Re-engineering and Performance Optimization of University Cyber AgenT (UniCAT) System

BY

SHIEH YU XUAN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONS)
INFORMATION SYSTEMS ENGINEERING

Faculty of Information and Communication Technology

(Perak Campus)

JAN 2014

DECLARATION OF ORIGINALITY

I declare that this proposal entitled “**RE-ENGINEERING AND PERFORMANCE OPTIMIZATION OF UNIVERSITY CYBER AGENT (UNICAT) SYSTEM**” is my own word except as cited in the references. The proposal has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature :___________________

Name : SHIEH YU XUAN

Date : 7 APRIL 2014

ACKNOWLEDGEMENTS

I would like to extend my utmost appreciation to my supervisor, Dr. Cheng Wai Kheun and co-supervisor, Mr.Illeladewa who has given me this promising opportunity to participate in making this project a success. Besides, patience guidance from my supervisor worth remark alongside with the success of this project, million thanks to you.

Nonetheless, to the all the group members of this project who always express the positive attitude and motivation continuously, never reject the relentless meeting for integration discussion and feasibility review meeting.

No to be missed, thanks Lye Guang Xing, a close friend of mine in providing the graphics images in bringing this software to the next level. Your untiring support on this project shall be noted here and my heart forever.

Finally, thank god for giving me strength in overcoming all the obstacle before me and also thank god for giving me the opportunity to meet with obstacles that act as a training ground for a better me.

ABSTRACT

University Cyber AgenT (UniCAT) System is a graphical social networking site which encompasses the multi-agent approach. The environment of the system is simulating the real world situation of UTAR Perak Campus and its population, which also implies that the automated behaviors of agents within the 2D map are actively projecting the actual preferences of the users. In addition, the system comprises several functionalities of social networking which interact with user in a graphical fashion.

This project is to reengineer and optimize the structure and features of the UniCAT system. In specific, the plan of this project is to tackle the actual or potential flaw of the current system and integrate the dismantled modules into one centralized UI. Lastly, bundle the modules into complete system and package it for user installation.

The author has studied the existing UniCAT system and discovered blemishes of which several modules' user interface are not consolidated. It brought severe impact to the user experience, which will be explicated later. Besides, considering that this program might be used by numerous users concurrently, limitation which may contribute inefficiency to the complete system has been discovered.

To tackle the weaknesses, similar software which are incorporating the multi-agent system or social networking element have been studied. By going through research and detailed study on the existing system, a fascinating solution which advocates greater user experience and usability has been produced and will be explicated soon.

In a nutshell, the interesting part of this project is to transform the current concept-driven prototype to a completely publishable desktop application. It involves substantial reengineering of the program structure, redefine the way of use and finally combining both

multi-agent system and social networking components into a highly interactive graphical social networking system.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
BDI	Belief, Desire, Intention
CPU	Central Processing Unit
I/O	Input/Output
LAN	Local Area Network
POI	Point Of Interest
UI	User Interface
UniCAT	University Cyber Agent
SNS	Social Network Services

(This page is leave blank)

CHAPTER 1: INTRODUCTION

1.1 Problem Statement

In revision of the existing University Cyber AgenT (UniCAT) System, the problem is that the social networking modules are not designed to be integrated with the existing 2D map in unified user interface. Moreover, every social networking module was developed separately in user interface and it was meant to degrade user experience and system usability. Further, the current prototype system is not optimized to be readily expandable in terms of functionality because there is no central hub for new component to join the system.

This project has to be created mainly because the current system is not up to a standard where it is publishable. The existing partially server-based system is adopting the java applet technology where synchronization among substantial clients has been a great challenge which it needs to be reengineered in this project. In addition, reengineering of the applet system will further improve the performance.

1.2 Project Scope

The project scope is separated into 2 parts which are re-engineering and enhancement. Reengineering refers to development of a main user interface as a central user interface hub for all the existing components, compatibility issue is expected to be resolved during the integration process. Besides, reengineering also transforms the existing java applet system to a standalone desktop application system, therefore the load of machine will be migrated from server to the client computer.

The enhancement refers to improvement of the existing modules in terms of functionality and usability. Besides, the known limitation of the current system will be countered in order to lift the usability. By the end of the enhancement process, the basic concept of the module will remain, however the way of using it and the comprehensiveness will be upgraded.

However, this project does not include the configuration and the optimization of server bandwidth limitation which may essentially degrade the user experience in the client program.

1.3 Objective

This project focuses on **developing a central user interface** which consolidate the existing 2D map and all the social networking modules. Then, all the user interaction will take place in the one and only full screen window, there is no switching of windows by triggering any social networking module.

In specific, all the **social networking modules should be able to display its graphical component alongside with the 2D map in the same windows** once it has been triggered, however with the limited space in one screen, the system will automatically react to user focus and only show necessary information to retain simplicity. Besides, user should now be able to interact with the existing 2D map and the agent with social networking modules.

Another focus of the project is **to transform the applet based system to a completely standalone desktop application**. Partial database and multi-agent movement engine will be migrated from the server to the client computer to reduce server workload. After the transformation, an installation package of the system will be deployed to the UniCAT webpage for client beta testing.

In precise, the **non-time-critical data will adopt server-client data synchronization method**. Which also means those data will be stored in the client side database. At the same time, client database will synchronize with the central server database periodically. This implementation is intended to improve the system's internal behavior and without compromising the external behavior.

Not to neglect, the **functionality of the social networking modules will be upgraded to the publishable standard**, this including adding new functions to complement the existing one. Not only that, to ensure the completeness of the integration process, those SNS processes that need to be done on the browser will be integrated into the client program.

Eventually, executable file, user libraries and the client database will be **packaged into a standalone installation file** and ready to be deployed online for user download.

1.4 Contributions

By the end of this project, an undesirable prototype will be reengineered and results in a publishable application. The reengineering process will have the following impact:

- With all the system components being gathered and consolidated, the client program can be perceived as a complete standalone desktop application.
- Future expansion of the functionality has been made easier, unified component structure can be reused and readily plug into the newly developed central hub.
- The performance of the server will be enhanced by fully eliminating the applet technology on the client program.
- The complete client program will be packaged in an installation program which allow user to do configuration free installation on a fresh personal computer.
- The comprehensiveness of social network services is amplified and expect to attract more users for greater functionality

Aside of the low level impact, there are also society impact:

- Potential student is now able to play the game and familiar with the environment before coming to a whole new study environment.
- The players will have a consistent platform to share and acquire local information among each other.
- The preference of the local user can be gathered as a precious statistics data which is very favorable to the marketer, scientist, entrepreneur and government.
- The measurement of cultural difference across border is made possible with the implementation of this system for different places.

1.5 Background Information

University-Cyber Agent System (UniCAT), a system that created for the purpose of studying the behaviour of population in UTAR Kampar. The system applied client-server architecture, on the client sides, web-based interface was provided for the log in function of user and also application-based interface for the user to play and interact with the system. It has a virtual map of Eastlake Kampar, and each of the users is represented by an agents. The agents will reflect the user's behaviour based on the information provided by users.

UniCAT was then extended to have social network services including chat system and file sharing system. These functions were implemented for the ease of freshman from UTAR to exchange information in terms of studies, foods and so on. They are able to share the information of events that are happening in Eastlake. In other word, they do not need to come to Kampar all the way just to find a staying place before the semester began. Any important information, they can be informed through the system, since all of them are connected in the virtual world.

In this project, there are several key technologies that drive the whole system, the apparent one would be the multi agent technology (Multi-agent system, 2013) that uses the Jadex library (Active Components, 2012). It enables the agent to simulate the real user behavior by adopting the Belief, Desire, Intentions (BDI) software model (Belief–desire–intention software model, 2013). Belief is the agent’s belief about the world, Desire is the objective of the agent and Intentions is what the Agent has chosen to do.

CHAPTER 2: LITERATURE REVIEW

2.1 Review Description

A comprehensive program is always stand up of huge amount of features and functions. With the rapid expanding software environment, how the features can be decently presented to the users became a serious challenge to the developers. They are cracking their head to squeeze more and more features into a program and yet, needless to harm the simplicity of the interface which may eventually degrade the user experience. In the following analysis, the author will focus on how do the existing systems in the market use the sweet point between interface simplicity and broad functions.

2.2 Existing System

2.2.1 WeeWorld

The first analyzed system is a web-based social network game, WeeWorld (WeeWorld, 2013). This game encompasses the concept of socializing in virtual world. Player is represented by an avatar and explore freely from one world (place) to another. In this game, user is able to make friends, join event, chat and participate in other socializing functions.

Firstly, user has to sign up and customize his own avatar before joining the world with the others. In every world, there are certain interesting events for player to participate, meanwhile a player may get to know other players through interacting with their avatar. In addition to that, a player may want to walk through the world without joining the event, they may chat on a tree house, balcony and literally any place.

The reason why this game was chosen is that behind the identical features in social networking, it has the similar concept of connecting people with UniCAT. In the UniCAT, multi-agent has been used to represent the real world player and able to think and act on behalf according to their personal preferences. In WeeWorld, avatar movement is controlled by the user, however the behavior of both systems are theoretically homogeneous if it origins from the same user. An example may explain where in UniCAT, if a player prefers to relax in playground, the game agent will always move to playground when its energy is low. Likewise, player will move the avatar to the playground world when he wants to relax. This is a fantastic social networking strategy of connecting people with same interest on the map.

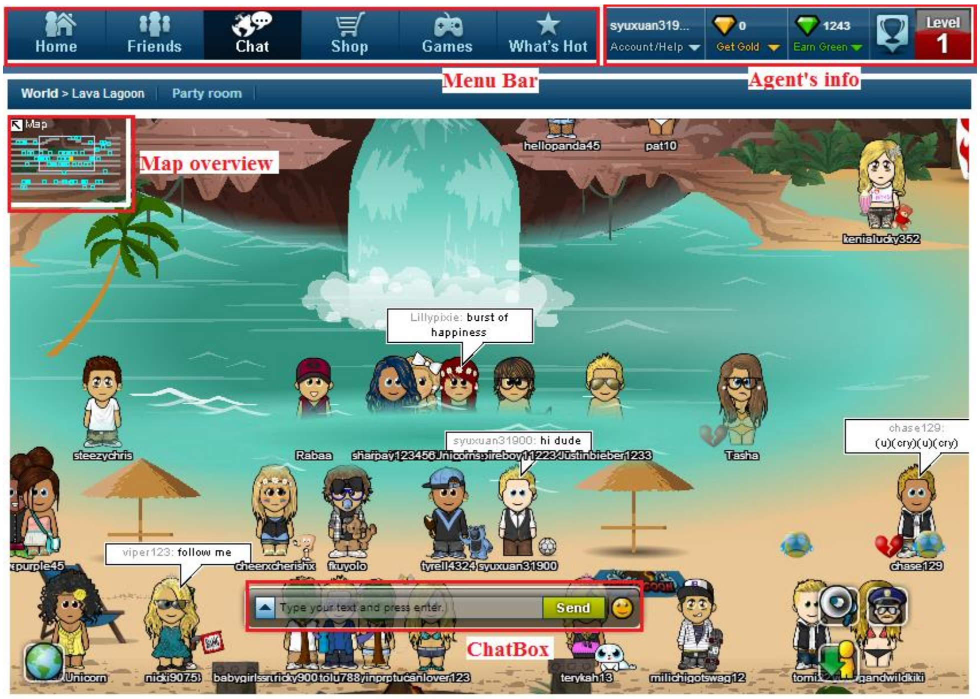


Figure 2-2-1-1 WeeWorld gameplay UI

Figure 2-2-1-1 shows the situation of multiple players socializing through their avatar. There is a menu bar at the top of the interface and it uninterruptedly appears in every situation. The other components are optionally displayed depending on the nature of the participated world.

Component on The Map (Avatar name, chat pop-up, chat box, button)

In this game, what good about the component design is that there are several frequently used components like chat box, essential buttons and map overview are easily accessible from the map, the semi opaque design also prevents player from losing context on the map and avatar itself.

On the other side, the agent name is displaying below the avatar disregard of the character length. It creates a visibility problem particularly when several avatars are placed together and the avatar name is overlap among one another, since there is no option to hide the name, it generates confusion for player to identify others name. The same situation goes to the chat pop-up given the situation 2 nearby avatars are talking at the same time.

Menu Bar and Navigation

In this game, everything can be accessed from the menu bar, there is no occasion where menu bar is hidden. Thus, everything is literally one click away, it delivers great user-friendliness to the user experience. Navigation wise, the developer has given a great deal of concern on this area, minimal user intervention is required from one world to another. For instance, if a player wants to teleport from home screen to the gym world, what he needs is point to the ‘Chat’ and select ‘World” from the drop down list, then the list of world will appear as options.

However, the placing of the menu bar which is at the outside top of the map will create issue for this browser based game particularly with the contemporary practice of using widescreen display. The issue emerges when the player wants to see both the agent info and the map simultaneously, because of the limited height of a browser body, player may sacrifice either agent info or partial map. This issue can greatly reduce the satisfaction of what others just brought.

Social Networking Functions

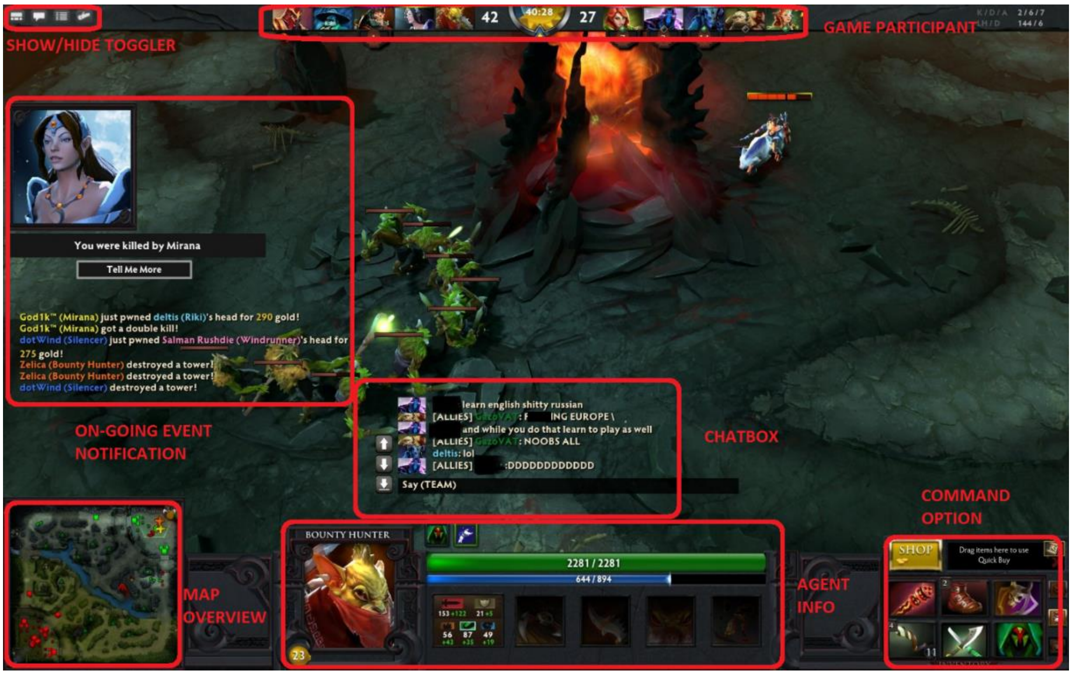
This game comprises rich features of social networking, it encourages player to broaden his/her friends network by offering game credit as a reward for every friend gained. Moreover, it allows player to expand the game network by penetrating the news through the social networking site friends such as Facebook. The most awesome feature is that it emphasizes on bring people together by common interest, the most activities occurred in the game is men workout and socialize in gym world and ladies are shopping and chatting at the shopping center.

However, despite it has connection with the Facebook, there is no friend suggestion available from the area. It is a duel where it encourage the player to connect new friend but on the other hand lose track of friends which are already connected in the real world. This problem however is an advantage for those player who is looking for a breakthrough in the friend network and avoid duplication of friends linking in different social networking mechanism.

2.2.2 Defense of the Ancient

The second analyzed system is the Defense of The Ancient (DotA) (Valve Corporation, 2013). This game has some features which are very much the same with the existing UniCAT system, such as multiplayer architecture, agent delegation, chatting, live event notification and map. Although the agent is actively controlled by the user, however there is still artificial intelligence concept in the agent, deciding which route to go and when to engage the enemy.

Firstly, when the game began, users have to choose which agent to represent them, different agent has different characteristics which is preloaded into the agent by default. Then, the users can make use of the unique character of each agent to assault the enemy. Meantime, they can chat with teammate or to all the players freely. Besides, the game system will automatically generate and display live event notification to the concerned player.



You were killed by Mirana

God1k™ (Mirana) just pwned deltis (Riki)'s head for 290 gold!
God1k™ (Mirana) got a double kill!
dotWind (Silencer) just pwned Salman Rushdie (Windrunner)'s head for 275 gold!
Zelica (Bounty Hunter) destroyed a tower!
Zelica (Bounty Hunter) destroyed a tower!
dotWind (Silencer) destroyed a tower!

[redacted] learn english shitty russian
[ALLIES] GazoVAT: F[redacted]ING EUROPE \
[redacted] and while you do that learn to play as well
[ALLIES] GazoVAT: NOOBS ALL
deltis: lol
[ALLIES] [redacted] :DDDDDDDDDDDD
Say (TEAM)

Figure 2-2-2-1 DotA gameplay UI

The figure 2-2-2-1 shows the DotA main UI during gameplay. This screen shows 6 important functions with 1 visibility toggle button. The toggle button is to let user to decide which function to show and which not.

Chat box & Live Event Notification

What good about this design is that the information appeared on the screen can easily grab user’s attention and at the same time the transparent background does not obstruct the user’s sight on the map. Besides, another advantage is that the information will fade in and out automatically without the need for human intervention, even the chat box can be triggered by just pressing ENTER.

However, considering the expansion of user base and UniCAT functionalities, this design would contain a huge flaw when the notification come in great amount, the map would be obstructed by the stacked up notifications and eventually because of the automatic fade out feect, user might not able to read all the notifications or messages before they fade out.

Main Controller (Agent Info, map overview, command option)

What good in this design is that the most used control and information is gathered in a fixed location which is the bottom of the screen. The brilliant design allows the map to have 80% of the full screen. Besides, the agent info will change according to the highlighted agent, so that the map will not be flooded with agent’s info, this is a very useful alternative to UniCAT when there are vast amount of users on the map.

However, this design only brilliant when there is little information to show. Due to its little allocated space, the big system like UniCAT cannot squeeze all its control button and information under this compartment.

2.2.3 Second Life

The third analyzed system is the Second Life (Second Life, 2013). This is a graphics intensive social networking game which features virtual reality, the environment in the game is simulating the real world environment of certain cities. It gives an outstanding experience for users to immerse themselves in the world where everything looks so familiar. What is Second Life? Second Life described that, “It is a place to connect, to shop, to work, to love, to explore, and a place to be different be yourself, free yourself, free your mind, change your mind, change your look, love your look, love your life” (The Nerdfiles.com, n.d.).

This game was chosen because it is greatly simulating the real world situation where players are able to conduct some activities among each other, for example they can trade their property, visit somebody house and hold a house party. Further, this is not a socializing game that only entertains the player, in fact there are some features that practically bring people together for constructive task. For example, players are able to do create business venture in the game and receive virtual money for all the transactions which it can be cash out as real money later on. This is an exceptional idea to attract people to come together to do something just like real world.

First, to play the game, user needs to sign up and customize the avatar before it can be deployed into the virtual world. In the game, players from all around the world are engaged in activities that one can thought of in real world. They can do fishing, chatting, meeting, dancing and many more which they may not able to do it together in real world. Besides, user can simply click on one player to view his/her details and get to know each other by using voice chat, text chat or gesture communication. Moreover, player is able to switch to place of different predefined cultures such as Rome, Dublin and Tokyo, then they are able to socialize with people from different culture in such a short period.

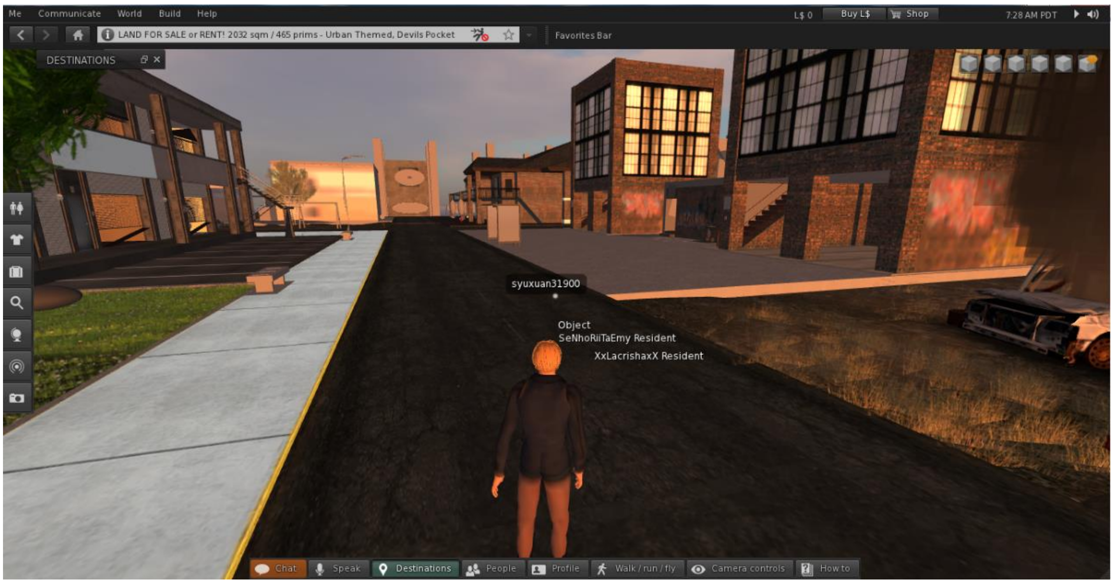


Figure 2-2-3-1 Second Life gameplay UI

Text Information on the Map

The UI design for this game is decent in terms of simplicity. All the notification is displayed on demand, thus there is no special notification compartment that may obstruct the gameplay map. Once there is an event of notification or description, the text messages will appear directly on the screen however there is no predefined location for that, it will take any place which is unoccupied.

This however creates a flaw to the system which may confuse the player. Since there is no consistent place to group certain kind of text information, it requires user to pay extra attention to read the text to figure out what is it about. This comes apparent when players are required to read the notification that they will normally neglect.

Navigation / Control Bar

The navigation and control bar is concise that it place itself at the left and bottom of the screen. The bar is so simple and unappealing that it allows user to focus on the map and at the same time provides great functionality on navigation throughout the system. Besides, all the socializing UI is collapsed and can be easily expand from the control bar, this design is suitable for all kind of user which emphasize on different function.

Social Networking Functions

The selling point of this game is its practical yet entertaining activities, the main focus of these activities is meant to bring people together to do what they are unable do in the real life. Doing business online is a great idea, however doing business online with friends network is a fantastic idea. Business is believed to expand further with a sense of bond and trust among friends, moreover with the rapid expansion of friendship across national border, advertising effect is enormous via friend recommendation.

However, this kind of activity contains the possibility of fraudulent transaction which requires heavy monitoring from the system moderator. Very likely, the workload of moderator is overwhelming and mishap may happen anytime. In the event of insufficient moderation power, it is a potential drawback that might damage the reputation of its own.

2.2.4 The Sims FreePlay

The fourth analyzed system is The Sims FreePlay (Electronic Arts, 2013). This is an Android game app which features social networking functions. This game includes not only the activity but the habit of the human, it allows the sims (avatar) to rear a pet and express the feelings to one another. This game is not a real time game but user is mostly interacting with the virtual sims which are controlled by the AI, however it is able to connect real user together in some event and score will be ranked on how well the player can socialize with the virtual sims.

This game was chosen mainly because of the server-client software architecture that is similar to the new direction of the UniCAT. Besides, the avatar has a similar needs attribute with the existing UniCAT system such as the status of hygienic, hungriness and strength. It is important to review how other system tackles the synchronization problem with huge number of clients. For instance, what feature should be compromised in order to ensure the overall wellness.

In this game, player as usual can create an avatar of his own. The avatar is simulating a real world behavior of a human which requires sleep, eat and social needs. By achieving a game predefined objectives, the game level will be elevated and player is able to create more and more virtual sims as his/her neighbor. With higher game level, player sims is allowed to use more interaction method such as hug, joking and so on to increase the social score. In this game, the virtual sims will walk around freely and do what they like such as whistling without player intervention.

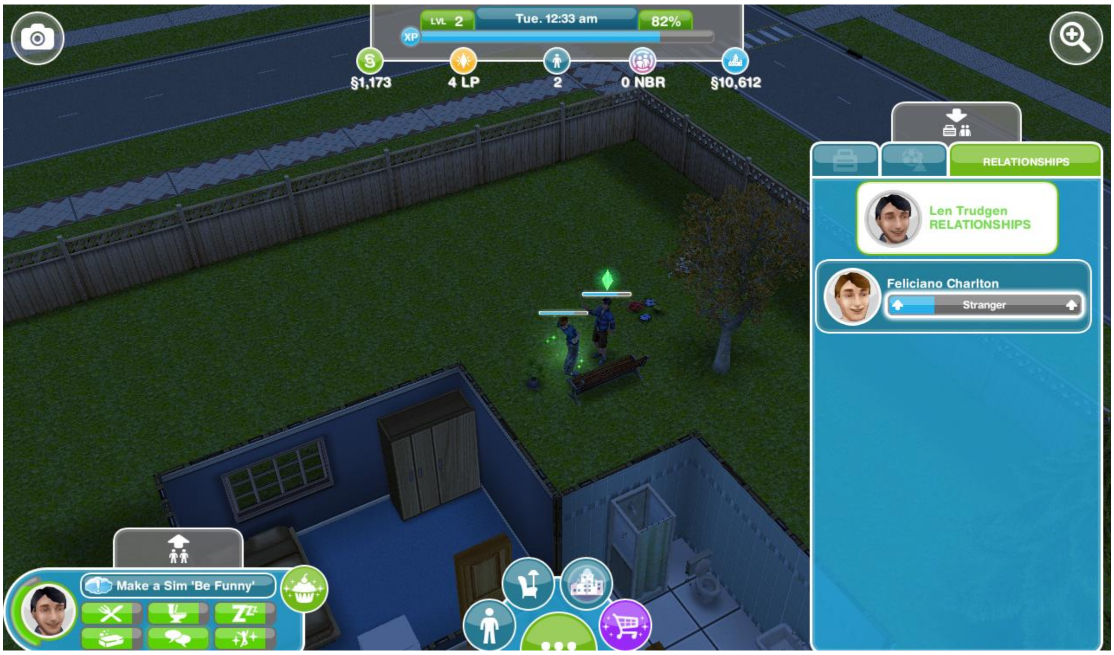


Figure 2-2-4-1 The Sims FreePlay gameplay UI

User Interface

The user interface is decent that it is very consistent, the important information container is always float on the edge of the map which doesn't obstruct the gameplay. For complex information, player is allowed to expand and collapse the container as necessary. Besides, to ensure the simplicity of the UI, all the function buttons are composed with solely image.

However, to navigate to a function such as shopping, there is a transition of the full screen map to the functions UI. It is apparently inappropriate especially when the player is frequently navigate throughout the functions, he/she will lose context of what has been done previously and what is running in the background.

Software Design

This game is adopting the server-client architecture which the multi-agent engine and calculation of the game is done at the client device. The role of the server is a central database which stores the player data and game related data, the game client will periodically exchange data such as the shopping item and player score with the server. This architecture allows huge amount of users to play the game at the same time without greatly compromising the performance of the server because the major workload is done at the client side.

This however will create a problem, it is not possible to have a real time synchronized gameplay across the server. For instance, if the same player log in the same account in different devices, the automated behavior of the virtual sims between 2 devices are different. It appears to be the drawback of this architecture and there is no way to effectively solve this problem permanently as real time synchronization among sophisticated game requires high requirement on the hardware such as network bandwidth, server power and client device.

2.2.5 Other Systems

There are many similar software or game in the market which serve the same purpose with the UniCAT. There are some other software that have been reviewed but not stated above, it is because they are not the best candidate to do comparison with the existing UniCAT system, and it may be due to software design, user interface or purpose that is unlike to the direction of the UniCAT. Below are the software list:

- Farmville 2 (Zynga 2013)
- Habbo Hotel (Sulake 2013)
- Meez (Donnerwood Media 2012)

2.2 Critical Remarks

2.2.1 System Comparison

Criteria	WeeWorld	DotA	Second Life	The Sims FreePlay	New UniCAT
Application type	Web application	Desktop application	Desktop application	Mobile application	Desktop application
Number of agent on the map	1-50	1-10	Unlimited	1-20	1-25
Real time agent synchronization	Yes	Yes	Yes	No	No
Event	Virtual	No event service	Partially real world	Virtual	Real world
Friend management	Yes	No	Yes	No	Yes
Private chat room	No	Yes	Yes	No	Yes
User interface transition	High	None	Low	High	Low

Table 2-2-1 System comparison table

2.2.2 Comparison Elucidation

Application type

WeeWorld is a low computational game which the avatar movement is controlled manually. Thus, it is possible for the server to handle the storage and computational power. However for UniCAT, it is a high computational game which every agent behavior on the map is automated and it requires a great deal of computational power, hence the workload has to be segregated from server to the client computer. Besides, desktop application allows data retrieval from the local drive which practically increases the performance of the software

Real time agent synchronization

Real time agent synchronization implies that the bandwidth and hardware requirement is high. However, it is required when the user is controlling his avatar and meet with the others in the game, so that users can engage with each other intentionally. In the UniCAT, avatar movement is automated and is determined by the system according to the registered user preferences. By adopting this concept in the client game engine, there is no need to synchronize agent movement across the clients, this is because the same game engine for all clients will work based on the same user preferences.

Event

In this criterion, event is divided into real world and virtual type. The Sims FreePlay features virtual event which is predefined during the software development or is created by the game admin. The player is able to join the event and socialize with other player who has the same interest. However for the UniCAT, it features real world event which is the other way around, player can create an real world event on the map which contain the event details, what differ is it gathers people with common interest physically.

Friend management

To link people together, friend management is a vital module, therefore it appears in most of the online game which emphasize on players interaction. Always, it serves as a constraint to the social networking services, such as chat service only available to friends. In the new UniCAT system, a comprehensive friend management module is included, player will be able to categorize friends into different group.

Private chat room

Majority of the game has a comprehensive friend management but has neglected the group chatting service. In WeeWorld, the chat message appear on the map and visible to all players, it has neglected the importance of private communication. . In the new UniCAT, player can create a private chat room and invite friends to join the conversation, what differs is that the messages are only visible to the intended recipient.

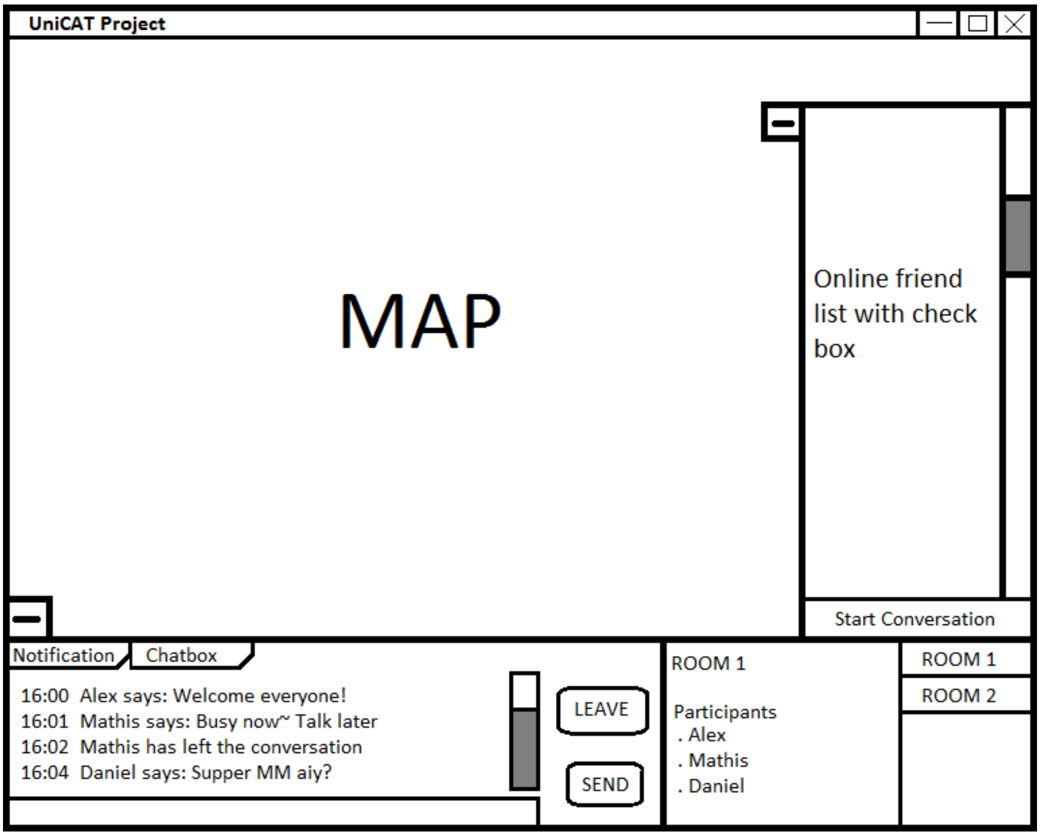
User interface transition

This criterion assess how often the full screen user interface has to transit to another by using different services. It is always used as a trade off in return of greater details of displaying information. An example will be The Sims FreePlay, the player cannot has both map and shopping UI altogether, instead the shopping UI will replace the map UI once it has been triggered. Frequent swapping of the whole UI may annoys the user. In the new UniCAT, UI transition is restricted to pop-up UI which partial background progress is still visible to the player, it creates a sense to the player saying that "Any extra services you open will not affect the current game progress".

2.4 Proposed solution

There is a common problem shared by the existing system which is too much information and control to fit in one screen. Hence, the new solution to this limitation is the "Mode UI". Mode UI emphasizes the non-transitional user interface and minimal UI effect on each service change, which means that the same user interface and the same compartment may display different information under different modes. In the following, the author has used 2 samples of the new user interface to explain how the solution really works.

2.4.1 UniCAT Chat Mode



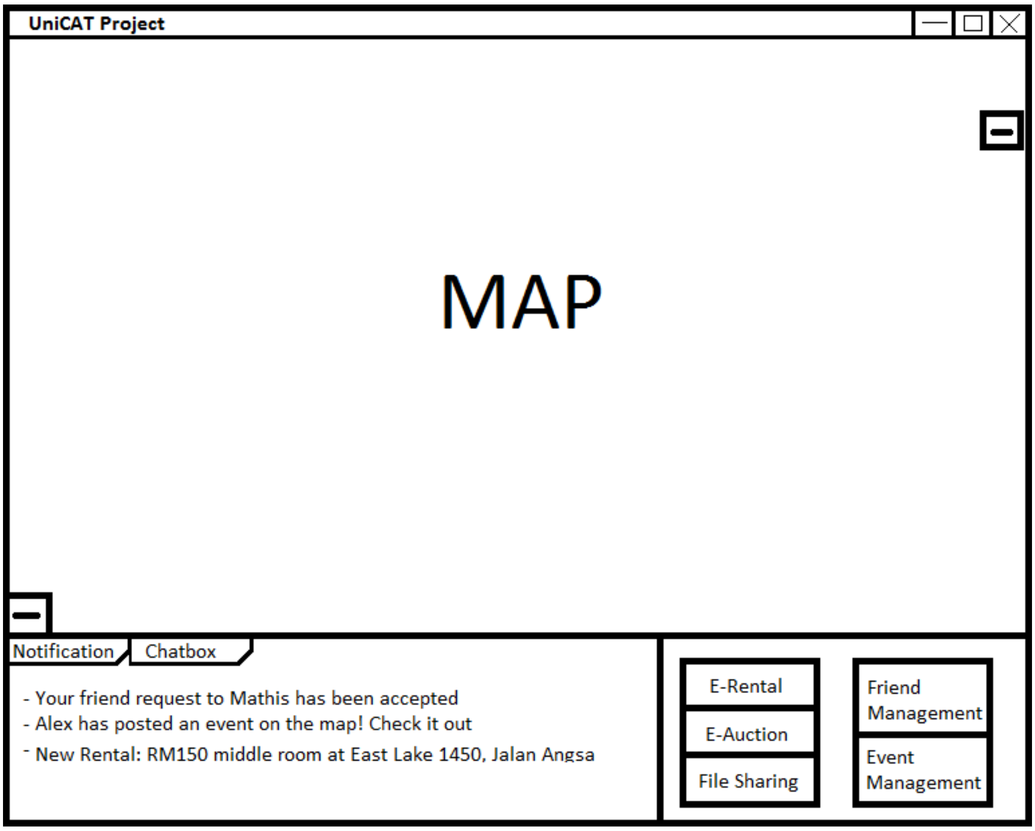
16:00 Alex says: Welcome everyone!
16:01 Mathis says: Busy now~ Talk later
16:02 Mathis has left the conversation
16:04 Daniel says: Supper MM aiy?

Participants
. Alex
. Mathis
. Daniel

Figure 2-4-1-1 UniCAT Chat Mode

This mode is activated by clicking the “Start Conversation” in the friend list or by clicking “Chat box” in the bottom left compartment. The online friend list is by default collapsed and only shows when chat mode is activated or manually clicking the show button attached with it. Besides, there are 2 functions in the multi-purpose compartment, first is the list of conversation participant, second is a list of conversation room which allow user to choose from. In addition, user can leave the room with the leave button.

2.4.2 UniCAT Normal-Notification Mode



- Your friend request to Mathis has been accepted
- Alex has posted an event on the map! Check it out
- New Rental: RM150 middle room at East Lake 1450, Jalan Angsa

Figure 2-4-2-1 UniCAT Normal-Notification Mode

This is the default mode after the UniCAT system has been initialized. There are live notifications in the bottom left compartment which covered notifications from friend request, map event and file-sharing. This mode can be activated by clicking the notification tab.

CHAPTER 3: PROJECT DESIGN

3.1 Project Methodology

This project development will adopt the **Evolutionary Prototyping** methodology (Teach-ICT.com Limited, n.d.). This reason is that the UniCAT system is going after a broad span of university students, therefore involve user in development will increase the user acceptance of the final system. Besides, there are always scenarios which vary among substantial amount of users that have been neglected by the developer, hence error can be detected earlier with user intervention. Further, the imagination-driven user interface requirement is ambiguous to the developer and subject to a developer's creativity, thus a prototype rating is important for further development.

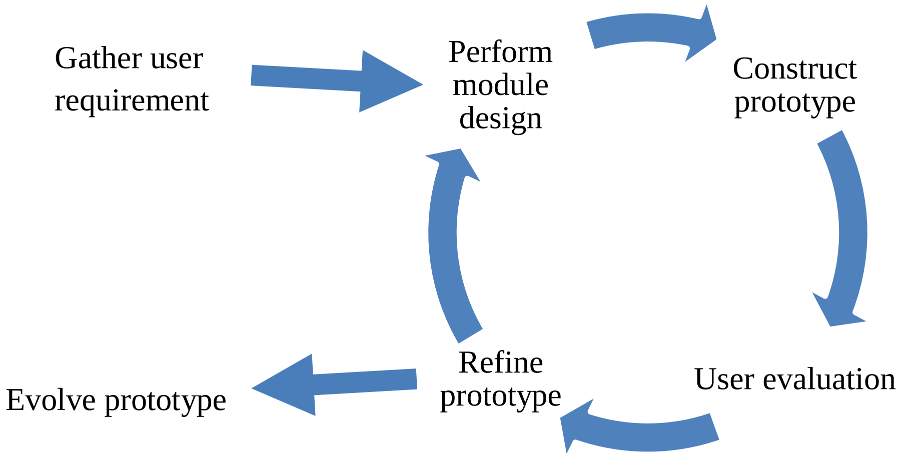


Figure 3-1-1 Flow chart of the Evolutionary Prototyping Methodology

3.1.1 General Work Procedures

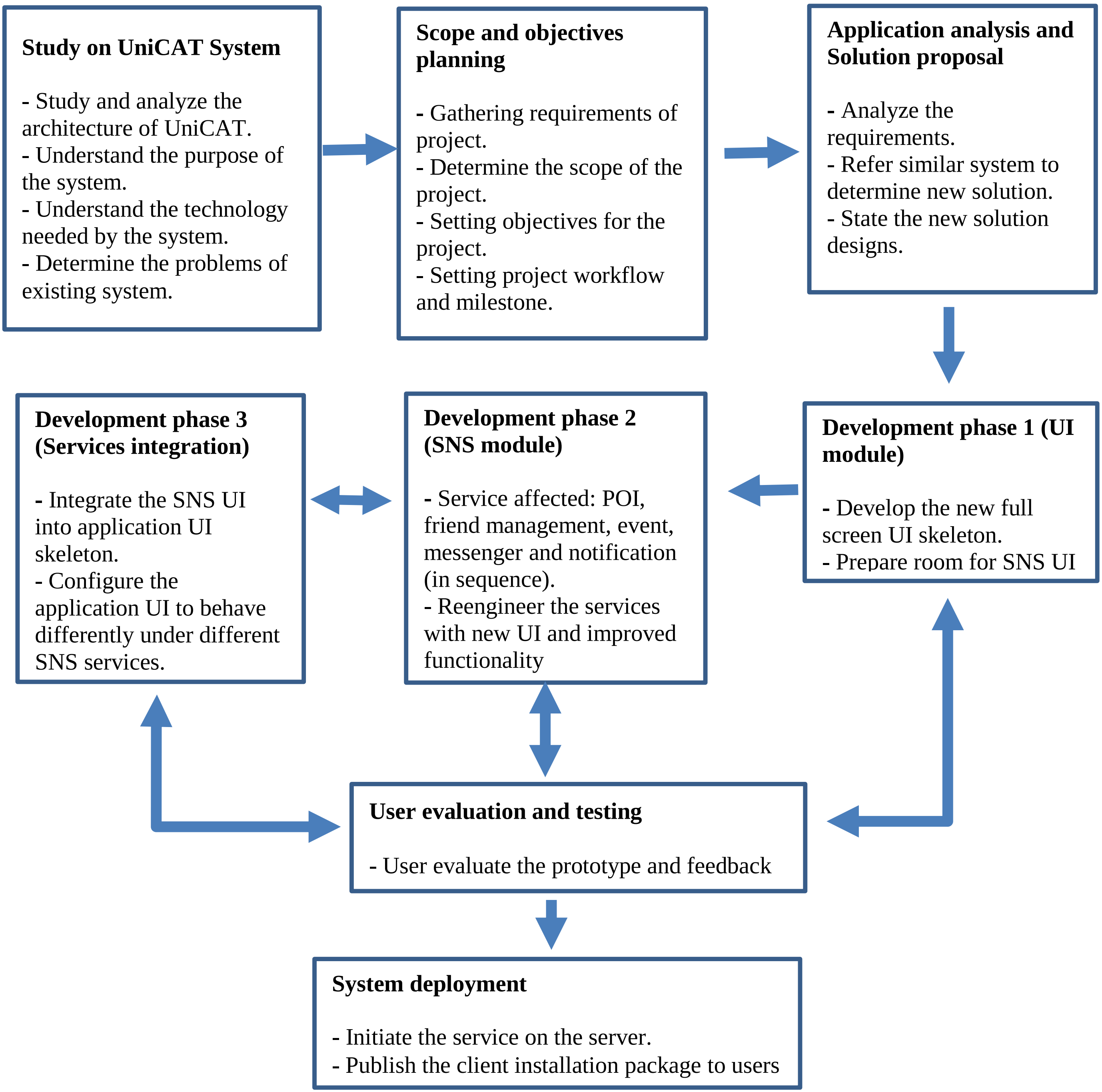

Study on UniCAT System
- Study and analyze the architecture of UniCAT.
- Understand the purpose of the system.
- Understand the technology needed by the system.
- Determine the problems of existing system.
Scope and objectives planning
- Gathering requirements of project.
- Determine the scope of the project.
- Setting objectives for the project.
- Setting project workflow and milestone.
Application analysis and Solution proposal
- Analyze the requirements.
- Refer similar system to determine new solution.
- State the new solution designs.
Development phase 1 (UI module)
- Develop the new full screen UI skeleton.
- Prepare room for SNS UI
Development phase 2 (SNS module)
- Service affected: POI, friend management, event, messenger and notification (in sequence).
- Reengineer the services with new UI and improved functionality
Development phase 3 (Services integration)
- Integrate the SNS UI into application UI skeleton.
- Configure the application UI to behave differently under different SNS services.
User evaluation and testing
- User evaluate the prototype and feedback
System deployment
- Initiate the service on the server.
- Publish the client installation package to users

Figure 3-1-1-1 Workflow for Evolutionary Prototyping

3.1.2 Work Procedure Elaboration

Before the initial planning stage, the author has done some research on the current version of UniCAT system. It includes the understanding of server-client architecture, the project history and the contribution of the previous developer. After the author has a clear picture of the system architecture, he started to learn the technologies involved, such as multi-agent approach, BDI model and eventually familiar with all the programming language involved such as Java, XML and JaDEX.

In the planning stage, the author started to discover the problems in the UniCAT system and construct the problem statement. After that, based on the research and findings, the project scope can be set now. The scope of the project is now narrowed down to only enhance the existing features of social networking such as chat function and live notification. In addition, the scope also includes the reengineer of main user interface and integration of social networking module. Next, the objectives and expected contribution can be derived from the project scope.

In the analysis and design phase, several existing systems have been chosen to be reviewed and the choosing criterion is based on the project scope. In the review process, the strengths and limitations of every system are recorded and analyzed. Then, the findings will be tabulated in a comparison table and description. Eventually, the new solution is proposed with the consideration of the comparison findings.

In the development stage, there are 3 phases. The first phase is to develop a full screen application UI skeleton with ready compartment for social network services UI. The second phase is the SNS development, it includes the development of new function, new user interface and tackle the existing functionality limitation. The third stage is to integrate the reengineered SNS module into the application UI and make necessary modification to ensure the information is displayed correctly and able to work with other SNS. Second and third phase will loop until all the SNS is reengineered.

At the end of every development phase, the prototype will be delivered to the user to evaluate and acquire feedback. If the user requires further refinement, then the prototype will be returned to the previous phase and make necessary changes, this continuous loop will go until the end of third development phase

In the system deployment stage, a client installation package is built and distributed to the users to install. Meanwhile, the service of the server is initiated and the feedback from users are actively recorded for maintenance purpose.

3.2 Implementation Issues and Challenges

The biggest challenge in the implementation phase is the memory constraint. By adopting the new solution, massive memory is required for several modules to work together. For example, add listener object to each agent for user to trigger a friend management function will be expensive, considering there are a great deal of agents on the map. Besides, to ensure a more responsive user experience, data has to be preloaded into the program rather than retrieve database on demand, it is a memory hungry mechanism as the data is increasing over time.

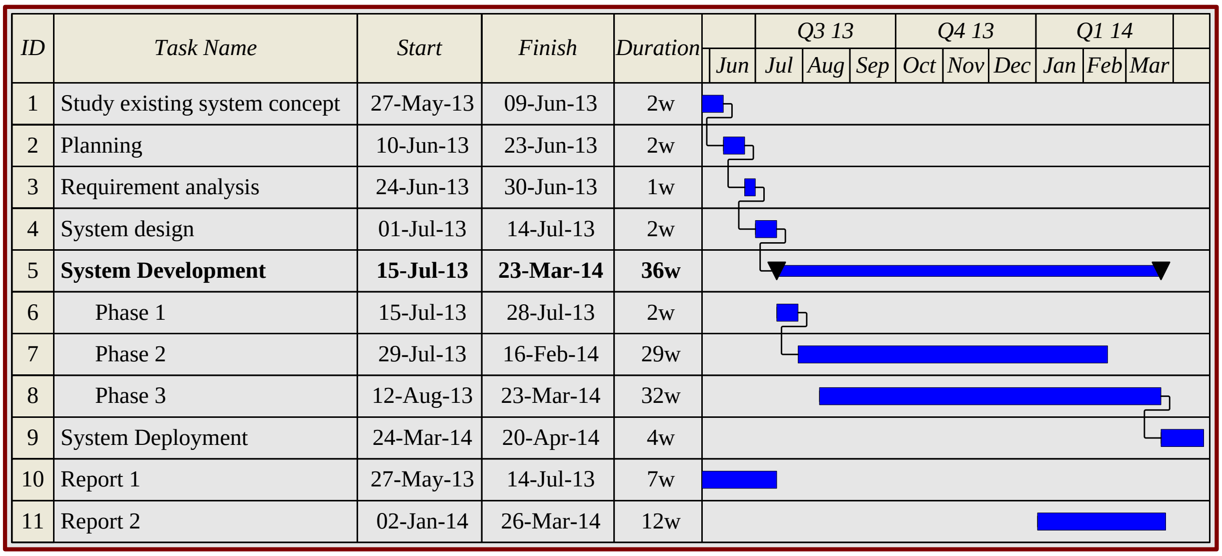
Next, another challenge is that when all the modules are integrated among each other, the screen space will become a scare resource. A smart mechanism is required to determine which module should be focused and how to cater information from other module simultaneously. The main idea for this is to prevent the user to lose context on the on-going as well as the background process. The tricky part is that this mechanism should be made dynamic so that it can accommodate the new module that will join later in the future.

3.3 Project Timeline

This project is scheduled to be finished in one university academic year which is approximately 47 weeks. There are 2 reports submission in the first and last trimester and then viva session is conducted soon after every submission. In the first viva demonstration, a prototype of the proposed solution should be ready to support the solution stand point and affirm the direction.

The second and also the final viva session is to demonstrate the completed new UniCAT system. By that time, the test validation and objectives checklist should be ready to be reviewed. After that, the deliverable of this project will be formally published to the public to use.

3.3.1 Gantt Chart



ID	*Task Name*	*Start*	*Finish*	*Duration*
1	Study existing system concept	27-May-13	09-Jun-13	2w
2	Planning	10-Jun-13	23-Jun-13	2w
3	Requirement analysis	24-Jun-13	30-Jun-13	1w
4	System design	01-Jul-13	14-Jul-13	2w
5	**System Development**	**15-Jul-13**	**23-Mar-14**	**36w**
6	Phase 1	15-Jul-13	28-Jul-13	2w
7	Phase 2	29-Jul-13	16-Feb-14	29w
8	Phase 3	12-Aug-13	23-Mar-14	32w
9	System Deployment	24-Mar-14	20-Apr-14	4w
10	Report 1	27-May-13	14-Jul-13	7w
11	Report 2	02-Jan-14	26-Mar-14	12w

Figure 3-3-1-1 Gantt Chart

3.3.2 Task Breakdown

Report 1	
Content	Duration (week)
Problem statement	0.5
Project scope, objectives	0.5
Contribution	0.5
Background information	0.5
Literature review	1
Critical remarks	1
Methodology and tools	1
User requirements	1
System design	0.5
Conclusion	0.5

Table 3-3-2-1 Report 1 schedule

Report 2	
Content	Duration (week)
Problem statement	1
Project scope, objectives	
Contribution	
Background information	
Literature review	
Critical remarks	
Methodology and tools	1
User requirements	3
System design	1
Implementation & Testing	3
Project review	2
Future work	1

Table 3-3-2-2 Report 2 schedule

System development	
Content	Duration (week)
Phase 1	
Central user interface	2
Phase 2	
Point of interest	5
Friend management	7
Event	5
Messenger	7
User Profile	5
Phase 3	
Unit integration	25
System integration	7

Table 3-3-2-3 System Development schedule

3.4 Requirement Specification

3.4.1 Functional Requirement

Central user interface module

- User interface must be full screen.
- The 2D map must be shown at all times either partially or completely.
- All the social networking modules must be able to trigger from the central user interface.

Authentication module

- User must login with username and password before entering the main dashboard interface.
- User may choose to remember login details so that details will retain on next application start.

Friend management module

- User should be able to find a user name from the search box and add a user from the result list as friend.
- User should be able to unfriend a user.
- Smart privacy setting allows user to put a maximum of 5 hashtag to describe himself.
- User should be able to view the interests of another.

Point of interest (POI) module

- User should be able to view the details of a POI by clicking it on the map.
- POI description should include a picture, uploader name, upload date, POI name and brief description.

- User should be able to submit a request to add POI by clicking a spot on the map, the request details may include POI image, name and description.
- Moderator or admin should be able to approve or reject the new POI request.

Event module

- User should be able to add an event request by clicking a spot on the map, the event details may include event image, date, time, venue, organizer, upload date and uploader name.
- User should be able to view the details of an event by clicking the event marker on the map.
- Moderator or admin should be able to approve or reject the new event request.

Messenger module

- User is allowed to send text messages to others regardless of the availability status of either side.
- User should be able to select one or multiple recipient from friends list.
- User should be able to see indication should there exist new reply or incoming message.
- User can create a chat room and leave a chat room.
- Chat history of participated room will be retained.

User profile module

- User should be able to change their personal information in the client program.
- User should be able to display certain personal information of other users.
- User should be able to select their interests to 5 maximum.
- User should be able to input 5 hash to describe himself without privacy setting.

- User should be able to upload his/her profile picture but only the latest will retain.

Installation package

- User should be able to install the application with a GUI installation wizard without typing any command
- User should be able to customize the installation destination for the application.

3.4.2 Non-functional Requirement

Synchronization

- All synchronization schedule must be started to run for the first time as the user login successfully.
- Agent user preferences should synchronize with server every 1 hour.
- POI module should synchronize with server every 2 hours.
- Event module should synchronize with server every 1 hour.
- Friend management module should synchronize with server in real time.
- Messenger module should synchronize with server in real time.
- User profile module should synchronize with server in real time.

Installation package

- The installation package should include a MySQL database, java executable file and the relevant library.
- The installation package will determine whether the computer is equipped with JRE and able to download it on demand.

- The installation package will assume there is no MySQL database preinstalled before the installation.

Others

- All the images from any modules should be stored on the server directory and retrieved on demand.
- All the data from any modules should be stored in database and not to hardcode in program code.

CHAPTER 4: PROJECT SPECIFICATION

4.1 Design Specification

4.1.1 Use-case Diagram

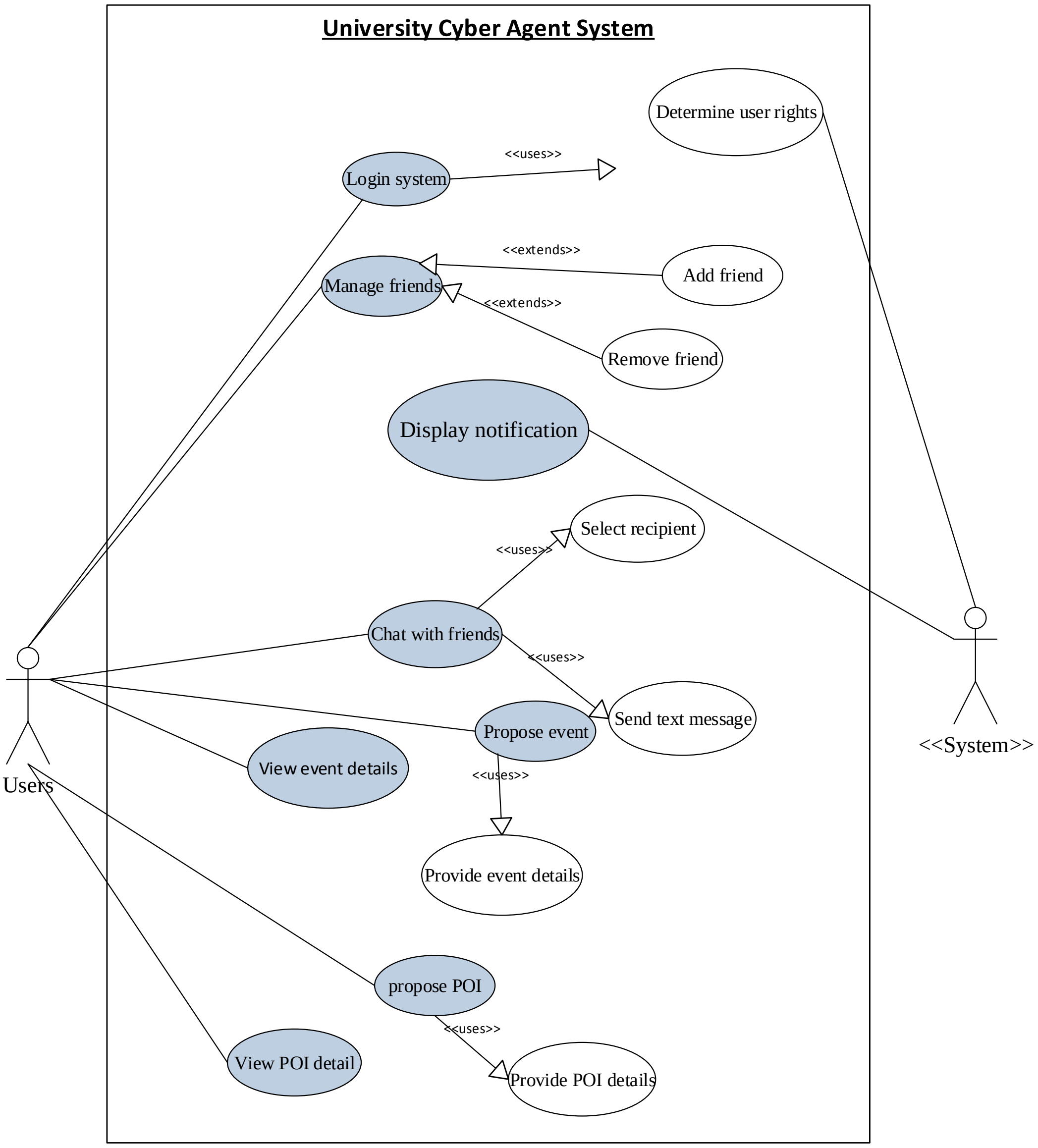


Figure 4-1-1-1 UniCAT SNS Use-Case Diagram

Assumption:

- System will display notification once it detects there is change in any social networking service.
- Shaded use case implies the function is either new or will be reengineered in this project.

4.1.2 Entity-Relationship Diagram

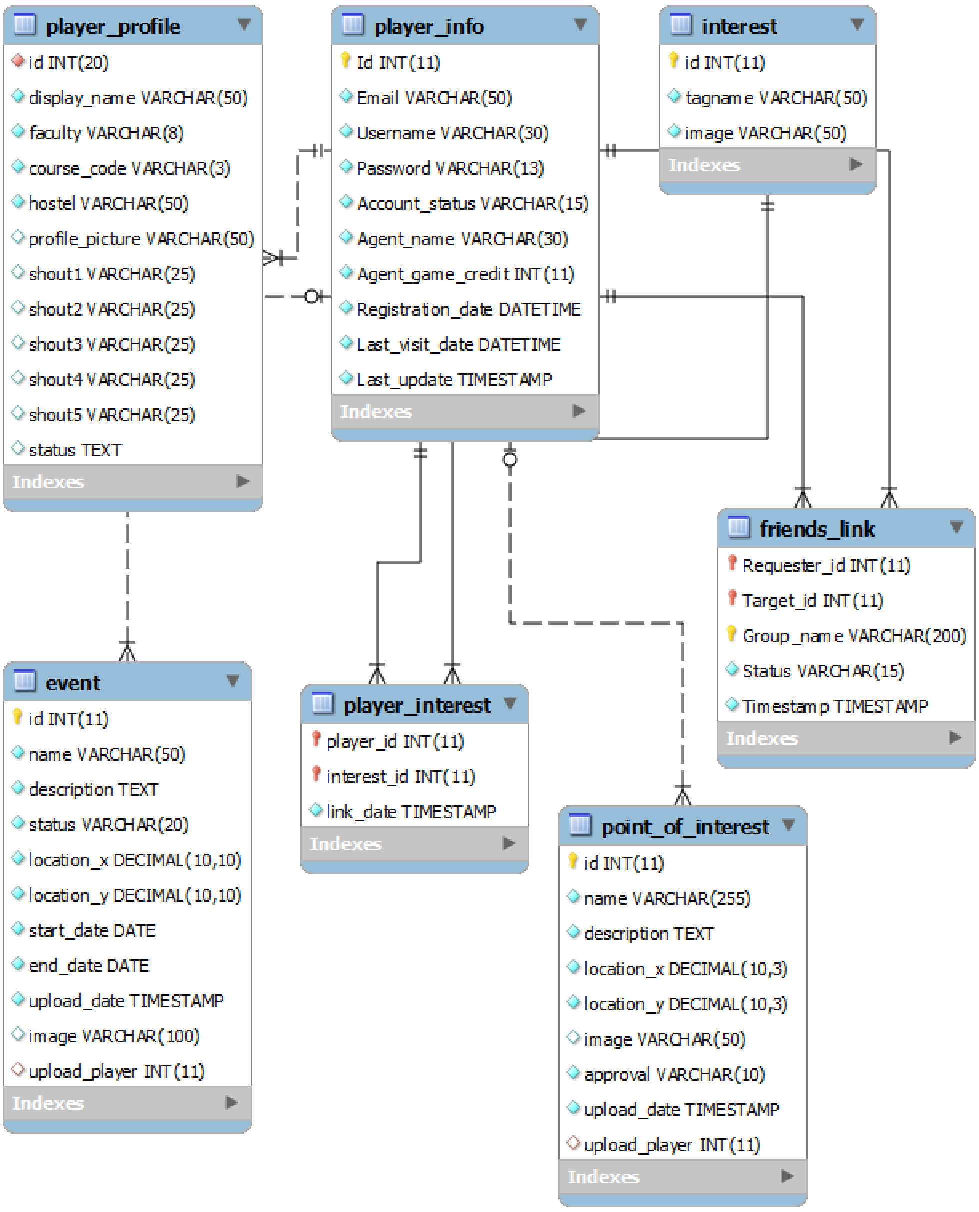


Figure 4-1-2-1 UniCAT SNS Entity-Relationship Diagram

- Figure 4-1-2-1 is depicting the proposed UniCAT system server database which act as a synchronization source for client database, hence client-side database is not illustrated.
- Figure 4-1-2-1 only includes social networking services database schema, there are still other tables related to the multi-agent system which have been omitted in the above diagram.

4.1.3 Sequence Diagram

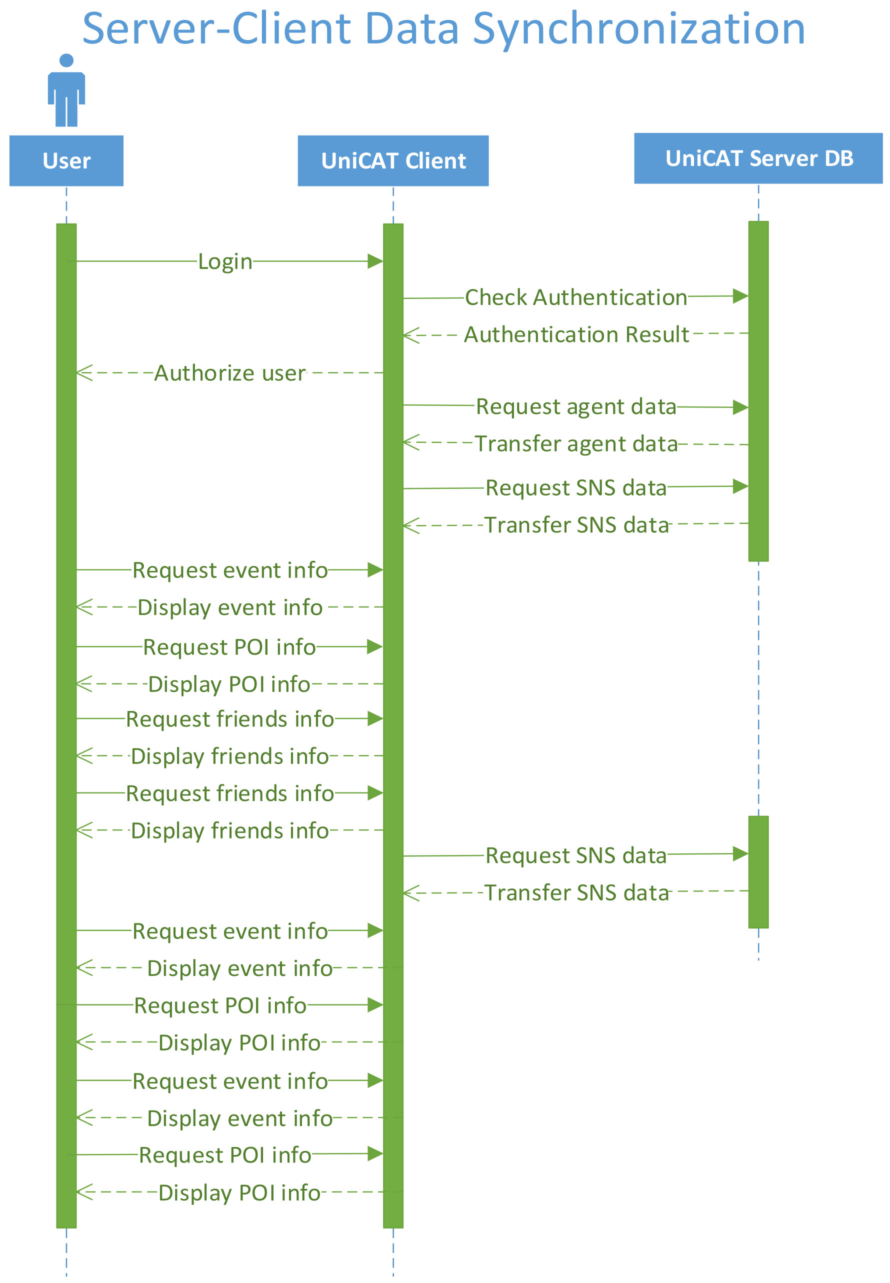


Figure 4-1-3-1 Server-Client Data Synchronization Sequence Diagram

Figure 4-1-3-1 is depicting the situation where Social Network Services modules are employing server-client data synchronization model. This diagram immediately exhibits the dominant advantage of adopting such model, it is obvious that server program activity is considerably lesser after client program serves as a major request handler, which means server I/O activity will be reduced.

4.2 User Interface Design

4.2.1 Main Dashboard

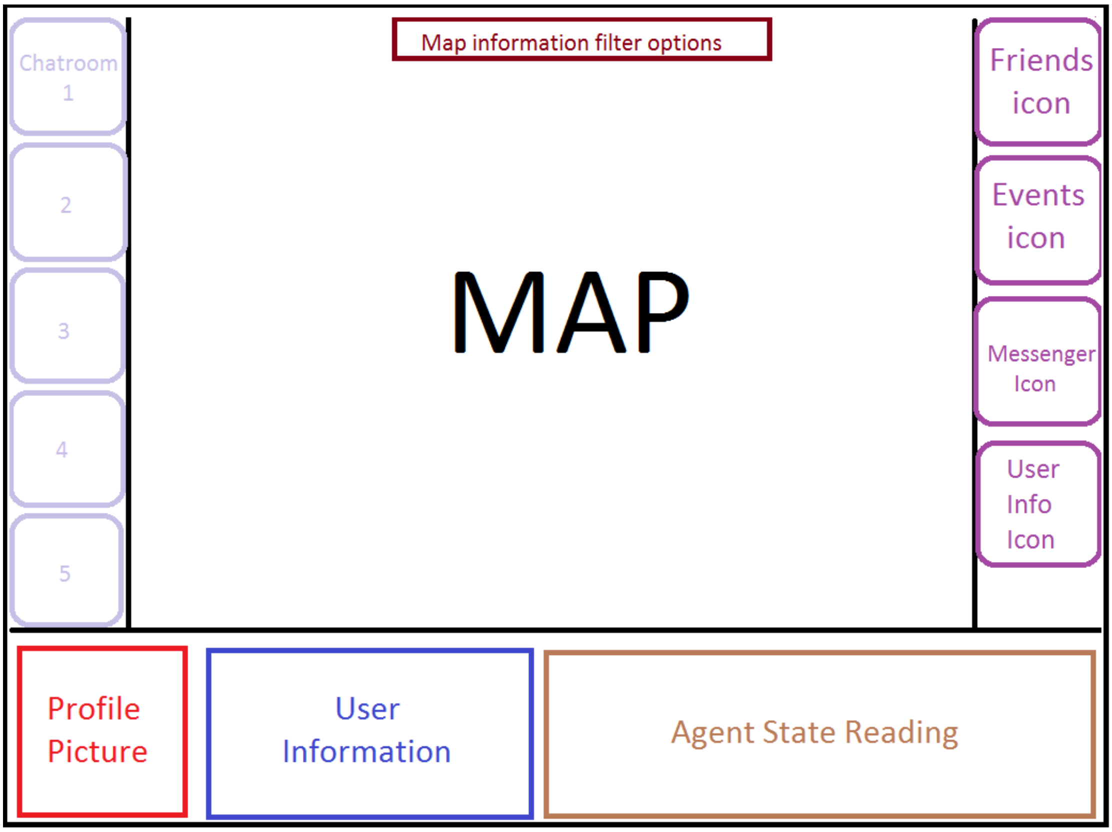


Figure 4-2-1-1 Main Dashboard UI Design

Figure 4-2-1-1 illustrates the actual design of the UniCAT client main windows. In this design, it uses a smart arrangement that enables major focus on the map and at the same time keeps an eye on the social network services update. Furthermore, this organization presents an informative display in parallel with comprehensive navigation possibility to access to every component of the software, this successful blend could hardly be seen in other software including those of simple one.

4.2.1.1 The Major Turnover

Compare to the existing UniCAT client program, this design is more than successful in consolidating all the standalone and sluggish user interface. By looking from the user experience perspective, the design has adopted a more professional and readable UI which includes the makeover of the agent information display and the navigation link presentation.

The ultimate turnover is that it practices an extensive multitasking environment, which is also one of the reason of incorporating the UI of scattered modules. Every panel on the map is designed to work independently without affecting any other panels, this has created a massive opportunity for the future development in creating a more flexibility function.

4.2.2 The Popup Window

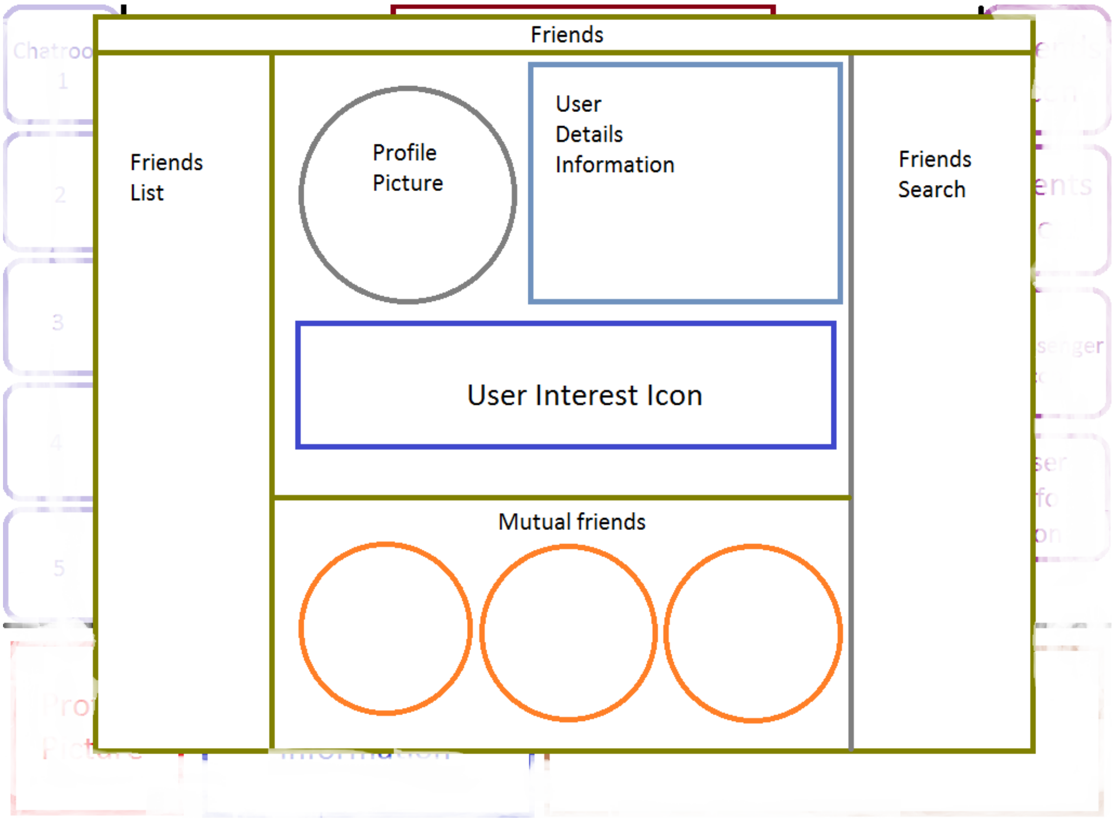


Figure 4-2-2-1 Friends Module Popup Windows

Figure 4-2-2-1 illustrates the design of a popup windows that shows a detailed friends module. This illustration is not only highlighting the arrangement of the user interface but also the design of the popup that leads to greater user experience. The distinguishing element of this design is the dynamicity of implementation, which will be further discussed later.

4.2.2.1 The Design of Popup Windows

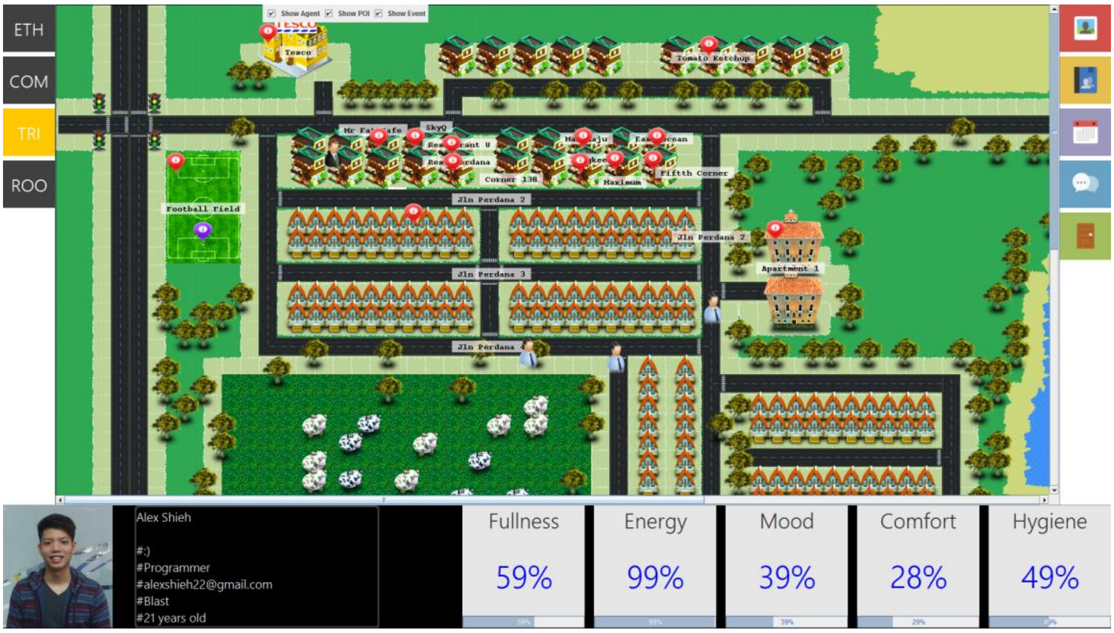
There is a notable difference between popup box and popup windows, the former has a relatively static setting which constraint the content presentation and the latter requires immense setting which includes low level decoration, screen size fitting and precise content positioning. The former are being employed by the existing UniCAT client and the latter will be used by the new client program

All the popup windows in client program are sharing the unified theme design which contained several characteristics such as the background transparency, font size and reaction in different screen resolution. Firstly, the background of the popup are partially transparent as to retain user context on the background on-going. Secondly, the arrangement and size of the content are set arithmetically to accommodate various screen resolution. Thirdly, unless explicit instruction, the handling of all the popup windows are standardized. For instance, way to close the popup windows.

CHAPTER 5: IMPLEMENTATION AND TESTING

5.1 Implementation

5.1.1 Main Dashboard



Alex Shieh

#:)
#Programmer
#alexshieh22@gmail.com
#Blast
#21 years old

Fullness	Energy	Mood	Comfort	Hygiene
59%	99%	39%	28%	49%

Figure 5-1-1-1 Main Dashboard Screenshot

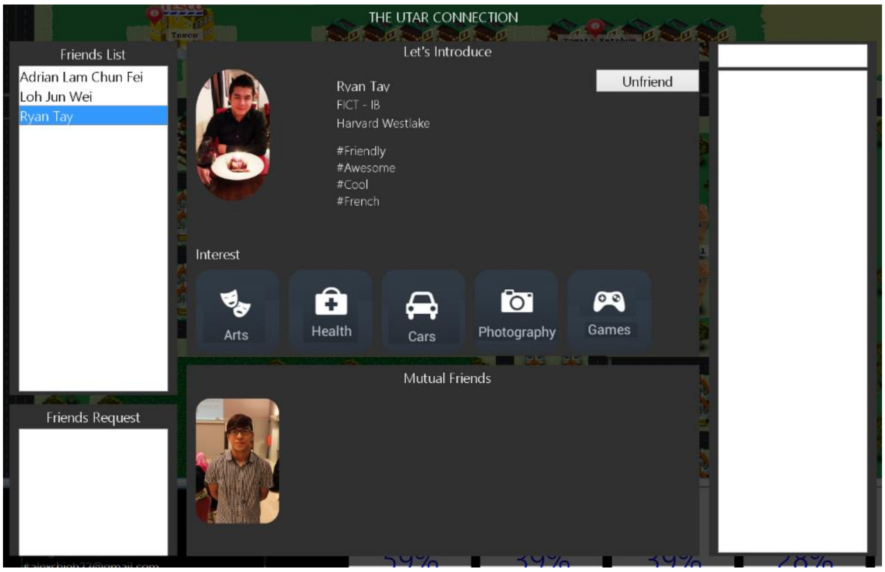
Incorporating both the multi-agent system and the social network services in the same windows requires tremendous loading. Therefore, multi threads have been used to initialize the user interface concurrently.

The major change from the previous is the total makeover of the layout, the exhibition of agent information (bottom), the presentation of the navigation link (right) and the Messenger notification (left).

Besides, the map came with a filter options to enable user to choose which element to be displayed on the map, be it event, POI or agent. Besides, instead of only

POI and event, now all the moving agents also carrying a mouse event listener to trigger event as necessary.

5.1.2 Social Network Services



Adrian Lam Chun Fei
Loh Jun Wei
Ryan Tay

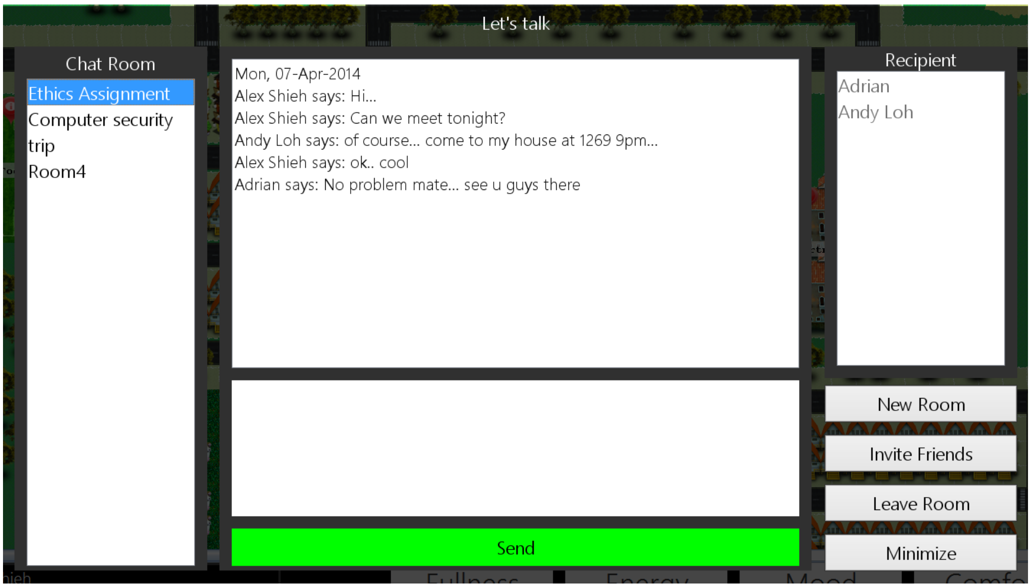
Ryan Tav
FICT - IB
Harvard Westlake

#Friendly
#Awesome
#Cool
#French

Figure 5-1-2-1 Friend Popup Windows Screenshot

This module was total build from scratch, the functions include display friends profile, search for friends, display mutual friends, add friends, unfriend Friends, handling friend request, display friends personal interest. Some of these functionalities can be found in the previous UniCAT, however those can only be used externally on the web browser.

The notable feature of this module is the “Smart Privacy”, where user is given 5 hashtag quota to express themselves from time to time. This approach has eliminated the necessary of complicated privacy setting. Another remarkable feature is the user interest, where user can choose 5 out of 24 preloaded interest to indicate user’s personal interest.



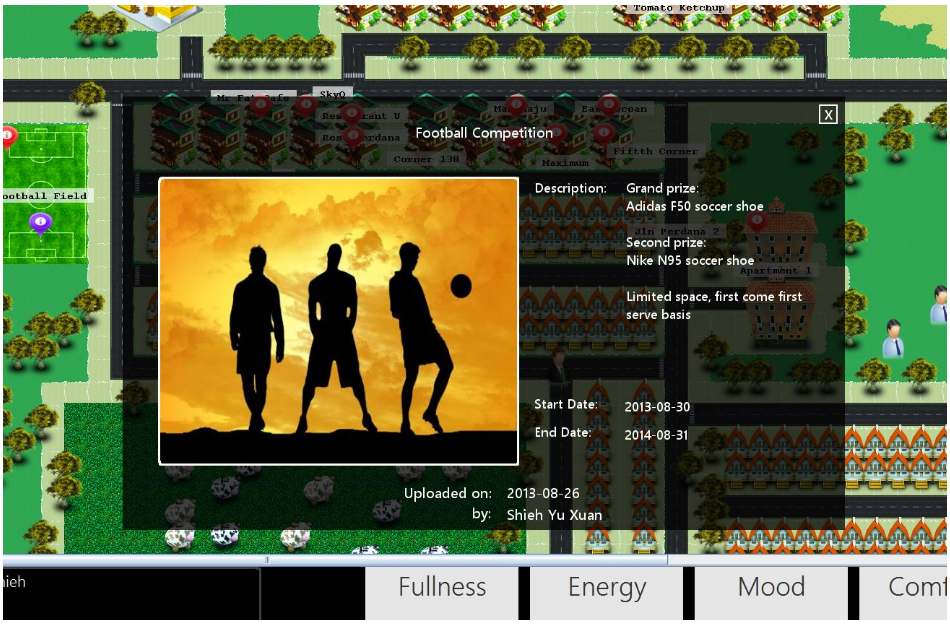
Mon, 07-Apr-2014
Alex Shieh says: Hi...
Alex Shieh says: Can we meet tonight?
Andy Loh says: of course... come to my house at 1269 9pm...
Alex Shieh says: ok.. cool
Adrian says: No problem mate... see u guys there

Figure 5-1-2-2 Messenger Popup Windows Screenshot

This module was built from scratch, new features include room-based chatting environment, multiple room support, chat history retention and recipient search. Although there is no specific place for notification like any other software, however there is an unread message indication in the main dashboard which highlights chat room link in orange color.

Aside from the viewable user interface and functionality, the backend setup is complicated. UniCAT client has to register its socket at the server at startup and register new chat room to the server as created, so the server can broadcast the messages by looking at the recipient list in the chat room and thereafter looking for the registered socket for each participated user.

All the messages will be stored in the server memory but not the database, this is to instill the peace of mind in user without being suspicious of breaching the privacy. All the users will retrieve all the created chat room and messages on UniCAT startup.



Football Competition

Description: Grand prize:
Adidas F50 soccer shoe

Second prize:
Nike N95 soccer shoe

Limited space, first come first serve basis

Start Date: 2013-08-30

End Date: 2014-08-31

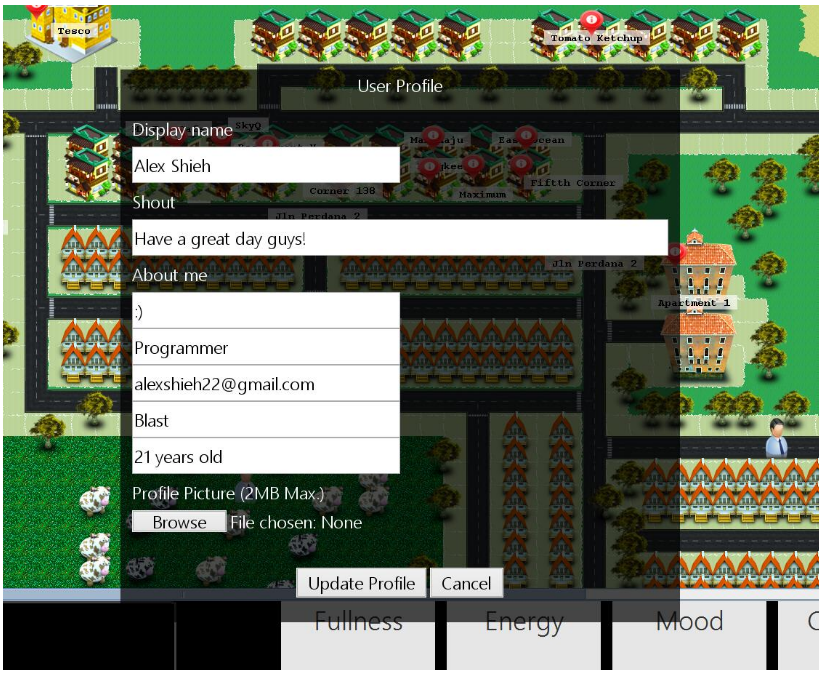
Uploaded on: 2013-08-26

by: Shieh Yu Xuan

Figure 5-1-2-3 Event Popup Windows Screenshot

This module was built by referring the previous UniCAT concept, new changes include abolishing the JLabel on each event that works as an image holder and event handler on the map which consumes much memory and changed to image draw on map and event trigger is determined dynamically according to mouse click position on the map.

Besides, both POI and Event has a whole new interface to submit a new POI or event to the moderator. The new submission system comes with validation setting which verify the upload image file size according to restriction and validate the mandatory field to be inserted.



User Profile

Display name

Alex Shieh

Shout

Have a great day guys!

About me

:)

Programmer

alexshieh22@gmail.com

Blast

21 years old

Profile Picture (2MB Max.)

Browse File chosen: None

Figure 5-1-2-4 Profile Update Popup Windows Screenshot

This module was built from scratch, the previous UniCAT did not incorporate any mechanism that holds the personal information. However, the new UniCAT requires such system to enable SNS modules to go further and lead to better user experience.

All the image upload including the profile picture has adopted a way to rename the picture and store on the server.

File name: me.jpg

New file name: [yyyyMMddHHmmss]_me.jpg

This approach is to eliminate the chances of colliding name and replacing each other and at the same time retain information on uploading date time.

5.1.3 Others

There are some insignificant but important modules and setup that were not explicitly mention above which include the approach in setting up database, connection and similar SNS module. Since the UniCAT has been redesigned to accommodate future enhancement, one can expect the characteristic and design are of similar from one module to another.

Mentioning the code construct in the implementation aspect, the new UniCAT has a major makeover on the code itself to comply with the conventional code design and practices. Besides, best practice has also been used in various scenario to ensure that performance is the least thing to worry.

5.2 Black Box Testing

5.2.1 Friends Module

Friends Module			
Case	**Test Action**	**Test Result**	**Status**
1	Click on Friends module navigation link.	Friends module popup windows appeared.	Pass
2	Click on one of the friends in friends list.	All the user details, pictures and mutual friend's picture displayed on the middle panel, button controls were appeared correctly.	Pass
3	Click on the "Unfriend" button after selected a user.	The user disappeared from the friends list.	Pass
4	Type one character on the user search box.	Nothing happened, no result.	Pass
5	Type more than one character on the search box.	The search result came out correctly with the only user name consisting the keyword.	Pass
6	Click on one of the users in the search result.	All the user details, pictures and mutual friend's picture displayed on the middle panel, button controls were appeared correctly.	Pass
7	Click on the "Add Friend" button.	The button state changed to disabled and text changed to "Friend request sent".	Pass
8	After case 7, select other user.	Result same with case 6, the button is enabled and text changed appropriately.	Pass
9	After case 8, select the previous user.	The button state changed to disabled and text changed to "Friend request sent".	Pass

10	Click on the user on Friend Request List.	All the user details, pictures and mutual friend's picture displayed on the middle panel, button controls were appeared correctly.	Pass
11	After case 10, click on the "Accept" button.	The button changed to "Unfriend" and user moved to Friends List.	Pass
12	Click on another user on Friend Request List and click "Deny" button.	The button changed to "Add Friend" and user is removed from Friend Request List.	Pass
13	Click on the outside of popup windows.	The popup windows was closed.	Pass

5.2.2 POI Module

POI Module			
Case	**Test Action**	**Test Result**	**Status**
1	Click on the POI marker on the map.	The POI popup windows displayed with appropriate information and image.	Pass
2	Click on the POI marker that has no image attached	The displayed POI image is the default "No Image" picture.	Pass
3	Click on the outside of popup windows.	The popup windows was closed.	Pass
4	Right click on the map and select "Add POI".	A popup windows appeared with form as content.	Pass
5	Fill in partial of the form and click "Submit" button	An popup message box appeared as warning for all mandatory field.	Pass
6	Fill in all mandatory field without attaching image and click "Submit" button	Success popup message appeared and popup windows was closed.	Pass

7	Attach an image size more than 3MB and click “Submit” button	Error popup message appeared as warning to change to smaller image	Pass
8	Attach an image size less or equal to 3MB and click “Submit” button	Success popup message appeared and popup windows was closed.	Pass

5.3 White Box Testing

5.3.1 Messenger Module

Messenger Module			
Case	**Test Action**	**Test Result**	**Status**
1	**Server testing:** Starts the following threads in this sequence. 1. Socket registration 2. Chat room registration 3. Message broadcast 4. Message box	All three threads successfully started and each listening in different ports respectively.	Pass
2	**Client – Server testing:** Client send the user id to the Socket Registration.	Socket Registration socket captured the message and started a worker thread to store the client socket and user id into a socket List.	Pass
3	**Client – Server testing:** Client send the chat room data to the server Chat Room Registration.	Chat Room Registration captured the data and started a worker thread to add the new chat room into server memory.	Pass
4	**Client – Server testing:** Client send the message to the Server Message Broadcast.	Message Broadcast captured the message and started a worker thread to broadcast the message to all the users with registered socket.	Pass
5	**Client – Server testing:** Server repeatedly send different messages from different chat room to the same socket.	Client is able to determine the chat room and add the new message into it.	Pass
6	**Client – Server testing:**	Server detected previously registered socket is	Pass

	Client shut off in the middle of chatting.	unknown host and automatically drop the socket from list.	
7	**Client – Server Testing:** Client send a message to a user that is not online.	If server cannot locate the user from socket list, it will store the message into a Message Box for recipient to retrieved when startup afterward.	Pass
8	**Client – Server Testing:** Client first startup, registered socket and look for server Message Box.	Server captured the request and send the pending message back to recipient.	Pass

CHAPTER 6: PROJECT REVIEW

6.1 Project Review

There are three main pie of objectives in this project:

- Decentralization of the structure of agent system.
- Reengineering of Social Networking System.
- Construction of intuitive and user friendly user interface.

These 3 objectives carry an immense workload and challenges in each of them, however in order to publish this software, this is the least to be done. The discussion below will explain the challenge in progression and the contributed solution.

6.1.1 Decentralization of the Architecture of Agent System

The process of decentralization was started from the previous project, however the progress only covered the migration of agent engine to the client program. This migration was completed as a research and testing phase, hardcode data, unresolved bugs have a heavy presence in the code structure. This serves as a great challenge as the code structure is not dynamic and developed in a way that it is not welcoming further development.

Hence, in order to complete the decentralization process, this project has to reconstruct the agent backend program to increase dynamicity and add in realistic function as a result of concern to performance. This additional scope has tremendously increase the man time on top of the already lengthy allocated time and causes delay in the preset milestone. Thus, this section has to cut corners on the initial target and change the priority to reengineer the agent system to make the future work substantially easier and possible so that the future development can be hindered from the challenge that is encountered now.

The following shows what have been achieved in this section:

1. Migration of hardcode map information (agent moving way point) to database.
2. Added function to synchronize agent information with the server.
3. Added dynamic function in getting friends information and startup as agent.
4. Added interactive event on moving agents to display comprehensive agent information.
5. Added number limit on displaying agent as a performance precaution.
6. Added concurrency/multi-core programming concept to enhance performance.
7. Constructed dynamic application structure to prepare for future map extension and functionality expansion.

6.1.2 Reengineering of Social Networking Systems

The program before this project came with event, point of interest, friend and messenger functionality. These functionality are only able to serve as a basic concept of the whole application, the actual usability of them is a disaster. For instance, messenger does not allow multiple chat room but only one and closing the chatting windows will crash the module. Another example would be the friend module which needed user to externally open web portal on browser to respond to friend request instead of integrating it with the program.

These design issue has caused the enhancement on these modules to be impossible. The solution adopted in making the program success is to reengineer all the modules, in another word, rebuild them from scratch, in a more realistic and usable way. This solution has intensely challenging the preset milestone and distracting the initial focus which is to increase the functionality in the social networking services family. Hence, the priority is given to the precision of client program instead of its counterpart admin program, database data may be needed to technically change by the administrator for them to moderate the social networking services.

The following shows what have been achieved in this section:

1. Added login mechanism for users.
2. Enhanced the information display for POI and event
3. Migrate the hardcoded POI and event information into the database.
4. Added map information filtering options.
5. Developed a consolidated friend module, new functions include viewing mutual friends, searching for users, smart privacy control, profile picture, personal interest indicator and friends making control.
6. Developed a messenger module, new functions include group chatting, chat-room based environment and chat over the internet instead of LAN.
7. Developed an event module, new functions include display all the current and past event in one place and automatically remove the event marker on map once expired.
8. Developed a user profile module, functions include allow user to change personal information, statuses and smart privacy setting.

6.1.3 Construction of User-friendly User Interface

The previous application came with scattered user interface for different module, every module will initialize its user interface in a new window. With several module in use simultaneously, heavy switches among the windows is a nightmare. The priority of this section is to improve user experience in using the application, HCI analysis was made on other application beforehand. The solution for this issue was to create a consolidated and unified user interface, totally redesigning how user should interact with the system.

The following shows what have been achieved in this section:

1. Transformed window-based application to full screen and at the same time cater multiple screen resolution.
2. Applied a unified theme display onto the user interfaces of all the social networking modules, all of them have a brand new look.
3. Added a dashboard at the bottom to display the information for the selected agent.
4. Added a control center with all the buttons to social networking modules.
5. Added a messenger center for chat room easy access and notification.

6.1.4 The Shift of Concern in Multi-Agent System

The multi-agent structural reengineering of this project was due to the concern of intense server hardware requirement in ensuring top notch performance in proportional with great number of simultaneous users. The decentralization has certainly removed the concern and the result was remarkable, this can be attributed to the removal of intense agent data synchronization between server and all the users, this has made the server a central data center instead of the former data and processing center. In other way of explaining, the server will communicate with the client in preset once per hour instead of once per second.

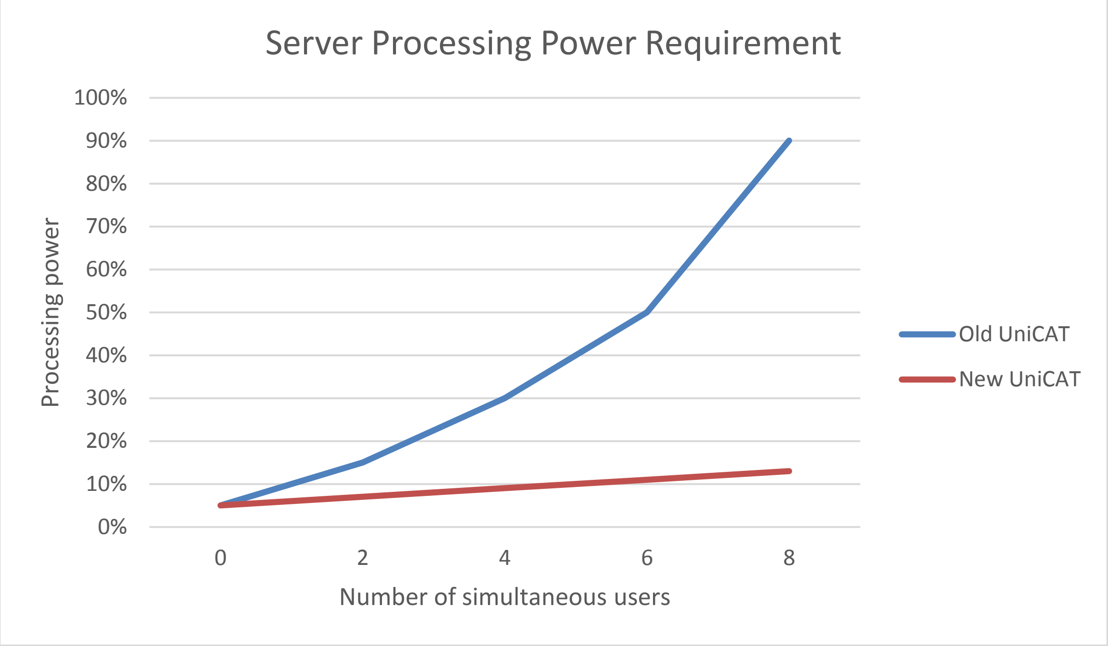


Figure 5-1-4-1 Server Processing Power Requirement Graph

Figure 5-1-4-1 depicts the relationship between server processor consumption reading with the number of simultaneous users. It contrasts the difference in processor usage reading between the old and the new UniCAT architecture, which apparently the new UniCAT system has minimal effect on the incline of user numbers. This phenomenon can be perceived as problem solved, beautifully.

However, the concern has now shifted to the client side. How many agents can be spawned and move at the same time? This concern remains the main consideration throughout the whole reengineering process, especially dilemma on whether showing more agents or ensuring a better performance on other modules.

By using a test bed with Intel i7 2.40GHz (Quad-core) CPU on the client side, the result was surprising. The requirement was harsher than the previous after full reengineering, which is not welcoming but explainable.

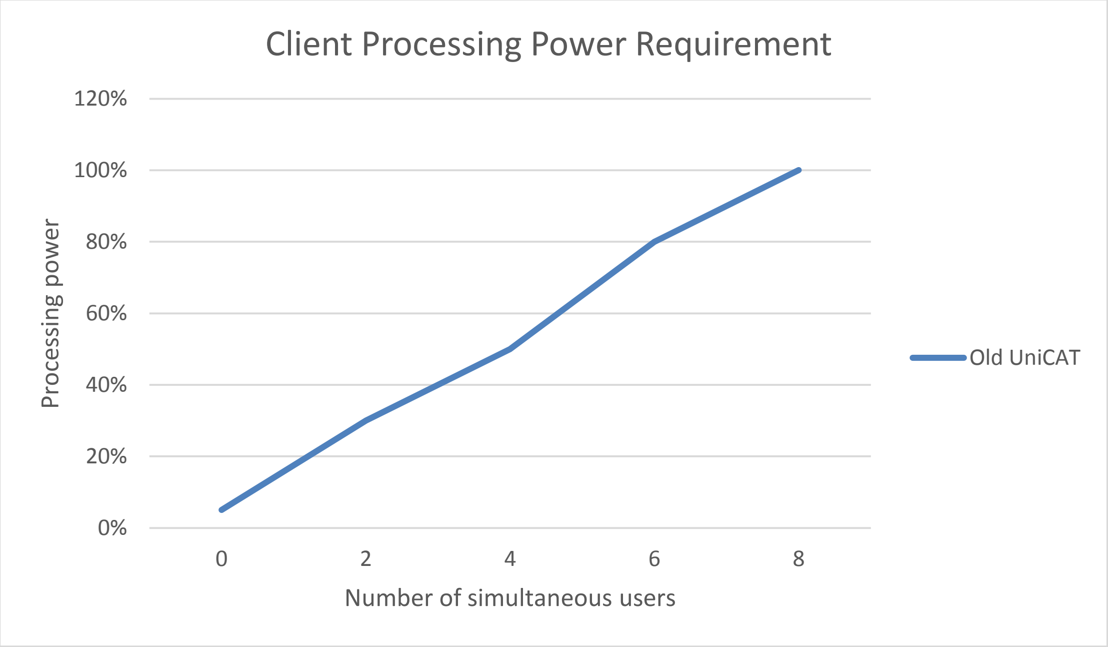


Figure 5-1-4-2 Client Processing Power Requirement Graph

Figure 5-1-4-2 shows the boosting of CPU readings with increasing agents moving on the map. Explaining the CPU reading difference in server and client, it may be due to the new functionality on the agent. Changes include moving the agent information from the map to the bottom panel, which in this process necessarily added agent sort, looping and mouse listener on every agent moving cycle and these extra components were believed to be the main CPU usage booster. Hence, the client program has limited a maximum 8 agents to be spawned and at the same time implemented mechanism to allow user to not show the agent therefore to preserve user experience of SNS module.

CHAPTER 7: CONCLUSION

7.1 Conclusion

The ultimate propeller of this project is clearly stated in chapter, which is to reengineer the UniCAT system to the publishable system. In between, it involves heavy modification in the existing structure and not only that, it too include integration of new features and functionality. Making a software publishable requires attention to details, which also emerged as a pitfall that dragged this project beyond preset timeline. However, innovative and alternate solutions were founded along the way to make the objective a clear hit.

The construction of the unified and consolidated user interface was fully completed. Even though things were not fully complying with the initial design due to technical and time constraint, however alternative ways were discovered to cover the original purpose. The complete of the new user interface means two things, not only user will acquire a much better user experience compared to the previous interface but also allow developer to have a peace of mind in developing extra modules to readily plug into the UniCAT system.

The migration of agent BDI engine from the server to the client side was fully concluded. Despite dominant advantage has emerged at the server, however concern must be shifted to the client program which was having a severe performance impact than the former due to the integration of all modules. Though, fortunate among the unfortunate says that few clients having performance stutter is better than the whole UniCAT having it.

The reengineering of the social network services took the longest time in this project, simply because the previous module was a design nightmare and did not allow further development to take place. Making UniCAT publishable has no other way but to back to the drawing board and rebuild all the SNS modules from scratch. Meantime, attention to details in every module requires precise blend between unified user interface and its core functionality. Although the social network services has left out some counterparts that served as a complement component such as notification, however the core functionality is preserved and future enhancement effort has been made easier.

Reengineering of UniCAT client has been a journey of up and down, creating solution and unknowingly inviting new issue. Innovation and early anticipation stand the vital element in making this project a success. Even though the standard of this project deliverable may not reach the desire publishable standard, however it can be used as the first publishable prototype system to collect research data, records user feedback and make a better comeback in the future.

7.2 Future Work

Firstly, the most important module that needs to be done is the administrator control panel. This is an independent application that shares the same server database with the client, to extract, analyze and control data. In terms of the agent system, admin control panel can access to all the users preferences and tabulate the statistics in a human understandable form (graph/chart). This is crucial for the purpose of research and development on the agent intelligent mechanism. Besides, a social networking service monitoring system is needed, administrator should be able to moderate the content that the users are sharing.

Secondly, the overview map is important as the UniCAT map is growing bigger. Pursuing this further, the mini map can have some division and color differently according to user preferences. For examples, if the user likes to watch cinema and eat Japanese food then the area with these entertainment will be highlighted, the reason behind is that users can now know where to meet their new friends which might have the similar interest. This idea is what makes a graphical social networking application outdo the existing text based social networking site.

Thirdly, a comprehensive self-updater program should be integrated into the UniCAT client application so that it can accommodate new development by downloading the latest version and update itself. In details, the updater program should be able to identify the affected part and only update that particular part of the program without the user awareness.

Fourthly, a central notification system is not discountable in social networking software, gathering all the recent information in a place will add score in program usability too. On the other hand, real-time push notification will greatly improve the user experience and it encourages more functionality to be added in the SNS family especially those which can effectively make use of the push service.

Lastly, a mobile apps should be developed for the users to communicate uninterruptedly, so that they can use the SNS function on the move. Meanwhile, mobile apps should be designed to capture the location data of the users to analyze their frequent destination and lead to a more precise user preference. Heavy computational module such as agent module can be excluded from the mobile version but new functionality that are fitting to mobile device should be added as substitution. For example, NFC touch to make friends, using gesture to perform action and check-in on map using GPS.

BIBLIOGRAPHY

Active Components, 2012. Jadex Features. Available from: <http://www.activecomponents.org/bin/view/About/Features>.[1 June 2013].

Belief–desire–intention software model, 2013. Available from: <https://en.wikipedia.org/wiki/Belief%E2%80%93desire%E2%80%93intention_software_model>.[11 June 2013].

Donnerwood Media 2012. Meez. Available from: <http://www.meez.com/>.[10 June 2013].

Electronic Arts 2013. The Sims™ FreePlay. Available from: <http://www.ea.com/the-sims-freeplay-ios>.[23 June 2013].

Multi-agent system, 2013. Available from: <https://en.wikipedia.org/wiki/Multi-agent_system>.[10 June 2013].

Second Life 2013. What is Second Life. Available from: <http://secondlife.com/whatis/?lang=en-US>.[15 June 2013].

Sulake 2013. Welcome to Habbo. Available from: <https://www.habbo.com/>.[19 June 2013].

Teach-ICT.com Limited, n.d. Evolutionary Prototyping. Available from: <http://www.teach-ict.com/as_a2_ict_new/ocr/A2_G063/331_systems_cycle/prototyping_RAD/miniweb/pg3.htm#>.[12 June 2013].

The Nerdfiles.com, n.d. What is Second Life. Available from: <http://www.thenerdfiles.com/page41.html>.[20 June 2013].

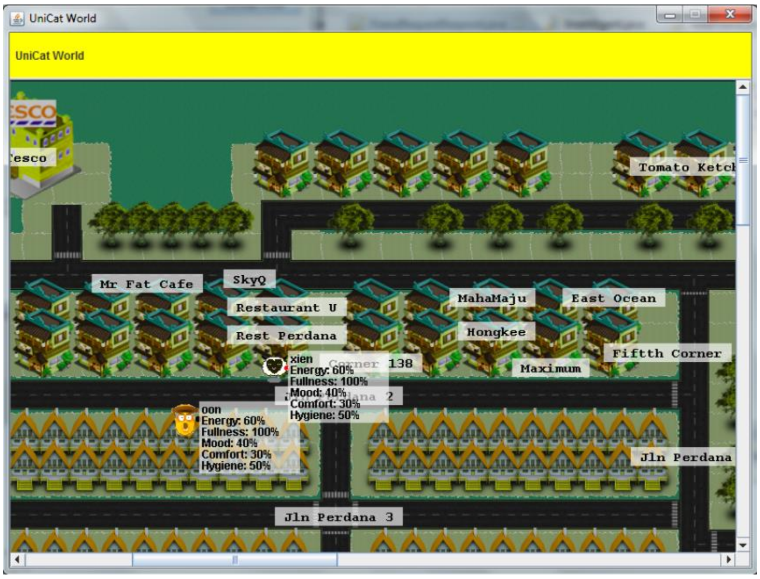
Valve Corporation 2013. Defense of the Ancients. Available from: <http://www.playdota.com/>.[20 June 2013].

WeeWorld 2013. WeeWorld. Available from: <http://www.weeworld.com/>.[22 June 2013].

Zynga 2013. CastleVille. Available from: <http://company.zynga.com/games/castleville>.[25 June 2013].

Zynga 2013. FarmVille 2. Available from: <http://zynga.com/game/farmville-two>.[26 June 2013].

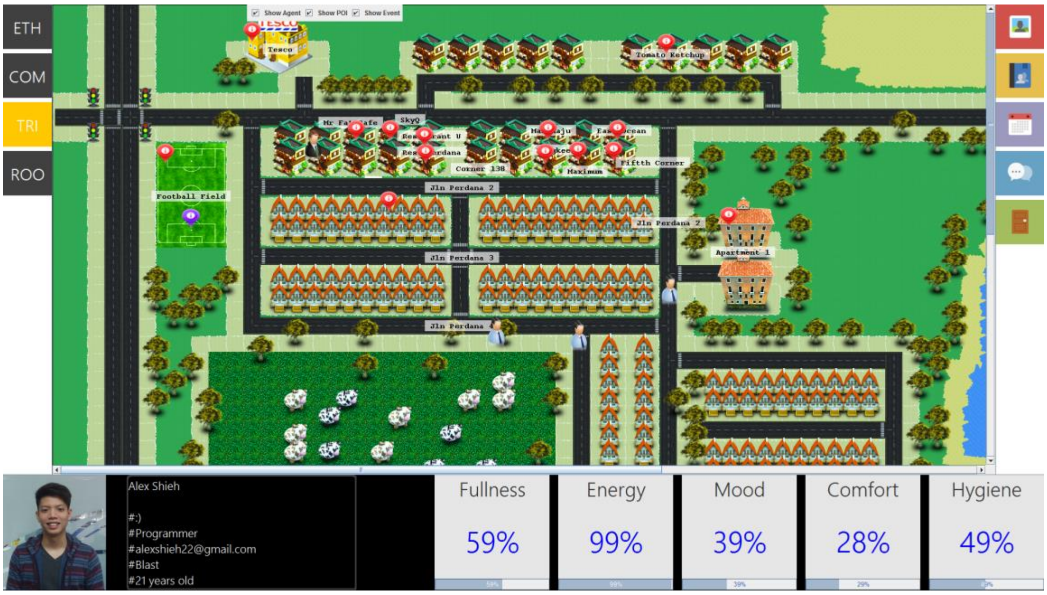
APPENDIX A – BEFORE/AFTER COMPARISON



xien
Energy: 60%
Fullness: 100%
Mood: 40%
Comfort: 30%
Hygiene: 50%

oon
Energy: 60%
Fullness: 100%
Mood: 40%
Comfort: 30%
Hygiene: 50%

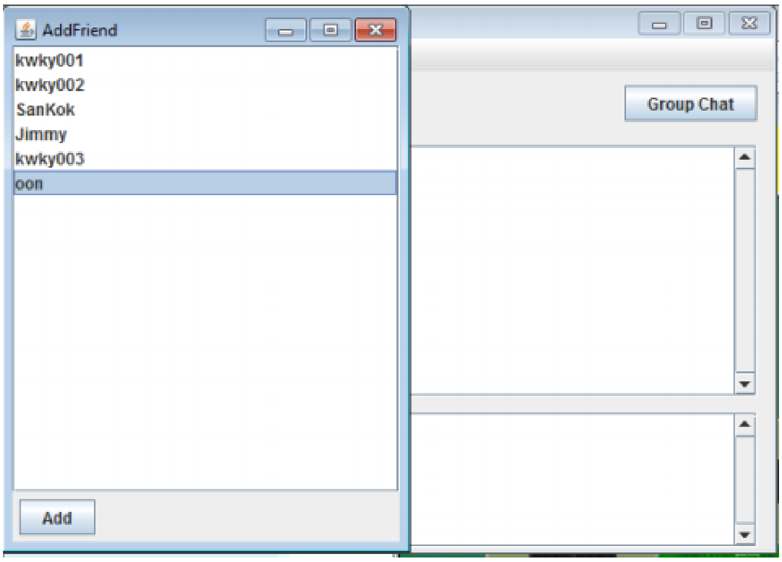
Appendix A-1 Main Windows in Old UniCAT



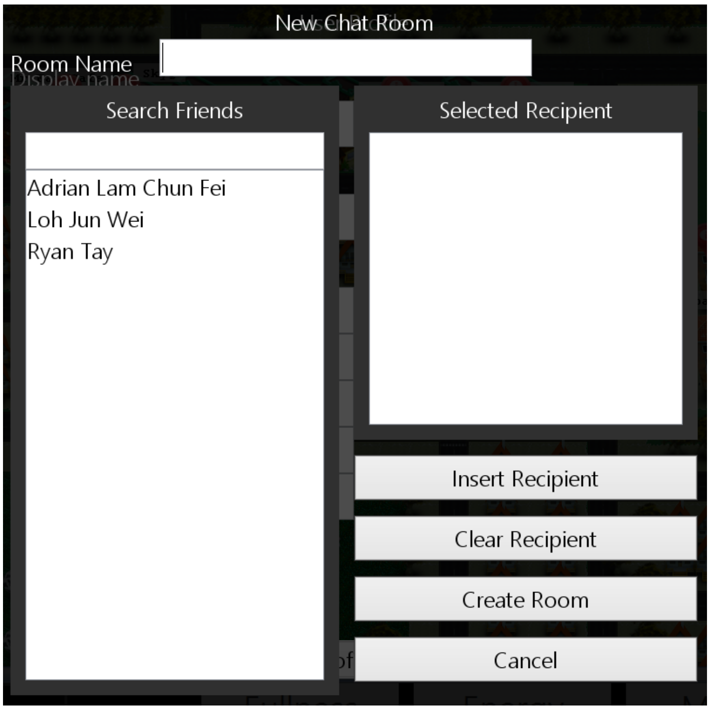
Alex Shieh
#:)
#Programmer
#alexshieh22@gmail.com
#Blast
#21 years old

Fullness	Energy	Mood	Comfort	Hygiene
59%	99%	39%	28%	49%

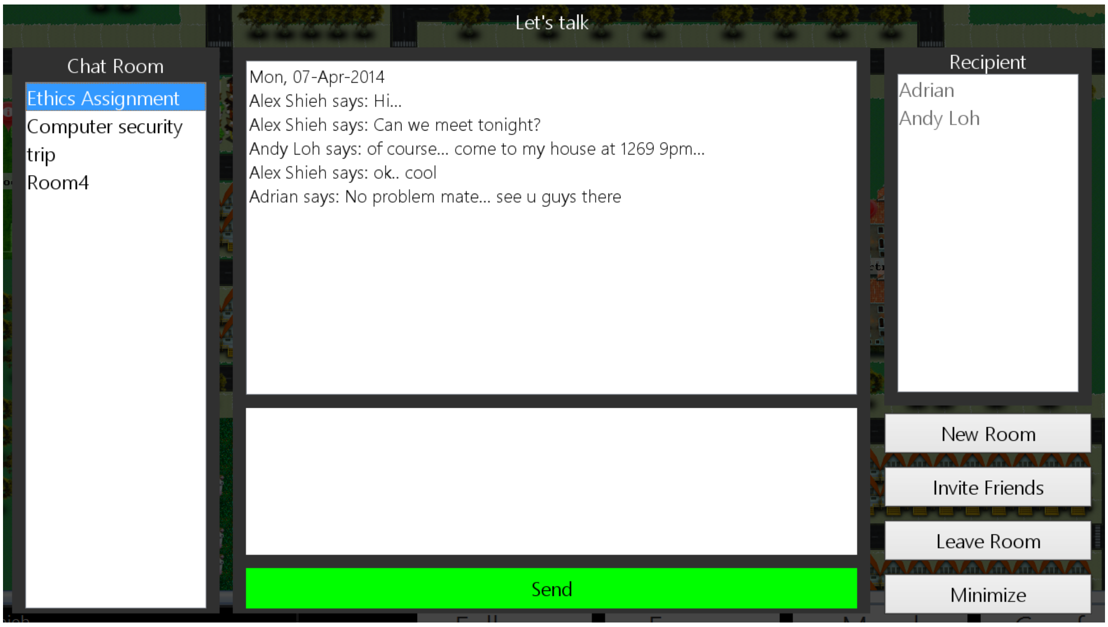
Appendix A-2 Main Windows in reengineered UniCAT



Appendix A-3 Create Chat Windows in Old UniCAT

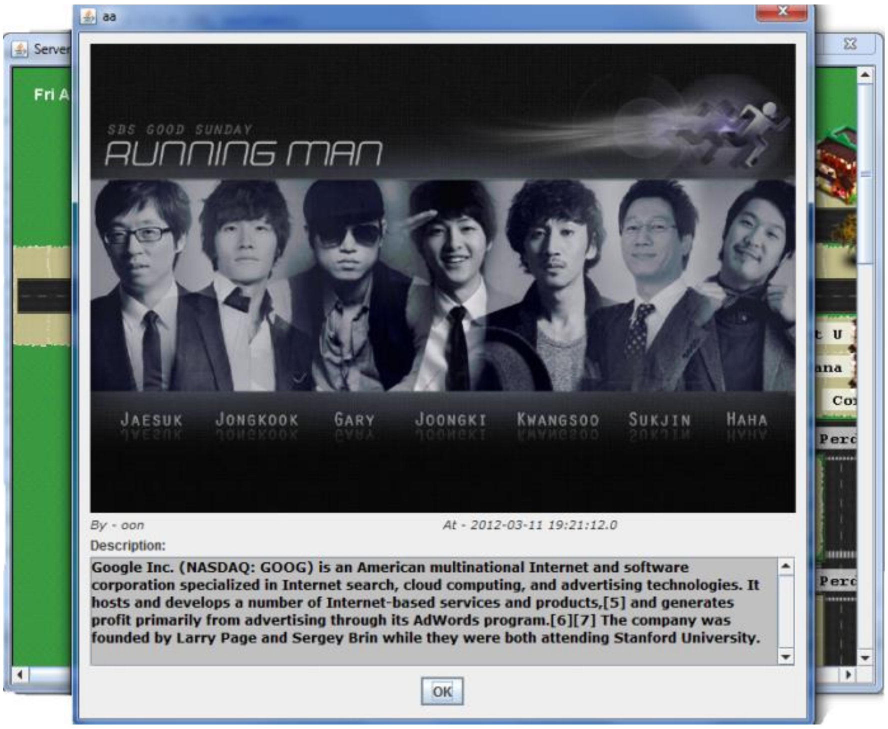


Appendix A-4 Create Chat Room Windows in Reengineered UniCAT



Mon, 07-Apr-2014
Alex Shieh says: Hi...
Alex Shieh says: Can we meet tonight?
Andy Loh says: of course... come to my house at 1269 9pm...
Alex Shieh says: ok.. cool
Adrian says: No problem mate... see u guys there

Appendix A-5 Messenger Chat Windows in Reengineered UniCAT



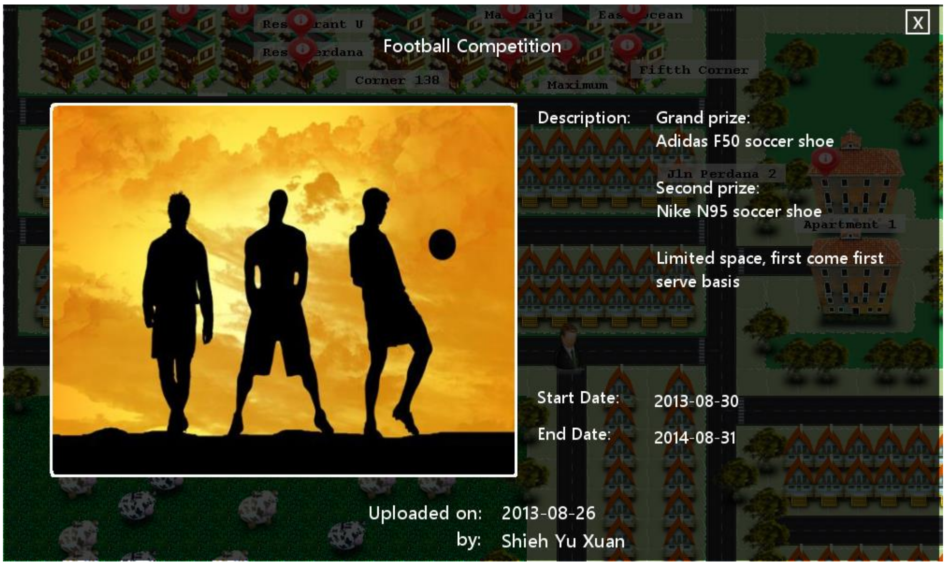
By - oon

At - 2012-03-11 19:21:12.0

Description:

Google Inc. (NASDAQ: GOOG) is an American multinational Internet and software corporation specialized in Internet search, cloud computing, and advertising technologies. It hosts and develops a number of Internet-based services and products,[5] and generates profit primarily from advertising through its AdWords program.[6][7] The company was founded by Larry Page and Sergey Brin while they were both attending Stanford University.

Appendix A-6 Event Popup Windows in Old UniCAT



Football Competition

Description: Grand prize:
Adidas F50 soccer shoe

Second prize:
Nike N95 soccer shoe

Limited space, first come first serve basis

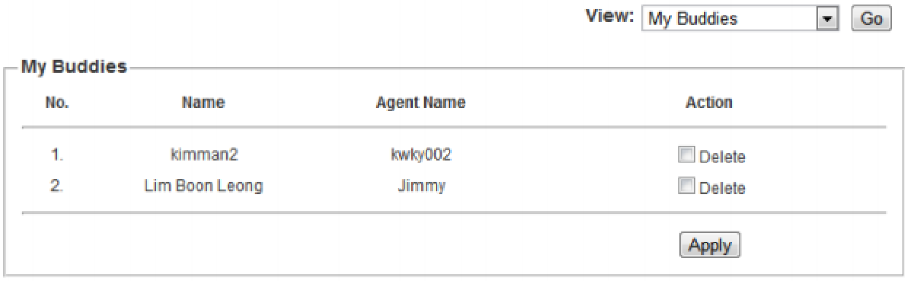
Start Date: 2013-08-30

End Date: 2014-08-31

Uploaded on: 2013-08-26

by: Shieh Yu Xuan

Appendix A-7 Event Popup Windows in Reengineered UniCAT

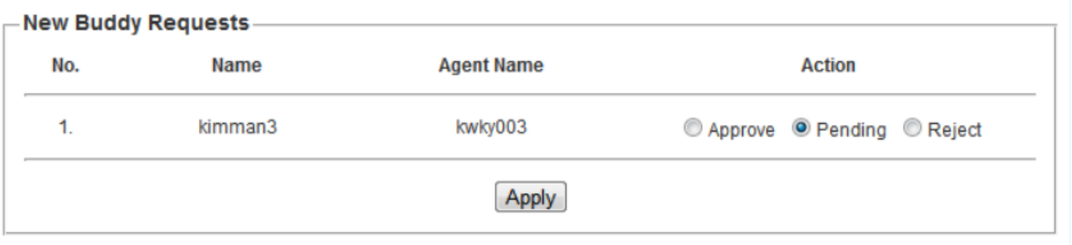


View: My Buddies

My Buddies

No.	Name	Agent Name	Action
1.	kimman2	kwky002	□ Delete
2.	Lim Boon Leong	Jimmy	□ Delete

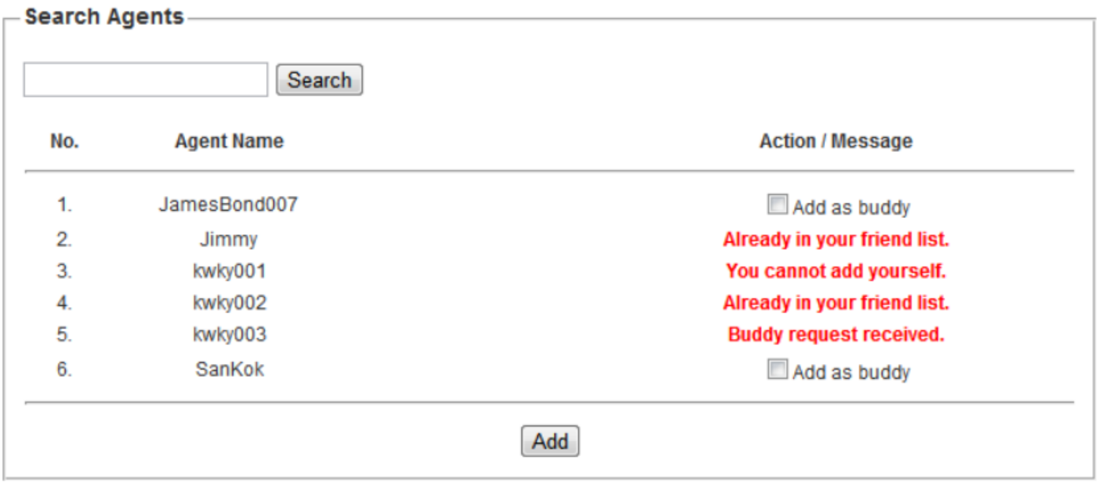
Appendix A-8 Website-based Add Friend in Old UniCAT



New Buddy Requests

No.	Name	Agent Name	Action
1.	kimman3	kwky003	○ Approve ● Pending ○ Reject

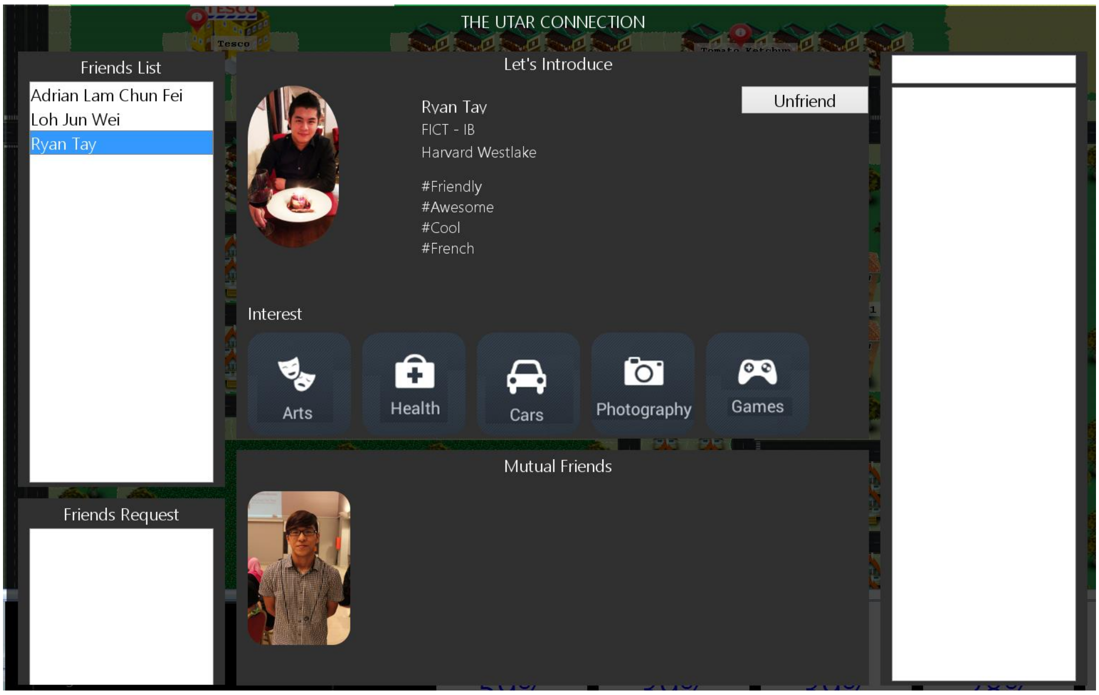
Appendix A-9 Website-based Friend Request in Old UniCAT



Search Agents

No.	Agent Name	Action / Message
1.	JamesBond007	□ Add as buddy
2.	Jimmy	Already in your friend list.
3.	kwky001	You cannot add yourself.
4.	kwky002	Already in your friend list.
5.	kwky003	Buddy request received.
6.	SanKok	□ Add as buddy

Appendix A-10 Website-based Search Friend in Old UniCAT



Adrian Lam Chun Fei
Loh Jun Wei
Ryan Tay

Ryan Tay
FICT - IB
Harvard Westlake

#Friendly
#Awesome
#Cool
#French

Appendix A-11 Friend Popup Windows in Reengineered UniCAT