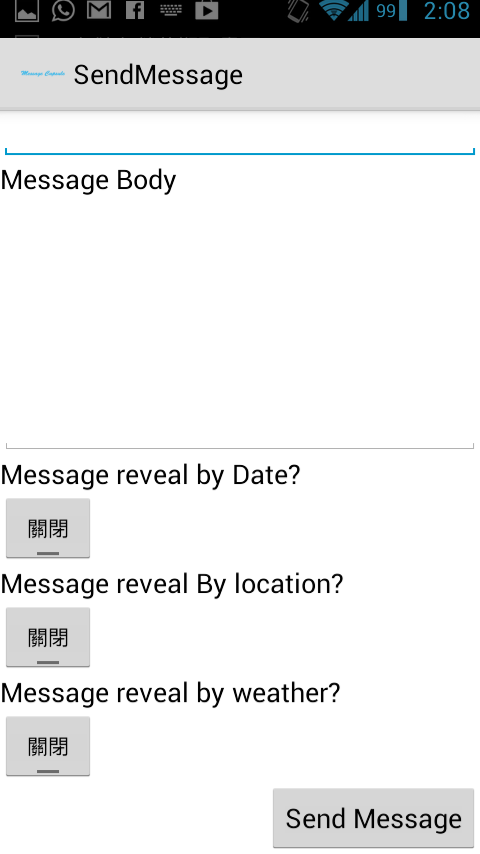


Figure 5 Send Message Screen

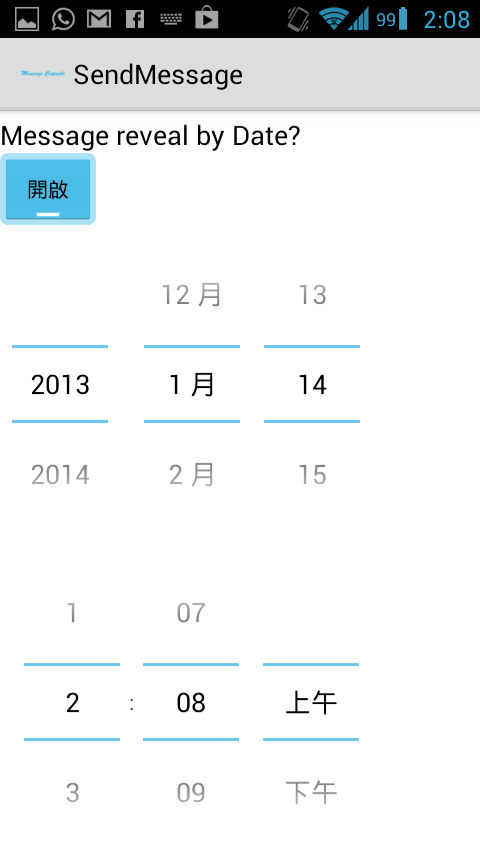


Figure 6Send Message Screen (expanded)

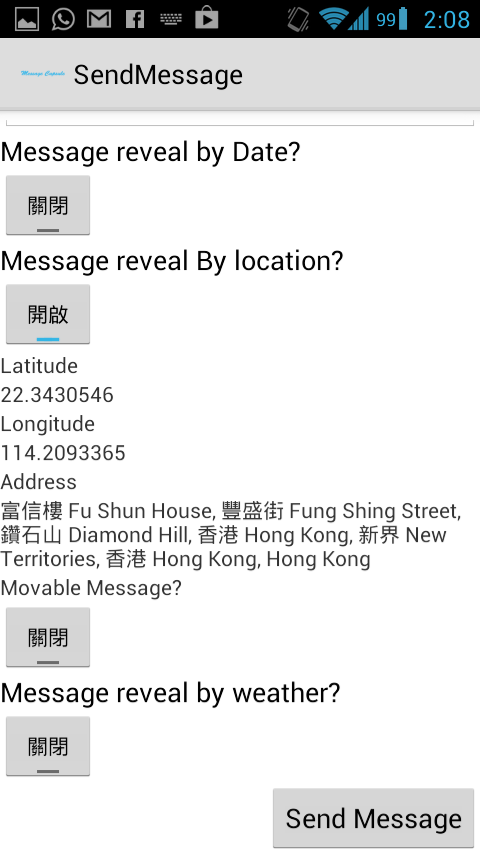
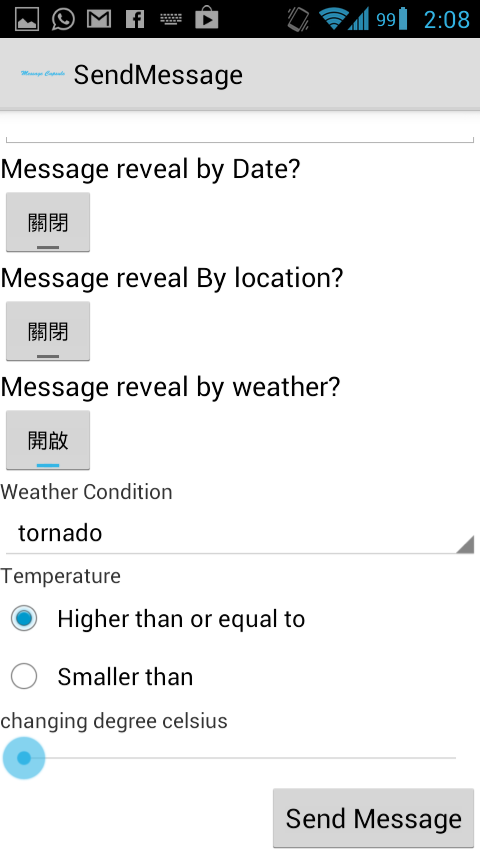


Figure 7Send Message Screen (expanded)



### User changing settings

Figure 8Send Message Screen (expanded)

### Output Design

### User receiving message in web

### User receiving message in mobile

### User checking own usage statistics

### Administrator monitoring overall user usage

### Interface Design

### Logo Design



This is the current logo design that appears as the application icon.

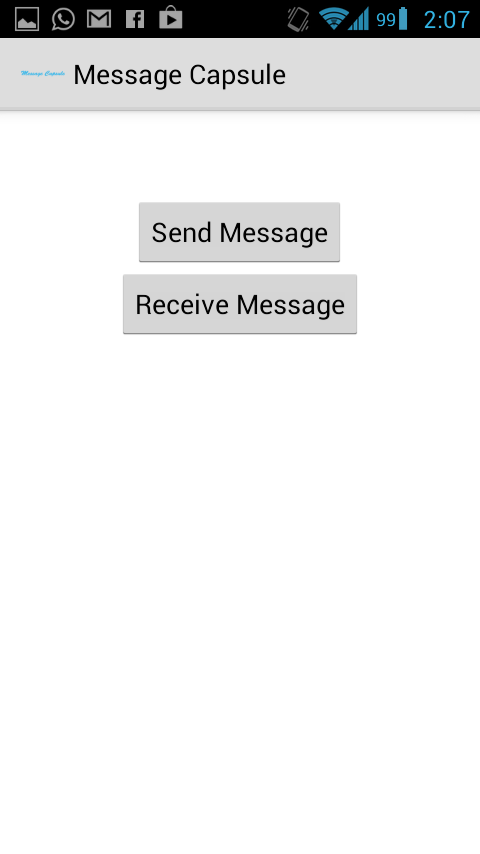
### Color scheme

There are different color scheme that can be used. The most common are:

* Mono Color Scheme
* Complement Color Scheme
* Analog Color Scheme

It is thought that **monotonous color scheme** gives an impact that the software is more reliable. Therefore, this scheme would be used for the overall interface design. However, for receiving message screen, multiple **analog color schemes** are used to give sender more control in what the receiver can see when they read the messages.

### Prototyping in Mobile



This is the main screen of prototype in mobile which it uses standard Android GUI. It makes user easily to learn how to use the application because there is consistency to the whole Android system experience.

### Prototyping in Web

### 

This is the main screen of prototype in web which it uses Twitter Bootstrap for interface development. Twitter Bootstrap is a library to provide standard web elements like menu bar, breadcrumbs, list so that user can navigate using the web version of the system easily.

### Data Modeling

### Detail ERD Diagram

In design stage, a detail ERD diagram is drawn to reflect implementation design.



### System Sequence Diagram

In the previous analysis stage, sequence diagram is drawn. In design stage, detail system sequence diagrams are drawn to represent the internal operations in the system to support each business activity.

### Process Modeling

### Use Case narrative (Requirement specification)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Use Case Name: | register account | |  |  |  |
| Primary Business Actor: | potential user |  |  |  |  |
| Other Participating Actors: | user, Facebook(external) | |  |  |  |
| Description: | This use case describes the event of a potential user registering as the system user. On completion, he or she may imports his/her interests directly from facebook or define interests by their own. The facebook contact also allows registered users to share this system to his/her facebook friends. | | | | |
|  |  |  |  |  |  |
| Precondition: | the potential user is currently not a user in the system. | | | | |
| Trigger: | the potential user selects the register account function or being invited by current user. | | | | |
| Typical Couse of Event: | Actor Action | System Response | |  |  |
|  | Step 1.The potential user provides login name and password and email addresses | Step 2. The system validates the information and request for facebook linkage. | | | |
|  | Step 3. The user gives out permission to access facebook information. | Step 4. The system obtains user interests and let users to choose and refine. | | | |
|  | Step 5. The user chooses interests to retain or add and submit. | Step 6. The system creates an account for the user and update database. | | | |
|  |  | Step 7. The system asks for user to invites more people to use the system. | | | |
|  | Step 8. The user invites potential users via facebook or email address | | | | |
|  |  |  |  |  |  |
| Conclusion: | The use case concludes when user creates an account. | | | | |
| Postcondition: | an new account create for user. | | |  |  |
| Business rules: |  |  |  |  |  |
| Implementation constraints and specifications: |  |  |  |  |  |
| Assumptions: |  |  |  |  |  |
| Priority: | High |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case Name: | send message |  |  |  |  |  |
| Primary Business Actor: | user(sender) |  |  |  |  |  |
| Other Participating Actors: |  |  |  |  |  |  |
| Description: | This use case describes the event of a user(sender) sending a new message. User(receiver) sends the current geolocation to the system, accompany with some filtering options and the message body. | | | | | |
|  |  |  |  |  |  |  |
| Precondition: | user(sender) must enables GPS/WIFI to allow detection of geolocation | | | | | |
| Trigger: | This use case is initiated when user(sender) selects the send message function | | | | | |
| Typical Couse of Event: | Actor Action | System Response | |  |  |  |
|  | Step 1: The user(sender) provides his or her geolocation | Step 2: The system responds by giving a sending form. | | | | |
|  | Step 3: The user provides his or her desired message form (short or long distance) and type of message body ( text, audio, video) and desired group of people to see the message. | Step 4: The system responds by verifying that all required information has been provided. | | | | |
|  | Step 5: The user inputs the desired message direction by motion sensor. | Step 5: The system responds by calculating the corresponding direction and distance. | | | | |
|  |  | Step 6: The system adds a new Point of interest to the database. | | | | |
|  | Step 7: The user acknowledges that the message has been added. | | | | | |
|  |  |  |  |  |  |  |
| Conclusion: | This use case concludes acknowledges that the message has been added. | | | | | |
| Postcondition: | A new Point of interest adds. The user statisitics increments. | | | | |  |
| Business rules: |  |  |  |  |  |  |
| Implementation constraints and specifications: | animation of throwing messages and usage of sensors in phone | | | | |  |
| Assumptions: |  |  |  |  |  |  |
| Priority: | High |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name: | send message(business people) | | |
| Primary Business Actor: | business people | |  |
| Other Participating Actors: |  |  |  |
| Description: | This use case describes the event of a business people sending a new message. User(receiver) send the desired geolocation, some filtering options and the message body. | | |
|  |  |  |  |
| Precondition: | business people must provide payment information to pay for the advertising message. | | |
| Trigger: | This use case is initiated when business people selects the send message function | | |
| Typical Couse of Event: | Actor Action | System Response | |
|  | Step 1: The user(sender) provides his or her geolocation | Step 2: The system responds by giving a sending form. | |
|  | Step 3: The user provides his or her desired message form (short or long distance) and type of message body ( text, audio, video) and desired group of people to see the message and/or discount and/or amount of messages to be spread | Step 4: The system responds by verifying that all required information has been provided. | |
|  | Step 5: The user inputs the desired message direction directly. | Step 6: The system adds a new Point of interest to the database. | |
|  | Step 7: The user acknowledges that the message(s) has been added. | Step 8: The system lead business people to pay for the advertise message | |
| Conclusion: | This use case concludes when the system lead business people to pay for the advertise message | | |
| Postcondition: | A new Point of interest adds. The user statistics increments. Payment for business people added. | | |
| Business rules: |  |  |  |
| Implementation constraints and specifications: | usegoogle map view for business people to select location they want to add the message. | | |
| Assumptions: |  |  |  |
| Priority: | Medium |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Use Case Name: | capture message | | | | | | | | |
| Primary Business Actor: | user(receiver) |  |  |  |  |  |  |  |  |
| Other Participating Actors: |  |  |  |  |  |  |  |  |  |
| Description: | This use case describes the event of a user(receiver) requesting to capture message. User(receiver) sends the current geolocation to the system, accompany with some filtering options. On completion, the system returns surround message that the user(receiver) desires to see. | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |
| Precondition: | user(receiver) must enables GPS/WIFI to allow detection of geolocation | | | | |  |  |  |  |
| Trigger: | This use case is initiated when user(receiver) selects the capture message function | | | | | |  |  |  |
| Typical Couse of Event: | Actor Action | System Response | |  |  |  |  |  |  |
|  | Step 1: The user(receiver) provides his or her geolocation and filtering options(optional) | Step 2: The system responds by verifying that all required information has been provided. | | | | | | | |
|  |  | Step 3: The system returns list of surrounding message according to required information | | | | | | | |
|  | Step 4: The user filters returns messages by checking/unchecking boxes (e.g friends only, similar interests, similar personalities) in the filter panel. | Step 5: The system returns list of surrounding message according to filtering options. | | | | | | | |
|  |  | Step 6: The system requests for update of user's geolocation in regular interval. | | | | | | |  |
|  | Step 7: The user provides his or her latest geolocation. | Step 9: The system returns list of surrounding message according to new geolocation | | | | | | | |
|  | Step 10: The user selects a message and read and/or reply it. | Step 11: The system receives the read action and/or replies and update the database accordingly. | | | | | | | |
|  | Step 12: The user quits the function. | |  |  |  |  |  |  |  |
| Alternate Courses: | (If user runs the apps in background) | |  |  |  |  |  |  |  |
|  | Step 1: The user(receiver) provides his or her geolocation and filtering options(optional) | Step 2: The system responds by verifying that all required information has been provided. | | | | | | | |
|  |  | Step 3: The system returns list of surrounding message according to required information( e.g match the desired group of people to see the message ) | | | | | | | |
|  | Step 4: The user filters returns messages by checking/unchecking boxes (e.g friends only, similar interests, similar personalities) in the filter panel. | Step 5: The system returns list of surrounding message according to filtering options. | | | | | | | |
|  |  | Step 6: The system requests for update of user's geolocation in regular interval. | | | | | | |  |
|  | Step 7: The user provides his or her latest geolocation. | Step 9: The system returns list of surrounding message according to new geolocation | | | | | | | |
|  | Step 10: The user selects a message and read and/or reply it. | Step 11: The system receives the read action and/or replies and update the database accordingly. | | | | | | | |
|  | Step 12: The user quits the apps. | |  |  |  |  |  |  |  |
| Conclusion: | This use case concludes when user(receiver) exits the function. | | | |  |  |  |  |  |
| Postcondition: | The receive of message increment user(receiver)'s statistics. New replied message is inserted into the database. | | | | | | | |  |
| Business rules: |  |  |  |  |  |  |  |  |  |
| Implementation constraints and specifications: | AR view to be provided for user selecting surround message in Typical Couse | | | | |  |  |  |  |
| Assumptions: |  |  |  |  |  |  |  |  |  |
| Priority: | High |  |  |  |  |  |  |  |  |
| Open issues: | 1. how many and how far of surrounding message should be returns to user at a single time. | | | | | |  |  |  |
|  | 2. There is an existing plugin to allow ease of using ar view, but it would require license if it is for commerical use. | | | | | | | |  |
|  | 3. When should the system requests for update of user's geolocation. | | | | |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | check statistics | |
| Primary Business Actor: | user |  |
| Other Participating Actors: |  |  |
| Description: | This use case describes the event of a user checking his or her own usage statistics and achievement in the system. | |
|  |  |  |
| Precondition: | The user is a registered user. | |
| Trigger: | This use case is initiated when user(sender) selects the check statistics function | |
| Typical Couse of Event: | Actor Action | System Response |
|  |  | Step 1. The system returns a compact view of user statistics. |
|  | Step 2. User selects a specific item of statistics | Step 3. The system returns a detail view of that statistics/achievement. |
|  |  |  |
| Conclusion: | The use case concludes when the user exits this function. | |
| Postcondition: |  |  |
| Business rules: | More options are given to user for sending and capturing message according to their usage of the system. | |
| Implementation constraints and specifications: |  |  |
| Assumptions: |  |  |
| Priority: | Medium |  |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | change message location | |
| Primary Business Actor: | Time |  |
| Other Participating Actors: | weather web services( external) | |
| Description: | This use case describes the time event of changing message location by retrieving weather conditions from web weather services | |
|  |  |  |
| Precondition: |  |  |
| Trigger: | Certain interval | |
| Typical Couse of Event: | Actor Action | System Response |
|  |  | Step 1.The system request for wind speed from web services in certain region of the message. |
|  |  | Step 2. The system calculated the new position of the message. |
|  |  | Step 3. The system updates the position of the message. |
| Conclusion: | The user case concludes when the position of all long distance messages are updated. | |
| Postcondition: | update long distance message position | |
| Business rules: | Only long Distance message would change message location regularly | |
| Implementation constraints and specifications: |  |  |
| Assumptions: |  |  |
| Priority: | Medium |  |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | notify message being seen | |
| Primary Business Actor: | Time |  |
| Other Participating Actors: | User(sender), User(receiver) | |
| Description: | This use case describes the time event of user(sender) being notified that the message sent has been capture and/or replied by user(receiver). On completion, user(sender) may add user(receiver) in their contact and further reply. | |
|  |  |  |
| Precondition: | User(sender) sends a message and user(receiver) captures that message. | |
| Trigger: | This use case is initiated when user(receiver) captures a message. | |
| Typical Couse of Event: | Actor Action | System Response |
|  | Step 1. user(receiver) captures a message and replies it. | Step 2. System acknowledges the capture to user(sender) and returns the reply to user(sender). |
|  | Step 3. user(sender) choose to add user(receiver) in the friend list. | Step 4. System adds the user(receiver) contact to user(sender) friend list. |
|  |  |  |
| Alternate Courses: | Step 1. user(receiver) captures a message. | Step 2. System acknowledges the capture to user(sender). |
|  | Step 3. user(sender) gets notification. | |
|  |  |  |
| Conclusion: | This use case concludes when user(sender) read the notification. | |
| Postcondition: | user(sender) gets notification. | |
| Business rules: |  |  |
| Implementation constraints and specifications: | the notification is shown on the notification bar of smartphone system. | |
| Assumptions: |  |  |
| Priority: | Medium |  |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | handle problematic message |  |
| Primary Business Actor: | Administrator |  |
| Other Participating Actors: | User |  |
| Description: | This use case describes the event of an administrator handling problematic message. The existing and new problematic message flagged by either administrator and user is shown to the administrator. Administrator judges the message flagged. The message would be removed if it is inappropriate. Administrator has the right to forbidden user to use the system again as penalty. | |
|  |  |  |
| Precondition: | Some message is flagged by user. |  |
| Trigger: | This use case is initiated when administrator requests to check any flagged message list. | |
| Typical Couse of Event: | Actor Action | System Response |
|  | Step 1: The administrator requests for the list of flagged message. | Step 2: The system responds by retrieving the list of flagged message. |
|  | Step 3: The administrator selects some of the flagged message and requests to remove them with reason input. | Step 4: The system confirmation the removal of selected message from message. |
|  | Step 5: The administrator confirms the action. | Step 6: The system proceeds to remove the messages from user. |
|  |  | Step 7: The system informs both administrator and the message sender that message is removed. |
|  | Step 8: The administrator selects some of the flagged message and requests to retain them with reason input. | Step 9: The system proceeds to unflagged the messages. |
|  |  |  |
| Alternate Courses: | Step 1: The administrator requests for the list of flagged message. | Step 2: The system responds by retrieving the list of flagged message but there is no flagged message. |
|  |  |  |
| Conclusion: | This use case concludes when administrator removed or unflagged a problematic message. | |
| Postcondition: | The flagged message is either removed or unflagged. | |
| Business rules: |  |  |
| Implementation constraints and specifications: |  |  |
| Assumptions: |  |  |
| Priority: | Low |  |

|  |  |  |
| --- | --- | --- |
| Use Case Name: | pays for advertisement message | |
| Primary Business Actor: | business people | |
| Other Participating Actors: | (External) online payment system | |
| Description: | This use case describes the event of a business people sending a new message. Business people provides their payment information and pay for the message. | |
|  |  |  |
| Precondition: | business people must provide payment information to pay for the advertising message. | |
| Trigger: | This use case is initiated after business people finishes the send messages use case. | |
| Typical Couse of Event: | Actor Action | System Response |
|  | Step 1. The business people provides payment information | Step 2. The system redirects business people to the online payment system for payment. |
|  |  | Step 3. The system validates the payment result |
|  |  | Step 4. The system makes the advertising message visible to user. |
|  | Step 5. The business people is being notified with the payment result | |
|  |  |  |
|  |  |  |
| Alternate Courses: | Step 1. The business people provides payment information | Step 2. The system redirects business people to the online payment system for payment. |
|  |  | Step 3. The system validates the payment result (fail) |
|  |  | Step 4. The system makes the advertising message remain hidden to user. |
|  | Step 5. The business people is being notified with the payment result | |
|  |  |  |
|  |  |  |
| Conclusion: | The use case concludes when the business people is being notified with the payment result | |
| Postcondition: | The advertise message visible to user. | |
| Business rules: |  |  |
| Implementation constraints and specifications: |  |  |
| Assumptions: |  |  |
| Priority: | Low |  |
| Open issues: | How long should such advertising message visible to user and how should it being paid | |

1. THOMPSON, SHARON H., and ERIC LOUGHEED. "Frazzled By Facebook? An Exploratory Study Of Gender Differences In Social Network Communication Among Undergraduate Men And Women." College Student Journal 46.1 (2012): 88-98. Academic Search Premier. Web. 1 June 2012. [↑](#footnote-ref-1)
2. http://www.psychologytoday.com/blog/brain-wise/201209/why-were-all-addicted-texts-twitter-and-google [↑](#endnote-ref-1)