

Information Retrieval Using Mobile Agent (Aglet)

by

Ma Sochheat

Dissertation submitted in partial fulfilment of
the requirements for the
Bachelor of Technology (Hons)
(Information & Communication Technology)

JULY 2006

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

6

QA

76.9

.03

5678

- 1) Database management
- 2) Data mining
- 3) Intelligent agents (computer software)

CERTIFICATION OF APPROVAL

Information Retrieval Using Mobile Agent (Aglet)

by

Ma Sochheat

A project dissertation submitted to the
Information Technology Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
BACHELOR OF TECHNOLOGY (Hons)
(INFORMATION & COMMUNICATION TECHNOLOGY)

Approved by,



(Mr. Mohd Hilmi Hasan)

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

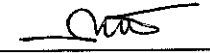
July 2006

TABLE OF CONTENTS

CERTIFICATION		i
ABSTRACT		ii
ACKNOWLEDGEMENT		iii
CHAPTER 1:	INTRODUCTION	1
	1.1 Background of Study	1
	1.2 Problem Statement	5
	1.3 Objectives and Scope of Study	6
CHAPTER 2:	THEORY	8
	2.1 Aglet Architecture	9
	2.2 Aglet API	9
	2.3 Aglet Life Cycle	10
	2.4 Aglets Security and Communication	11
	2.5 Agent Transport Protocol	11
	2.6 Database management	11
	2.7 Benefit of Mobile Agent	12
CHAPTER 3:	METHODOLOGY	13
	3.1 Software process model	5
	3.1.1 Planning	14
	3.1.2 Analysis	14
	3.1.2.1 Tahiti function.	19
	3.1.2.2 System architecture.	19
	3.1.2.2 ATP	19
	3.1.3 Design	21
	3.1.3.1 Use Case Diagram	22
	3.1.3.2 Class Diagram.	23
	3.1.3.3 Sequence Diagram	23
	3.1.4 Implementation.	24
	3.1.5 Testing.	24
	3.2 Tools require	24
CHAPTER 4:	RESULTS AND DISCUSSION	25
	4.1 Finding	25
	4.2 How the system works	25
	4.3 Aglet Life Cycle	26
	4.3 Discussion	27
CHAPTER 5:	CONCLUSION AND RECOMMENDATION	29
	5.1 Conclusion	29
	5.2 Recommendations	29
REFERENCES		30
APPENDICES		31

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.



MA SOCHHEAT

ABSTRACT

This project will present 3 objective in HPS (Hand Phone System) that help user to search for hand phone detail information by: [1] *Brand such as Nokia, Samsung, and Sony Ericsson.* [2] *Version of the hand phone (model).* [3] *Price:* User can input the price range.

Instead of having to surf endlessly through the WWW (World Wide Web) and digesting huge amounts of possibly untrustworthy information, personalized mobile agents autonomously gather information about the item user want to see. Once they have found the destination or server (the piece of software); the agent can send the result back to the user. This project will be focusing on Information Retrieval and will be using mobile agent platform which is *aglets, Java runtime, Jdk1.0.5_08, and Microsoft Visual J++ 6.0* to design the GUI that build using Applet.

This project will undergo 5 stages which are; plan the hardware or software required for the project, then analyze and get the Product Requirement, next is design the process of implementing HPS System, implement the project by using mobile agent platform – aglets that involves programming activity and last is to test the product to ensure the product successfully completed and testify and fulfill the objectives of this project. The result and discussion about the system produced will highlights how the system works generally. Some other interested formula and idea from other project's discussion will also be used to further doing this project. As a conclusion, is project is aimed to create a simple HPS system using mobile agent technology and need to be improved in many aspects and that can be the recommendation for future work for expansion and continuation.

ACKNOWLEDGEMENT

I would like to thank Mr. Anang Hudaya Muhamad Amin who is my supervisor for the first part and he really gives me a lot of supports and guidance to do the project. He provides me the main resource which is the Aglet Programming book and shows me how to set up the server etc. He also helps me in the design stage and some of the coding in Aglet.

I also would like to give thank to Mr. Mohd Hilmi Hassan who is my supervisor for the second part and he really support me along the process of completing the project and a lot of comments on the report format so that it is follow the university format. I am really thank for his time that he spend with me so that I can finish it in time.

I would like to give credit to the Unniversiti Teknologi PETRONAS as well which provide a lot of resources for the student to do the research such as a very big library and the internet access in the lab or from the hostel.

LIST OF FIGURES

Figure 1: The Aglet Architecture

Figure 2: The Aglet API

Figure 3: Aglet life cycle

Figure 4: The selected methodology model

Figure 5: The command prompt window to run the agletsd (Tahiti Server)

Figure 6: Installing the platform

Figure 7: Login to the Tahiti server

Figure 8: creating new account

Figure 9: The Tahiti Main Window

Figure 10: Interface to create an agent

Figure 11: Result of creating agent

Figure 12: Dispose an agent

Figure 13: HPS system architecture

Figure 14: Agent Transfer Protocol

Figure 15: Use Case diagram

Figure 16: Sequence diagram

Figure 17: Class diagram

Figure 18: Agent Life Cycle

Figure 19: Mobile agents approach configuration (message variation)

LIST OF TABLES

Table1: Agent Life Cycle Description

CHAPPER 1

INTRODUCTION

1. INTRODUCTION

This introduction will explain the fundamental information of the project, which consists of an overview of the project, background of study of Aglet, advantages of mobile agent, problem statement(s), objective(s) and scope of study.

Overview of the project:

This project will use the Mobile agent technology (aglet) to implement the web-based application which user can search the information about the mobile phone. User will submit the requested information to an agent, and then the agent will carry the request to the server (Tahiti) and find the result in the database (Mysql) through the JDBC connection. The result will be sent back to the user after the retrieving is completed.

1.1 Background of study:

Aglets are Java objects that can move from one host on the network to another. That is, an aglet that executes on one host can suddenly halt execution, dispatch to a remote host, and start executing again. When the aglet moves, it takes along its program code as well as the states of all the objects it is carrying. A built-in security mechanism [5] makes it safe to host untrusted aglets. There are some main advantages of aglets such as:

- Provide an easy and comprehensive model for programming mobile agents without requiring modifications to Java VM or native code.
- Support dynamic and powerful communication that enables agents to communicate with unknown agents as well as well-known agents.
- Design a reusable and extensible architecture.
- Design a harmonious architecture with existing Web/Java technology.

There are many applications for which mobile agents are claimed to be well-suited. Most of these applications tend to involve searching for information on behalf of a user and possibly performing some kind of transaction when appropriate information is encountered.

Here is a list of some of the more commonly mentioned applications for mobile agents:

Data collection from many places

One of the main differences between mobile code, such as applets, and mobile agents is itinerary. Whereas mobile code usually travels just from point A to point B, mobile agents have an itinerary and can travel sequentially to many sites. One natural application of mobile agents, therefore, is collecting information spread across many computers hooked to a network.

An example of this kind of application is a network backup tool that periodically must look at every disk attached to every computer hooked to a network. Here, a mobile agent could roam the network, collecting information about the backup status of each disk. It could then return to its point of origin and make a report.

Searching and filtering

Given the ever increasing amount of information available on the Internet and other networks, the activity of collecting information from a network often amounts to searching through vast amounts of data for a few relevant pieces of information. Filtering out the irrelevant information can be a very time-consuming and frustrating

process. On behalf of a user, a mobile agent could visit many sites, search through the information available at each site, and build an index of links to pieces of information that match a search criterion [5].

Searching and filtering exhibits an attribute common to many potential applications of mobile agents: knowledge of user preferences. Although mobile agents do not have to be "representative" or "intelligent," they often are. Here, an agent is given knowledge of user preferences in terms of a search criterion and an itinerary, and sent out into the network on the user's behalf. It sifts through huge amounts of data for those pieces of information of particular interest to the user. At some point, it returns to the user to report its findings.

Monitoring

Sometimes information is not spread out across space (on the disks of many different computers hooked to the same network), but across time. New information constantly is being produced and published on the network. Agents can be sent out to wait for certain kinds of information to become available [5].

For example, an agent could go to a stock market host, wait for a certain stock to hit a certain price, then buy or sell some of it on behalf of its user. Another example is personalized news gathering. An agent could monitor various sources of news for particular kinds of information of interest to its user, and then report back when relevant information becomes available.

This kind of application highlights the asynchronous nature of mobile agents. If we send out an agent, we don't need to sit and wait for the results of its information gathering. We can program an agent to wait as long as it takes for certain information to become available. Also, we needn't stay connected to the network until an agent returns. An agent can wait until we reconnect to the network before making its report to us.

Targeted information dissemination

Another potential use of mobile agents is to distribute interactive news or advertising to interested parties (internet push model) [5]. Unfortunately, this means mobile agents, like e-mail, can be used for spam -- indiscriminate distribution of information, usually advertising. (There will likely be a market for agents that filter the spam out of incoming mobile agents.)

Besides searching databases and files, agents can gain information by interacting with other agents. If, for example, we want to schedule a meeting with several other people, we could send a mobile agent to interact with the representative agents of each of the people we want to invite to our meeting. The agents could negotiate and establish a meeting time.

In this case, each agent contains information about its user's schedule. To agree upon a meeting time, the agents exchange information.

Bartering

Electronic commerce is another good fit for mobile agent technology [5]. A mobile agent could do our shopping for us, including making orders and potentially even paying. For example, if we wanted to fly from Silicon Valley to an island in the South Pacific, an agent could visit databases of flight schedules and prices for various airlines, find the best price and time, make reservations for us, and pay with our credit card number.

Electronic commerce also can take place between agents. For example, there could be an agent host dedicated to the buying and selling of automobiles. If we wanted to buy a car, we could give an agent knowledge of our preferences, including a price range and potentially a negotiation strategy. We would send our agent to the dedicated host, where it would mingle and haggle with agents seeking to sell a car.

If a potential match were found, our agent could report back to us, and we could contact each other in person to make the final arrangements. Alternatively, our agent potentially could consummate the deal on our behalf. If the opportunity is a good

one, our agent may have only a few microseconds to act before someone else's agent buys the car.

Parallel processing

Given that mobile agents can move from node to node and can spawn subagents, one potential use of mobile agent technology is as a way to administer a parallel processing job [5]. If a computation requires so much CPU time as to require breaking up across multiple processors, an infrastructure of mobile agent hosts could be an easy way to get the processes out there.

Entertainment

One last example of a potential application for mobile agents is entertainment. In this scenario, agents represent game players. The agents compete with one another on behalf of the players. Each player would program an agent with a strategy, then send the agent to a game host. If the game host is executing on a computer in Las Vegas, then perhaps the agents could play for real money.

1.2 Problem Statement

Before the users decide to buy any Hand phone, I believe they want to make sure that it is what they want in terms of price, brand, and the properties of the hand phone. So it is very important that users can get the right hand phone with their own selected criteria. Mobile agents autonomously gather information about the item user want to buy. Once they have found the destination or server, the agents can send the result of the best requested item on behalf of the user. *This project is focus more on how the agents communicate with the user on the application layer and talk to the server or other agent to find out the result as response to the user (Data retrieval using agent).*

1.2.1 Significant of the project

Reduce communication time and costs – because of the mobile agent technology it is enable the agent to communicate with other agent even in the different network [5] so it is save a lot of time for the user to wait for the result.

Limited local resources – the processing power and storage on the local machine may be very limited, thereby necessitating the use of mobile agents.

Asynchronous computing [5]– after the request is submitted, the mobile agent can be set off and the result will send to the user automatically once the user connected.

1.3 Objectives and Scope of Study

The objectives of the project are:

- User can brows to see the provided version of hand phone
- Help the user to experience the capability of Mobile Agent when the information being requested
- Help user to get the result faster
- Provide user more option

This *Hand Phone System (HPS)* will help user to search for hand phone detail information by selecting the criteria that they like:

I. Brand

- a. Nokia
- b. Samsung
- c. Sony Ericsson

II. Type of the hand phone (model)

III. Price

- a. User can input the price range

1.3.1 The Relevancy of the project

This project is relevant to the organization that handles business information online and serving their customer visas the technology. This project convinced that the use of mobile agent technology is very interesting, especially in the telecommunication and business domain where this technology can bring solutions that are suitable to this sector. Intelligent agent can be used to reduce the complexity of the business process. The works in intelligent networks area allowed separating services from the network itself, which bring more flexibility and improvement to the network. Here, the main focused of an agent is mobility. The other ones such as intelligence, negotiation and communication between agents are also important.

1.3.2 Feasibility of the Project within the Scope

This project is feasible by using *mobile agent platform – aglets-2.0.2.jar*, *Java runtime (JRE)*, and *Java 2 Software Development Kit – JDK1.5.0_08*, *Visual J++ 6.0.*, *BlueJ Java editor*. There are five (5) stages to complete the project successfully which are:

- Plan the hardware or software required for the project.
- Analyze and get Product Requirement
- Design the process of implementing the system
- Implement the project by using mobile agent platform – aglets that involves programming activity
- Test the product to ensure the product successfully completed and fulfill the objectives.

CHAPTER 2

LITERATURE REVIEW

2. LITERATURE REVIEW AND/OR THEORY

Aglets is one of the Mobile Agent System and the reason is: Mobile Agents are intelligent autonomous programs that can travel across the heterogeneous network [5] in order to perform an assigned task. It is one of the popular and simpler ways of retrieving information from the Internet.

Aglets are fundamentally Java-based autonomous software mobile agents. An aglet carries its state and as well as data along with it while traveling across the network. The basic idea is create once, go anywhere.

There are a lot of significances of Mobile Agents such as:

- Reduces Network Traffic [1]
- High Scalability
- Asynchronous processing [1]
- Support for heterogeneous environments [1]

ASDK, developed by IBM, is a Java-based framework for implementing mobile agents called aglets. It provides a network agent class loader that enables mobility of agent code, data and state information. The freeware open source IBM aglets package version of 2.0.2 can be downloaded from the Sourceforge.net or the IBM website.

2.1 Aglet Architecture

Aglets architecture consists of two APIs and two implementation layers [6].

- Java Aglet API
- Aglets Runtime Layer – The implementation of Aglet API
- Agent Transport and Communication Interface (ATCI with ATP as an application-level protocol)
- Transport Layer

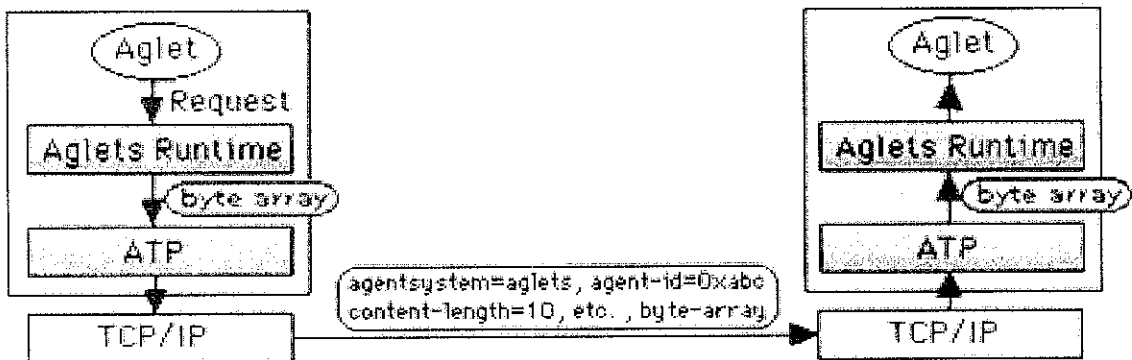


Figure 1: The Aglet Architecture

2.2 Aglet API

Internet agent developers can develop platform independent aglets written in Java programming language and expect them to run on any host that supports Aglet API [6].

- Aglet: Provides methods that control the mobility and lifecycle of an aglet.
- Aglet Context: Provides the execution environment at the remote site.
- Aglet Proxy: Provides a handle that is used to access the aglet.
- Message: An object exchanged between aglets.

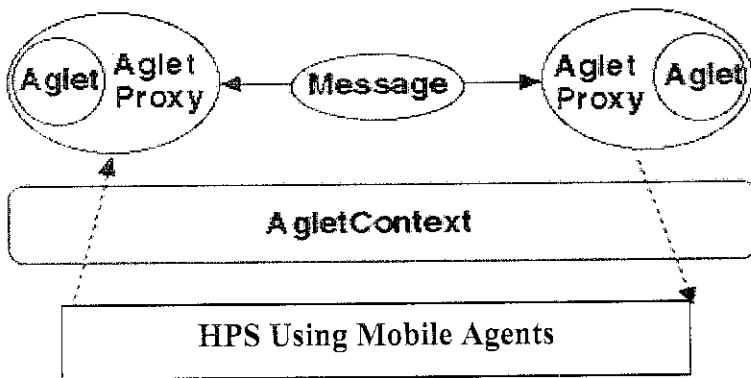


Figure 2: The Aglet API

2.3 Aglet Life Cycle

Different stages in an aglet execution are as shown in Figure 3.

Aglets can be: *created, cloned, dispatched, retracted, deactivated, activated, disposed* [5].

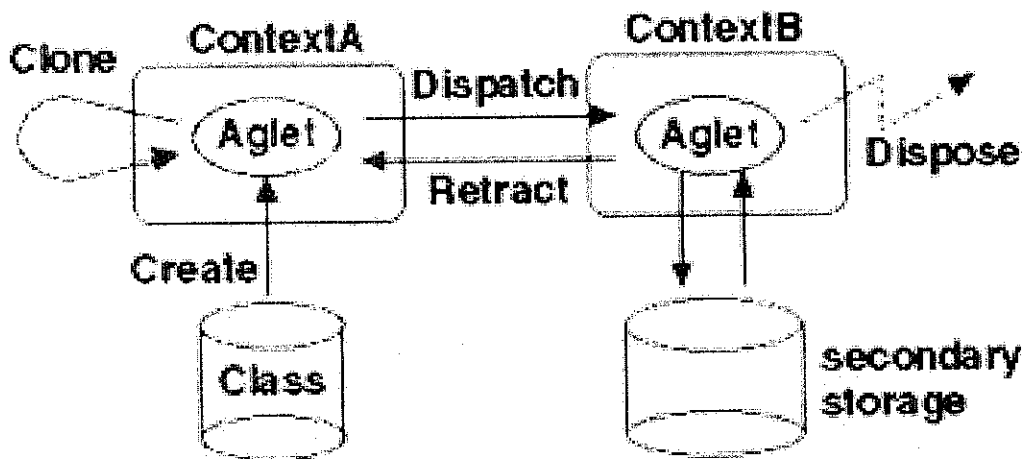


Figure 3: Aglet life cycle

2.4 Aglets Security and Communication

- Security in Aglets
- Security is a prime concern for mobile agent technology, and aglets provide an extensible security model in the form of an `AgletSecurityManager`, as a subclass of the Java Security Manager.
- Aglets Communication
- Messaging between aglets involves *sending*, *receiving*, and *handling* messages synchronously as well as asynchronously. Aglets communicate with each other by exchanging Message objects [5].

2.5 Agent Transport Protocol

The *Agent Transfer Protocol* (ATP) is an application-level protocol for distributed agent-based systems. ATP offers a simple and platform independent protocol for transferring agents between networked computers.

ATP offers the opportunity to handle mobility of agents in a general and uniform way. ATP/0.1 is implemented in the `atp-package` as a part of the *IBM Aglets Software Development Kit* [6].

2.6 Database management

On each database server, the aglet can be engaged in heavy data retrieval without suffering from network latency [1], since the database is accessed locally at a very high bandwidth. The client can be disconnected from the network while the information gathering takes place.

The database structure and server topology are not revealed to the network. Highly scalable[6], since providing the database structure of the database residing on a remote host and including the remote host address in the travel plan list of the aglet, would suffice the need of addition of new remote hosts to gather more information for the user.

2.7 The Advantages and Benefits of Mobile agents

The main advantage of mobile agents is that they can bring a program closer to the information resources. The mobile agent paradigm stipulates that the server should provide set of basic services. The client uses the services provided by the server by dispatching a program that is a mobile agent, to the server. The mobile agent makes use of the server's basic services, in the way that its owner intends. Besides that, there are many advantages that have been discovered for mobile agent technology such as; to overcome network latency, reduce design risks, reduce network load, can be adapted dynamically as well as can execute asynchronously and autonomously.

Advantages from mobile agent paradigm [1] & [5]:

- Reduce bandwidth consumption and network loads
- Allow dynamic deployment of application components to arbitrary network sites
- Encapsulate protocols
- Execute asynchronously and autonomously
- Can adapt by moving
- Run on heterogeneous platforms
- Most distributed applications fit naturally into the mobile agent model
- Intuitively suitable for mobile users and disconnected operation
- Mobile agents combine the strengths of techniques such as RPC, java servlets etc. into a single, convenient framework.

CHAPTER 3

ANALYSIS AND DESIGN METHODOLOGY

3. ANALYSIS AND DESIGN METHODOLOGY

3.1 Software process model:

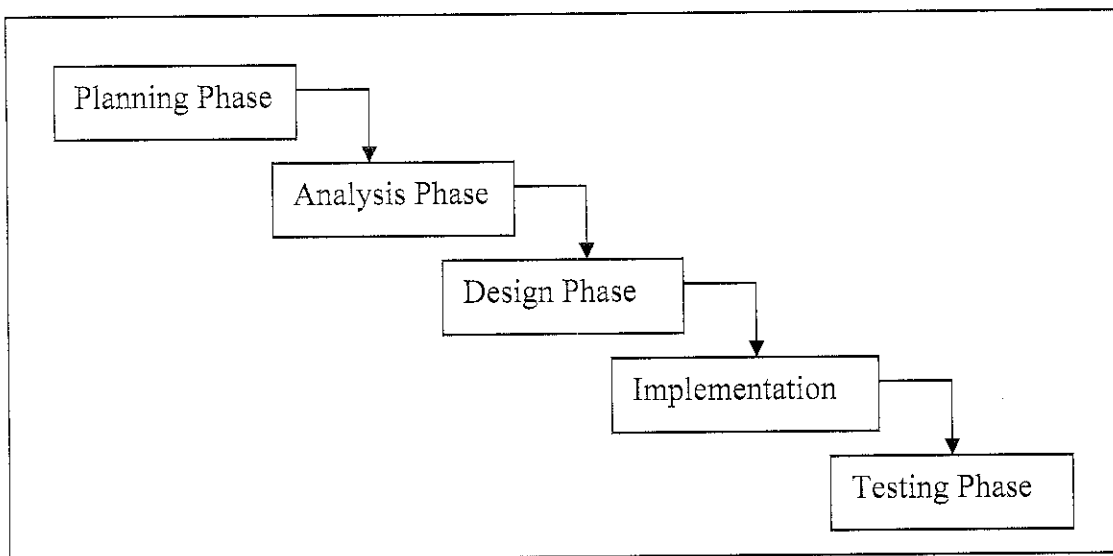


Figure 4: The selected methodology model

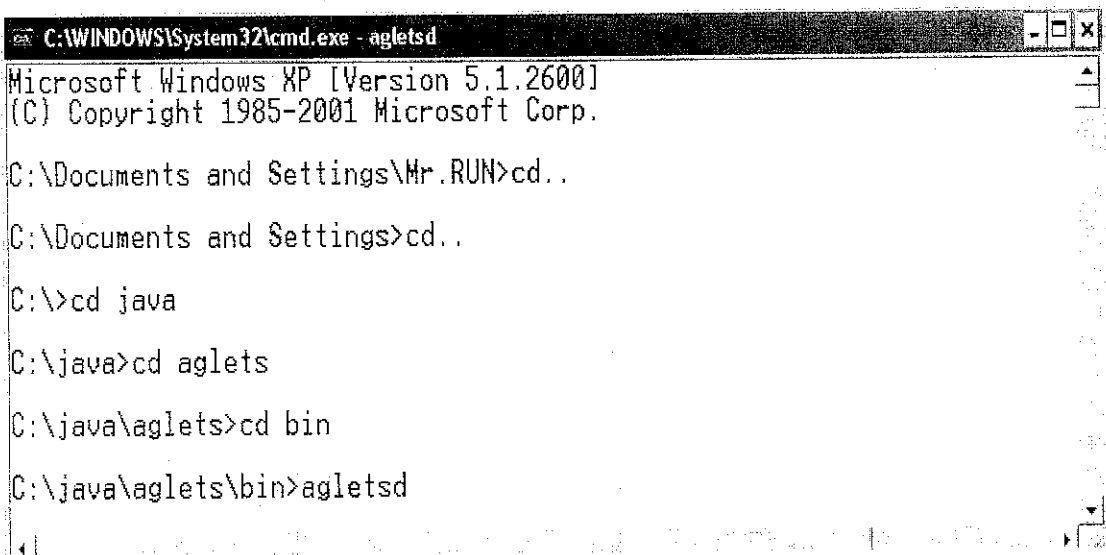
- Plan the hardware or software required for the project.
- Analyze and get Product Requirement
- Design the process of implementing the system
- Implement the project by using mobile agent platform – aglets that involves programming activity
- Test the product to ensure the product successfully completed and fulfill the objectives.

3.1.1 Planning Phase

The planning phase is the initial step in producing this “*Hand Phone System (HPS)*” project. First and foremost, it will be comprised a selection of project title and followed by proposal submission, where in this phase, project scope, objective(s) as well as problem statement(s) are well defined. Then this project will come out with preliminary research, progress report and also produce the dissertation draft before submitting the final dissertation report. At this stage, the main thing to be considered is the hardware and software requirement needed to undergo this project. The title has been analyzed and it requires using the mobile agent technology which allowed the uses of Java Aglet which comprises a java programming language. Therefore, here it is important to know better how the java programming language and its technology works. In addition, the Aglet is a new technology and to further understand how its work, the preliminary study should be done as well the guidance from the expertise should helped more.

3.1.2 Analysis Phase

This second phase will be the analyzing of literature reviews related too the proposed topic, research and study on the software been used, followed by producing a progress report. The major part for this phase is to analyze and get the product requirements. Here, this project will continue with development of UML diagram to further understand the flow of the project and system. The analysis on client, server, database and user interface should be done in this phase before continuing to the next phase. During the analysis phase, so many problems occur in running the Aglet server. Such as the wrong setting path of the java and Aglet itself in the PC, as well as the decompression of the Aglet-2.0.2.jar file cannot be accomplish successfully. However, the server can be used smoothly after fix it together with the expert, although the task only successfully managed behind the project schedule.

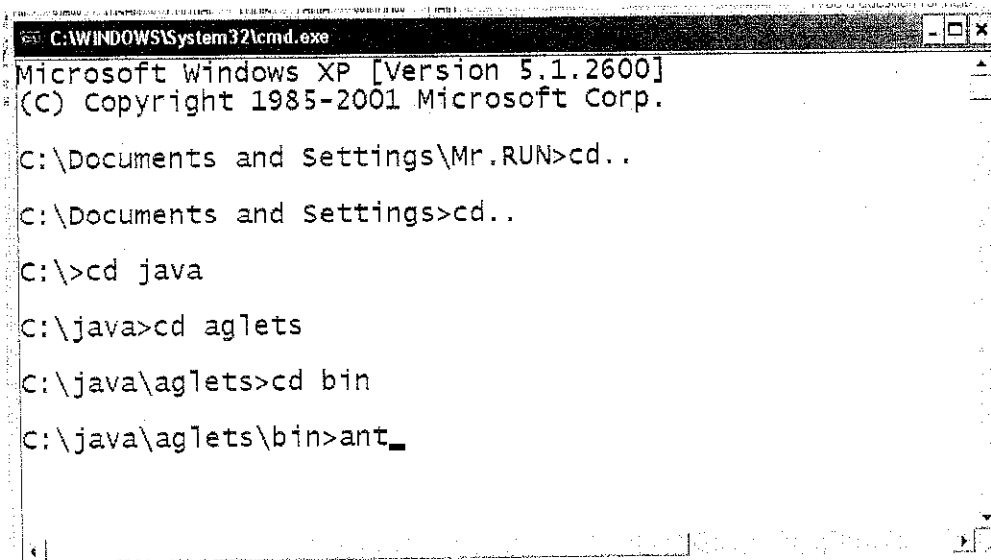


```
C:\WINDOWS\System32\cmd.exe - agletsd
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Mr.RUN>cd..
C:\Documents and Settings>cd..
C:\>cd java
C:\java>cd aglets
C:\java\aglets>cd bin
C:\java\aglets\bin>agletsd
```

Figure 5: The command prompt window to run the agletsd (Tahiti Server)

Before the Tahiti server can be run successfully by using **agletsd** command first we need to create the path in C:\java\aglets then we extract the Aglets-2.0.2.jar inside the aglets folder, then we run command prompt window by going to C:\java\aglets\bin and run ant command to install the platform (Figure 6). After it build successfully, it will create 6 more files in bin folder which are *java.polecy*, *agletsd*, *.keystore*, *agletd.bat*, *daemoncontrol*, *daemoncontrol.bat*. Then we need to copy *java.polecy*, *agletsd*, *.keystore* and put into C:\document and setting\(*your window profile folder*)



```
C:\WINDOWS\System32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Mr.RUN>cd..
C:\Documents and Settings>cd..
C:\>cd java
C:\java>cd aglets
C:\java\aglets>cd bin
C:\java\aglets\bin>ant_
```

Figure 6: Installing the platform

After the Tahiti server is running successfully, we can use default user id and password which is *anonymous* and *aglets* to login. See figure 7.

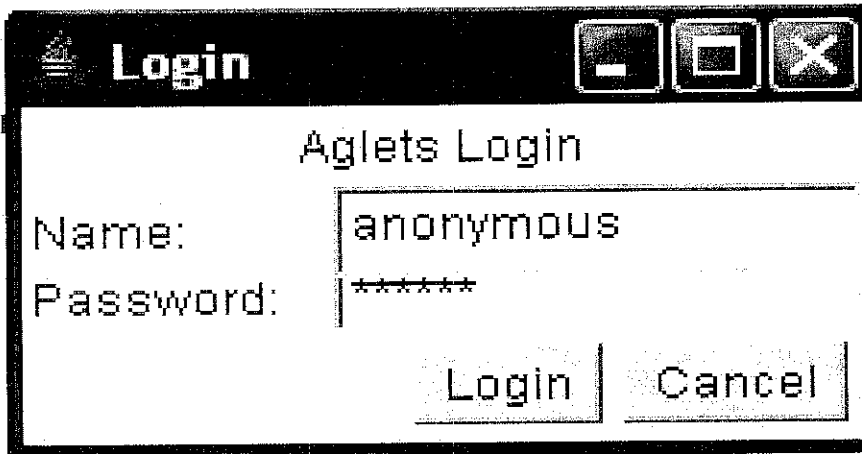


Figure 7: Login to the Tahiti server

We can create new account by using *keytool* command but before that we need to copy *keytool.exe* from *C:\jdk1.4.2_07\bin* and past into *C:\java\aglets\bin* first.

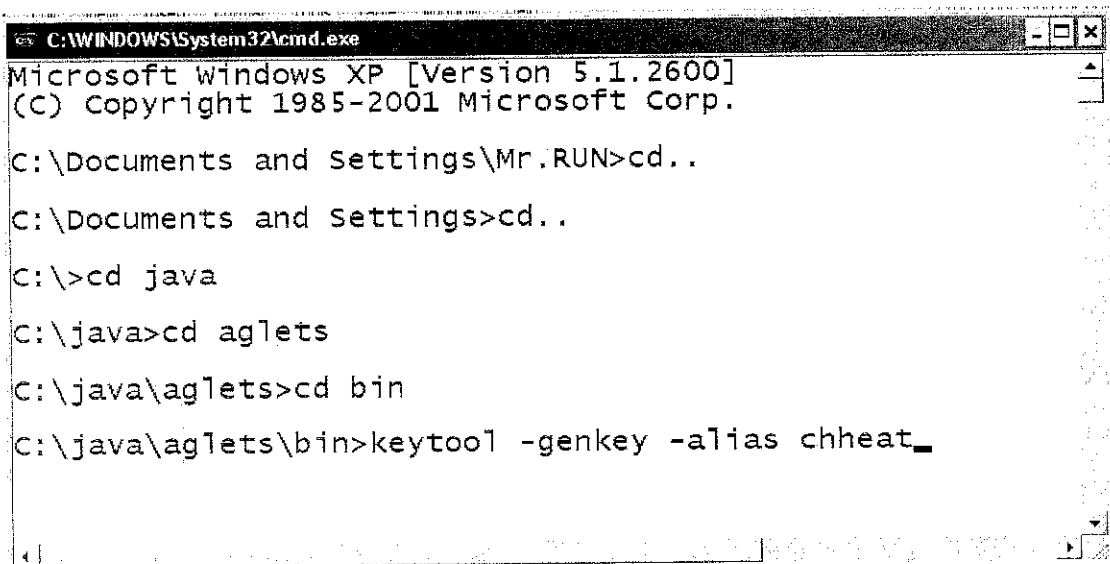


Figure 8: creating new account

After we run the above command, we need answer some question and enter the password for user cheat.

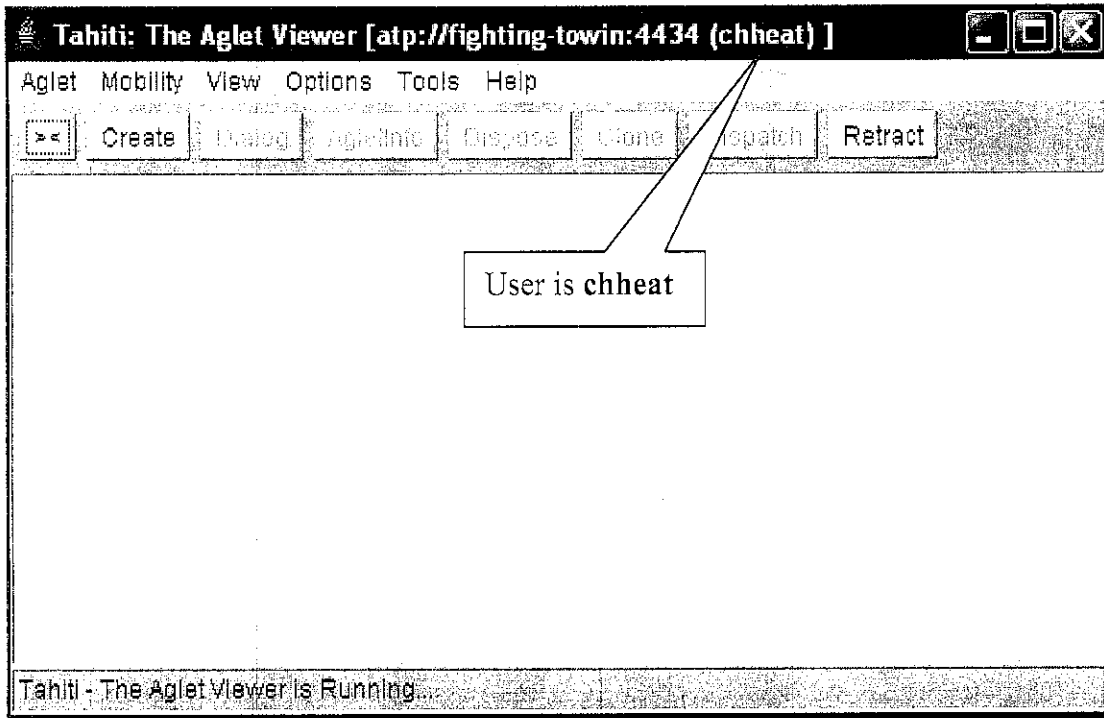


Figure 9: The Tahiti Main Window

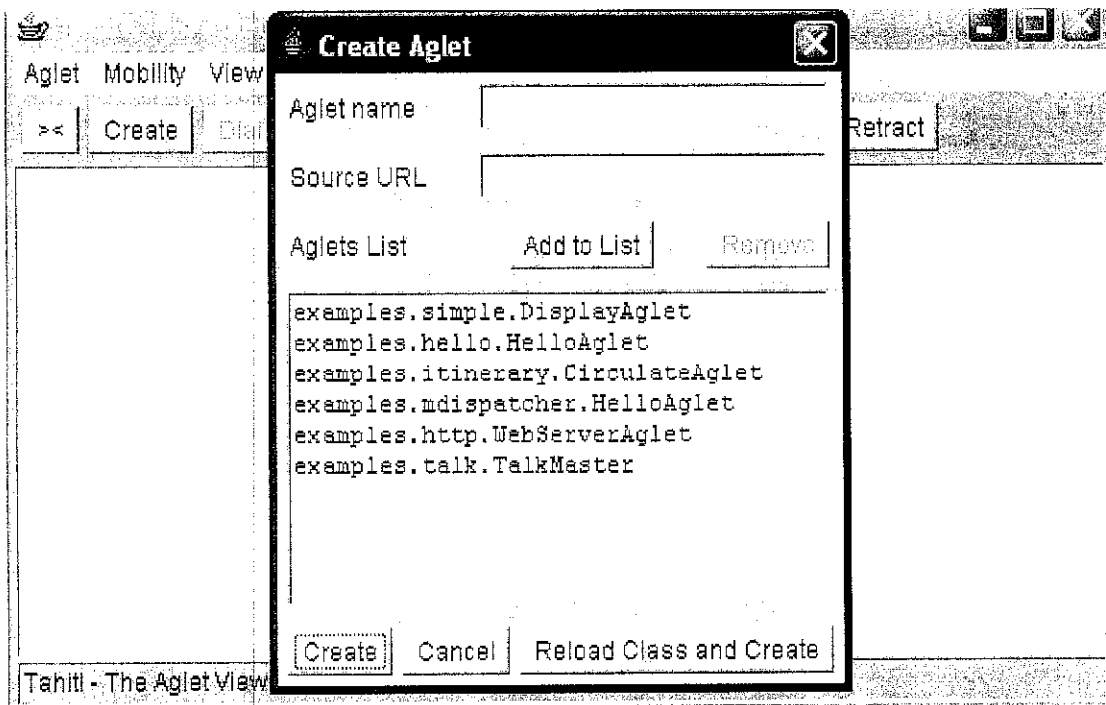


Figure 10: Interface to create an agent

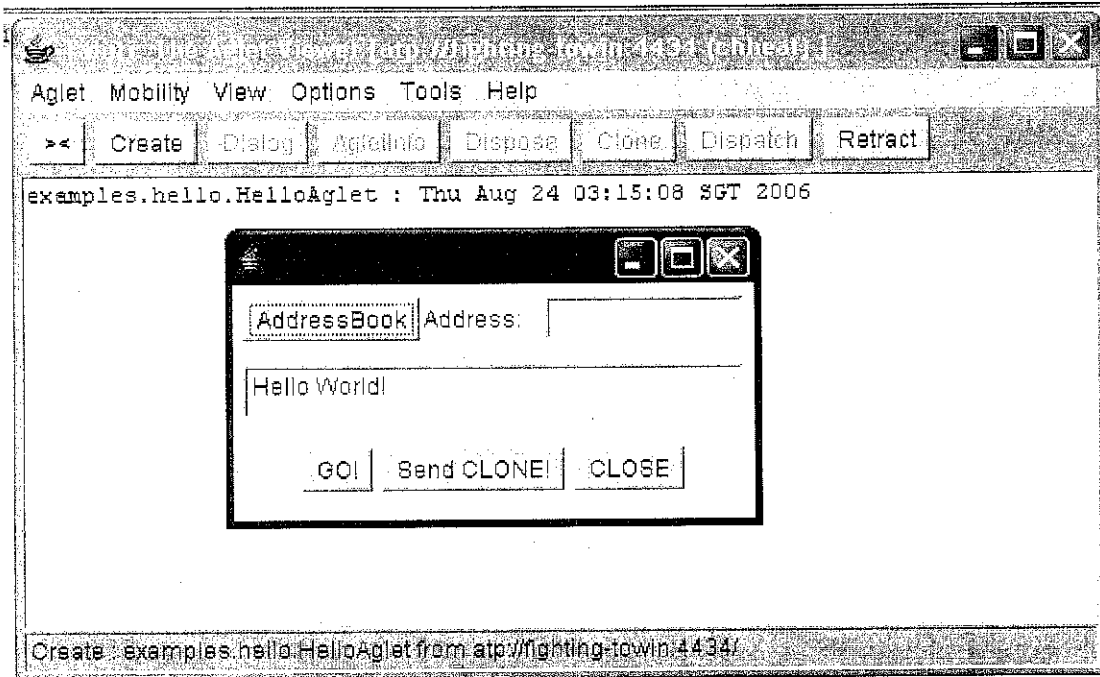


Figure 11: Result of creating agent

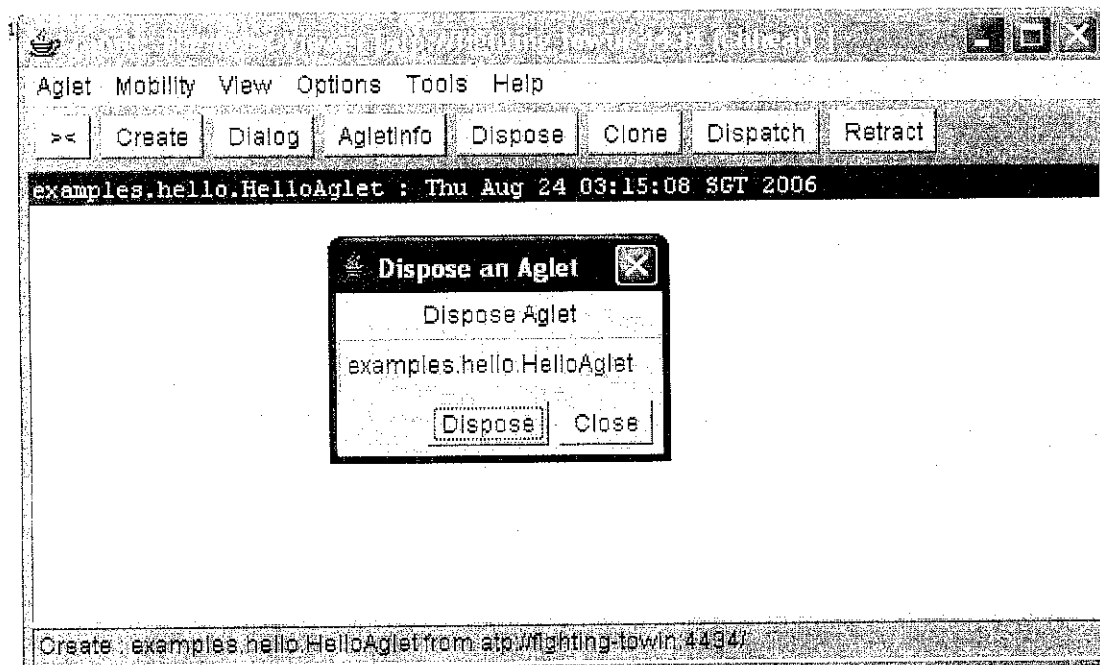


Figure 12: Dispose an agent

3.1.2.1 Tahiti Server Functionality

First of all, as shown in figure 4, the aglets server should be run by executed the command of agletsd. Then, it will appear a Tahiti login window (figure 6) and will ask the user the authenticate itself. If we use the default value, the username will be anonymous and the password will be aglets. In figure 8, I used a new account which is chheat with the new password as well. Figure 8, which is the Tahiti main window will be appear after the user click OK button for login session. In the Tahiti main window, there are 6 buttons operation for an aglet. To create an aglet, user need to “create”, choose the aglet in the list or type the name of the selected program if it is not in the list and “Reload Class and Create”. Every time user modifies and complies the program user need to “Reload Class and Create”. Refer to figure 9 for detail. The figure10 show the result of creating an agent and it also display one dialog box together with the program. Now, users have created their own agent. The agent can be clone and migrate to another place or server by clicking the clone and dispatch operation button in the Tahiti main window. However, figure 11 shows that the agent has been dispose and if the user clicks the “Dispose” button at the display box, the agent will be terminated in the Tahiti server.

3.1.2.2 System architecture

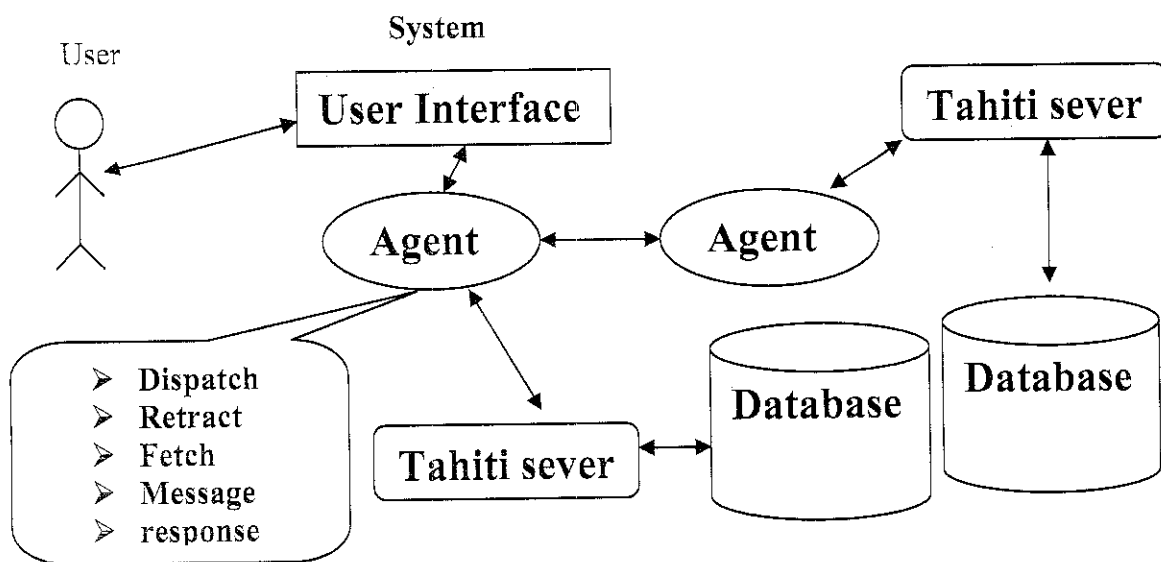


Figure 13: HPS system architecture

Based on figure 2, the first layer is user interact with the interface of the system. User doesn't know how the agent works after they submit the request. The second layer is the agent which needs to communicate with its own server or the other agent so that it can fulfill the user request and send the result back. For this architecture we can see that there are two agents and it has different server and database but these two agents can communicate with each other. We can run these two server on the same machine by specifying different port number used. The database class is reside in the agent and it will be used whenever is needed.

3.1.2.3 ATP (agent transfer protocol):

ATP (agent transfer protocol) is a simple application-level protocol designed to transmit an agent in an agent-system-independent manner. Its request consists of a request line, header fields, and content. The request line specifies the method of the request, while the header fields contain the parameters of the request. ATP defines the following four standard request methods:

Dispatch [5]

The dispatch method requests a destination agent system to reconstruct an agent from the content of a request and start executing the agent. If the request is successful, the sender must terminate the agent and release any resources consumed by it.

Retract [5]

The retract method requests a destination agent system to send the specified agent back to the sender. The receiver is responsible for reconstructing and resuming the agent. If the agent is successfully transferred, the receiver must terminate the agent and release any resources consumed by it.

Fetch [5]

The fetch method is similar to the GET method in HTTP; it requests a receiver to retrieve and send any identified information. (normally class files).

Message [5]

The message method is used to pass a message to agent identified by agent-id and return a reply value in the response. Although the protocol adopts a request/reply form, it does not lay down any rules for communication scheme between agents.

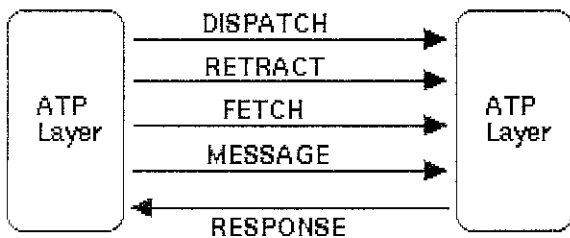


Figure 14: Agent Transfer Protocol

3.1.3 Design Phase

The third phase is design phase, which is design the process of implementing HPS (Hand Phone System). At the previous stage, this project has done the analysis on the idea behind HPS. Therefore, at this stage the design for those output from the analysis are to be drawn in an appropriate diagram or graphical representation. To be reminding, there is no complete design for implementation of the mobile agent technology yet at this stage. However, this design phase only produced the implementation of HPS.

3.1.3.1 Use Case diagram

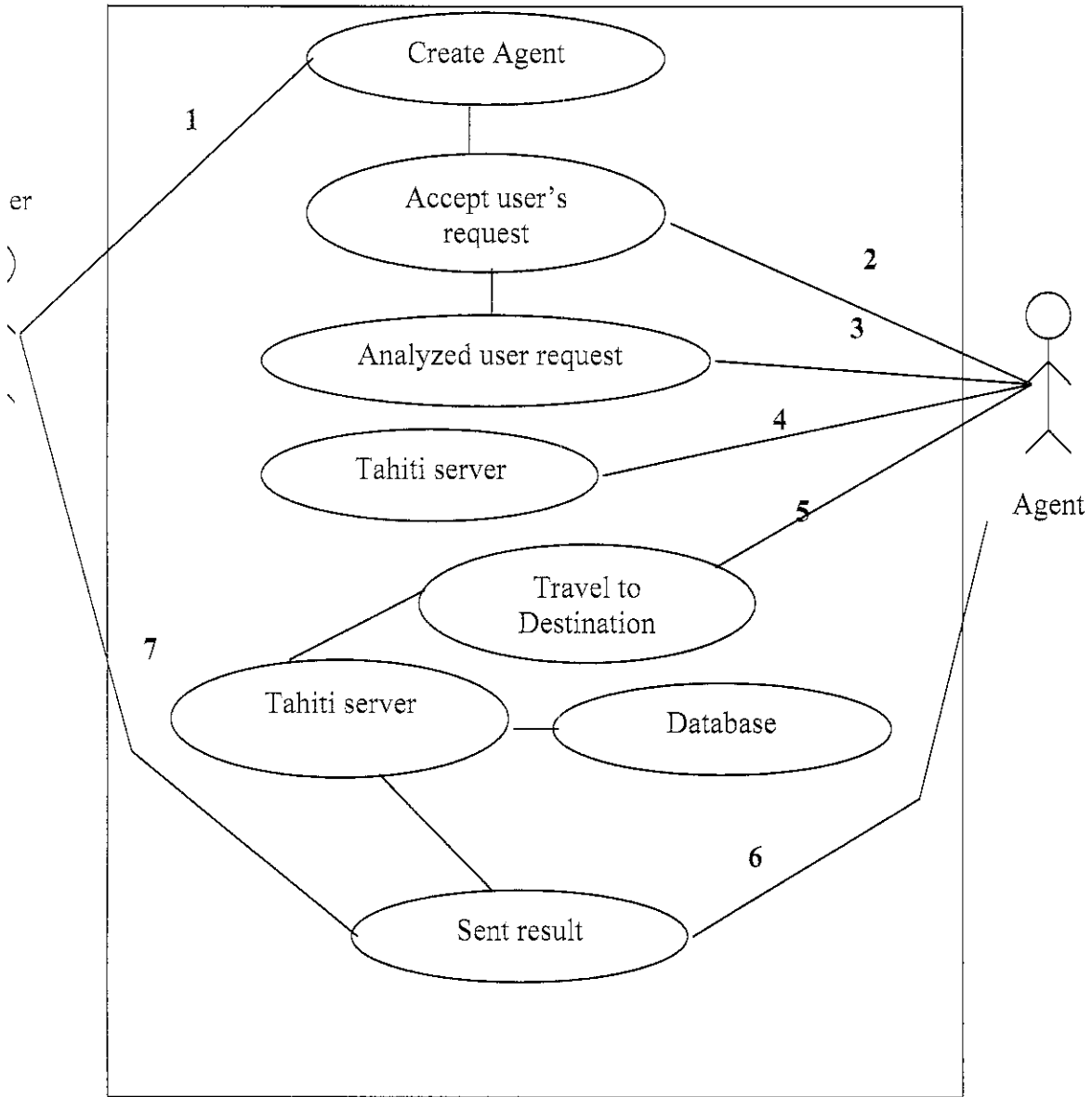


Figure 15: Use Case diagram

3.1.3.2 Sequence diagram

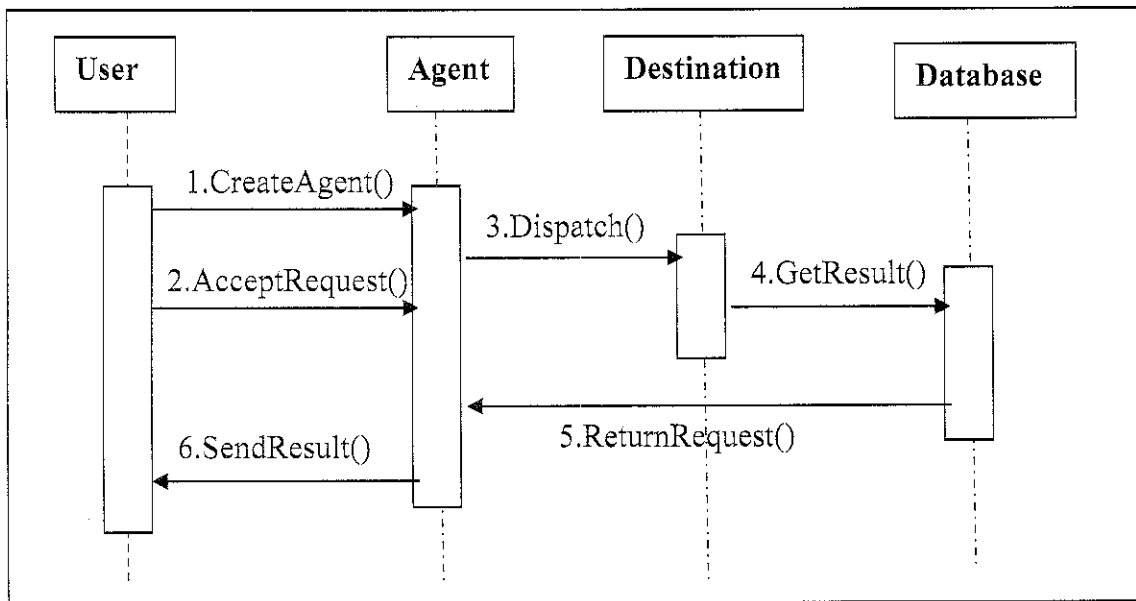


Figure 16: Sequence diagram

3.1.3.3 Class diagram

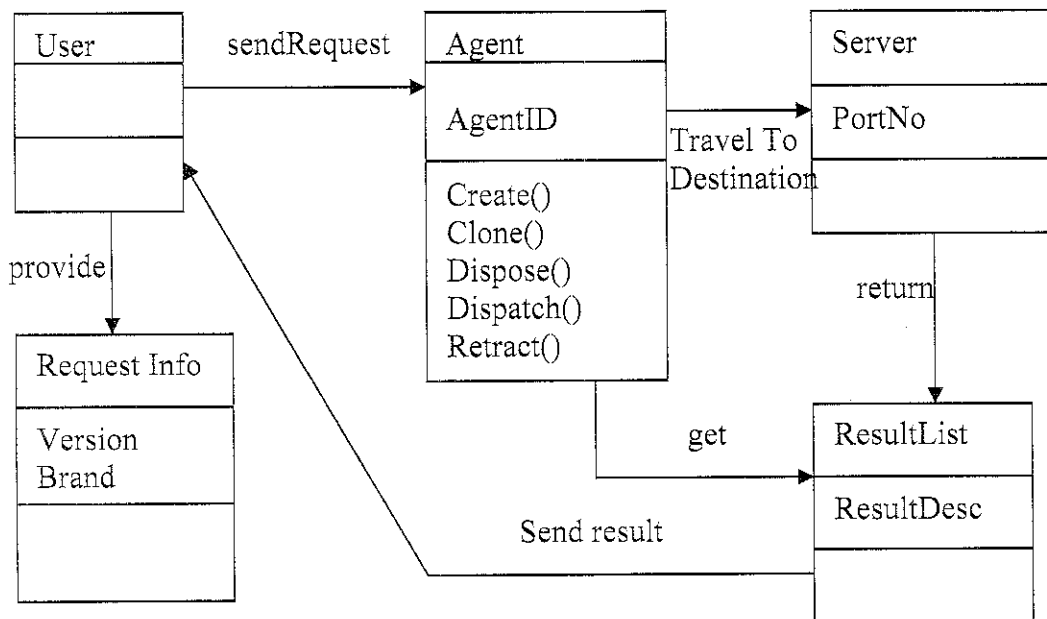


Figure 17: Class diagram

3.1.4 Implementation Phase

The implementation phase is the next phase after design. This phase will come out with the well structure client, server, and user interface design. After that, finalizing all requirements will be done. At this stage, the thorough understanding of java programming language should be adapted tremendously. This is the hardest part in completing the final year project, and need more time to finish it. However, due to the time constraints, the product of the system has been built as simplest as it can long as it meets the objectives of the project. The result of the implementation will be discussed later in the results and discussion session in chapter4.

3.1.5 Testing Phase

For the last phase in this project, all the programming and system testing must be conducted in order to make sure that this project will meet all the objectives and requirements as stated in the report as well as can settle the problem statement that have been defined. This activity involved the evaluation and discussion by the lecturers.

3.2 Tools Require

3.2.1 Software

- mobile agent platform – aglets-2.0.2.jar,
- Java runtime (JRE), and
- Java 2 Software Development Kit – JDK1.0.5_08,
- Visual J++ 6.0

3.2.2 Hardware

- Intel Pentium II processor (or equivalent 300 + MHz processor)
- Microsoft Window NT, 98SE, 2000, or XP
- 64 RAM

CHAPTER 4

RESULTS AND DISCUSSION

4. RESULTS AND DISCUSSION

This chapter will present the outcome of the project work. According to the methodology in Chapter 3, we will see the whole process of the system starting from the user issue the request until it can get the result.

4.1 Finding

The implementation of the HPS (Hand Phone System) using mobile technology really was a challenging for me, but it is quite a time-consuming task. Most of the time I am really stuck with the programming part and I need to refer so many book. Each Problem I need to find one book to look for the answer. So I spend a lot of time to do reading.

4.2 How the HPS Work Generally

First of all the user needs to provide the input in the search engine by giving the mobile version or the Brand of the Hand Phone. Please notice that in this project there are only three brands provided which are Nokia, Samsung, and Sony Ericsson. Once the request is issued, the Agent will accept the request and travel to the destination Tahiti Server to access the Database. DBMS aglet will use JDBC –ODBC to access the DB server. After the data retrieval is completed, the result will send to the user.

Once users key in the version of the Hand Phone, they will get the image and the full specification of the hand such as General, Size, Display, Memory, Feature, and

Battery. Another way is, user can just select the brand of the phone example Nokia then they will get the entire information Nokia version from DB server. The user still can get the result even they are disconnected from the network because the agent still send back the requested information to the user.

4.3 Agent Life Cycle

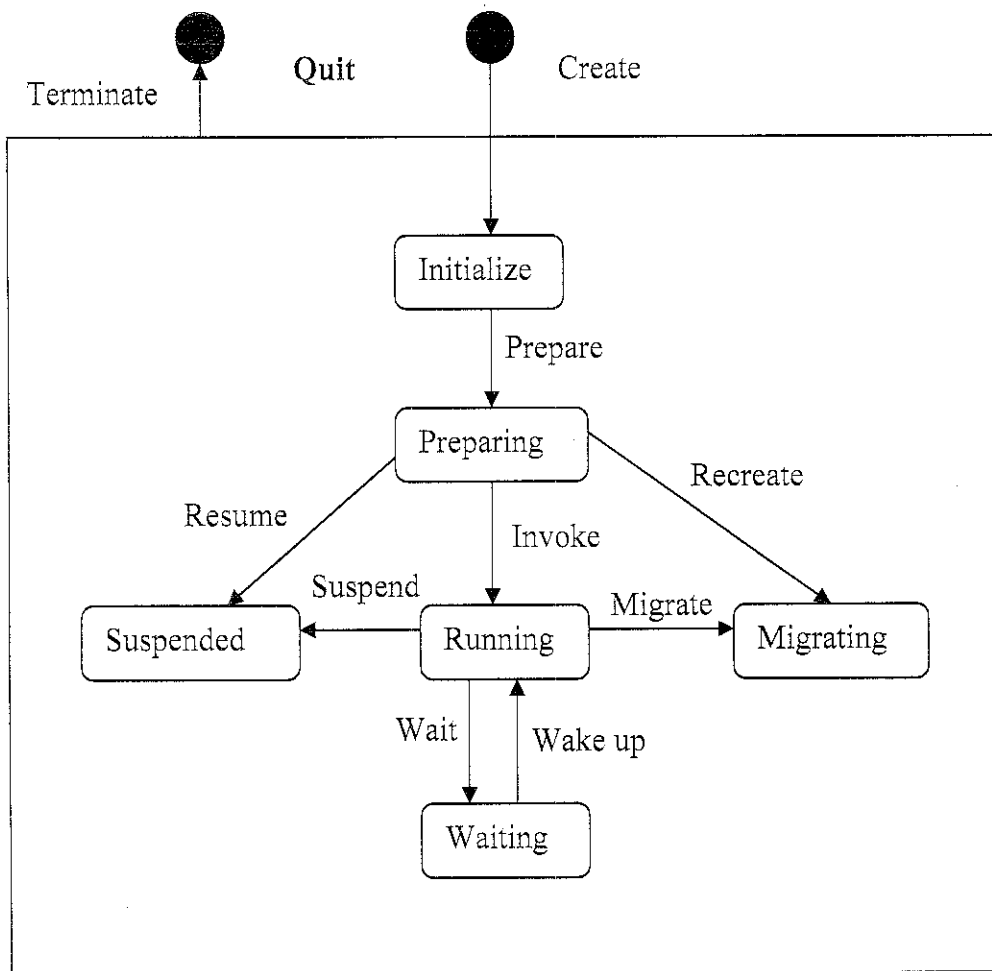


Figure 18: Agent Life Cycle

Create	Creation of a new agent is initiated by the user. Agent's life cycle begins
Prepare	The agent is prepared for action
Invoke	The agent is invoked by the management service. It is fully operational and can make its own decisions.
Suspend	Agent execution is halted temporarily by either the management service or the agent itself
Resume	The agent is resumed from the suspended state
Wait	The agent puts itself into the waiting state
Wake up	The agent is woken up by the management service

Migrate	The agent is serialized and put into migrating state. The decision to migrate can be made by the agent itself.
Recreate	The agent is deSerialized and recreated after migration
Quit	The agent itself chooses to ends its life cycle
Terminate	The management services forces the agent to terminate.

Table1: Agent Life Cycle Description

4.4 Discussion

In this HPS application is develop by using mobile agents to achieve Web database connectivity, and specifically, the best of the three variants proposed. Mobile agents are processes capable of pausing their execution on one machine, dispatching themselves on another machine and resuming their execution on the new machine. The idea in the JMA approach is to use one or more mobile agents to implement the middleware and carry out the requests of the client.

So we used *Aglets*, for two reasons: (a) availability of code, and (b) support for hosting mobile agents within applets without significant overhead based on the prior experience with their use. Aglets can be fired from within a special applet that provides an aglet execution environment similar to the general stand-alone aglet runtime environment called the *Tahiti Server*.

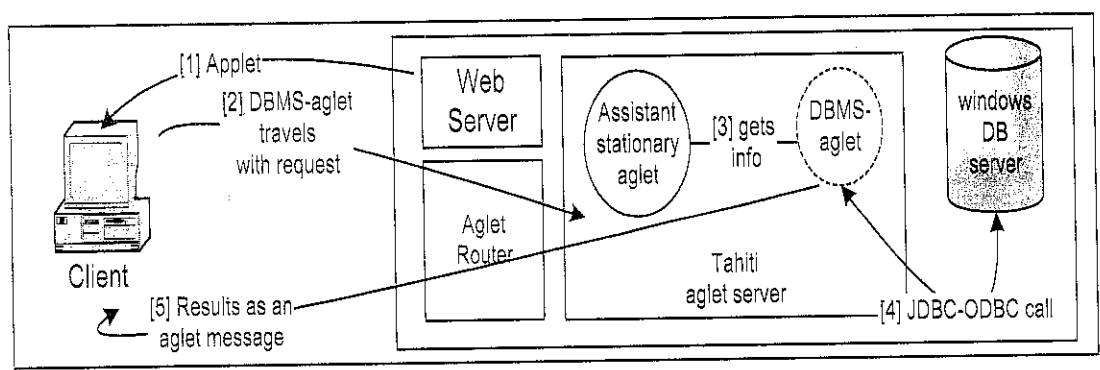


Figure 19: Mobile agents approach configuration (message variation)

In the JMA approach, the middleware consists of three components: The *DBMS-aglet*, the (*Stationary*) *Assistant-aglet* and the *Aglet Router*. The *DBMS-aglet* can connect to a database and submit queries. Each database server is associated with an *Assistant-aglet* identified by a unique aglet ID and a URL. An *Assistant-aglet*

provides the information necessary for a DBMS-aglet to load the appropriate JDBC driver and connect to the database server. An *Aglet Router* is required to route aglets and messages, dispatched from a Applet to any destination, and vice versa, because of the Java security restrictions.

When the user enters the first query (Figure 3), the client applet (an extension of the Applet Class) creates a DBMS-aglet with a specific URL-based itinerary (travel plan) and the specified query. The DBMS-aglet travels through the aglet router to the database server. Upon its arrival, the DBMS-aglet communicates with the Assistant-aglet to retrieve information on the database and drivers, loads the JDBC-ODBC driver, connects to the database server and executes the client's request. After returning the query result in a message to the client, the DBMS-aglet remains connected to the database server, waiting for a message with new requests from the client. This message passing is implemented implicitly as an RPC invocation from the client applet on the dispatched mobile agent. The cost of the initial query is

1. Initialization phase:

- A. The time for the client to create the DBMS-aglet
- B. The time for the client to initialize the DBMS-aglet (SQL statement, itinerary, etc.)
- C. The time for the DBMS-aglet to travel to the remote database server
- D. The time for the DBMS-aglet to negotiate with the assistant aglet
- E. The time for the DBMS-aglet to establish connection with the database

2. Execution phase:

- A. The time for the DBMS-aglet to query the database and send the results to the client using a message.

All subsequent requests require only one message from the client to DBMS-aglet, which includes the new SQL statement, plus the execution phase.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

An Agent characteristic of Java helps to achieve the objective of this project where by the user can get the searching result faster compare to other non-mobile technology. Once the user enter the mobile version or hand phone brand, agent will follow the procedure in the design phase to retrieve the information and send the result back to the user even thought they disconnect from the network.

5.2 RECOMMENDATION

This HPS (Hand Phone System) using mobile agent technology is really new for me and may be to other student as well. Today, the technology grows very fast and it provides the easier way to analyze and design the system plus a lot of benefit especially save time and cost. There are a lot of thing can be enhanced in the system as well as more detail research on the mobile agent technology together with its application. Below here are some of the future work expansion and continuation:

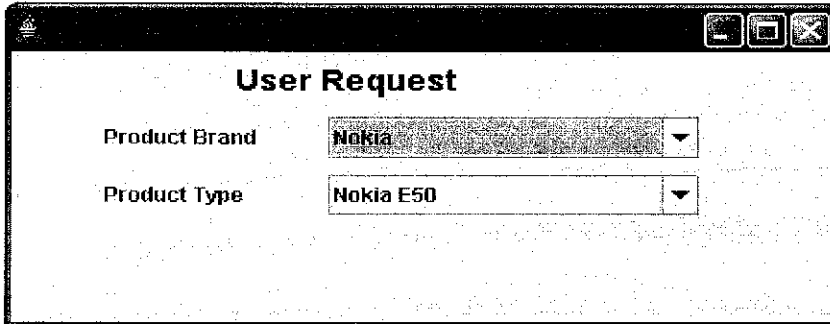
- Develop the same application by using different mobile agent toolkits.
- Research on Distributed Database using Mobile agent
- Develop the same system and host it on the web
- Provide the Download Function for the user to download the ring tone, mobile phone application, multimedia image etc

REFERENCES

- [1] Constructing Intelligent Agents Using Java, Second Edition, Joseph P. Bigus, Jennifer Bigus, 2001
- [2] Java Programming Projects, Activities Workbook, CEP, Inc & Sestak, 2001
- [3] Learn Microsoft Visual J++ 6.0 Now, Kevin Ingalls and Daniel Jinguji, 1998
- [4] JDBC Database Access with Java, Graham Hamilton, Rick Cattell, Maydene Fisher, 1997
- [5] Programming And Deploying Java Mobile Agents With Aglets, Danny B. Lange, Mitsuru Oshima,
- [6] <http://www.trl.ibm.com/aglets/>
- [7] <http://aglets.sourceforge.net/>
- [8] <http://www.bestbuy.com/site/>
- [9] Software Engineering 7 Book by Addison Wesley
- [10] <http://www.javaworld.com/javaworld>

APPENDICES

Result:

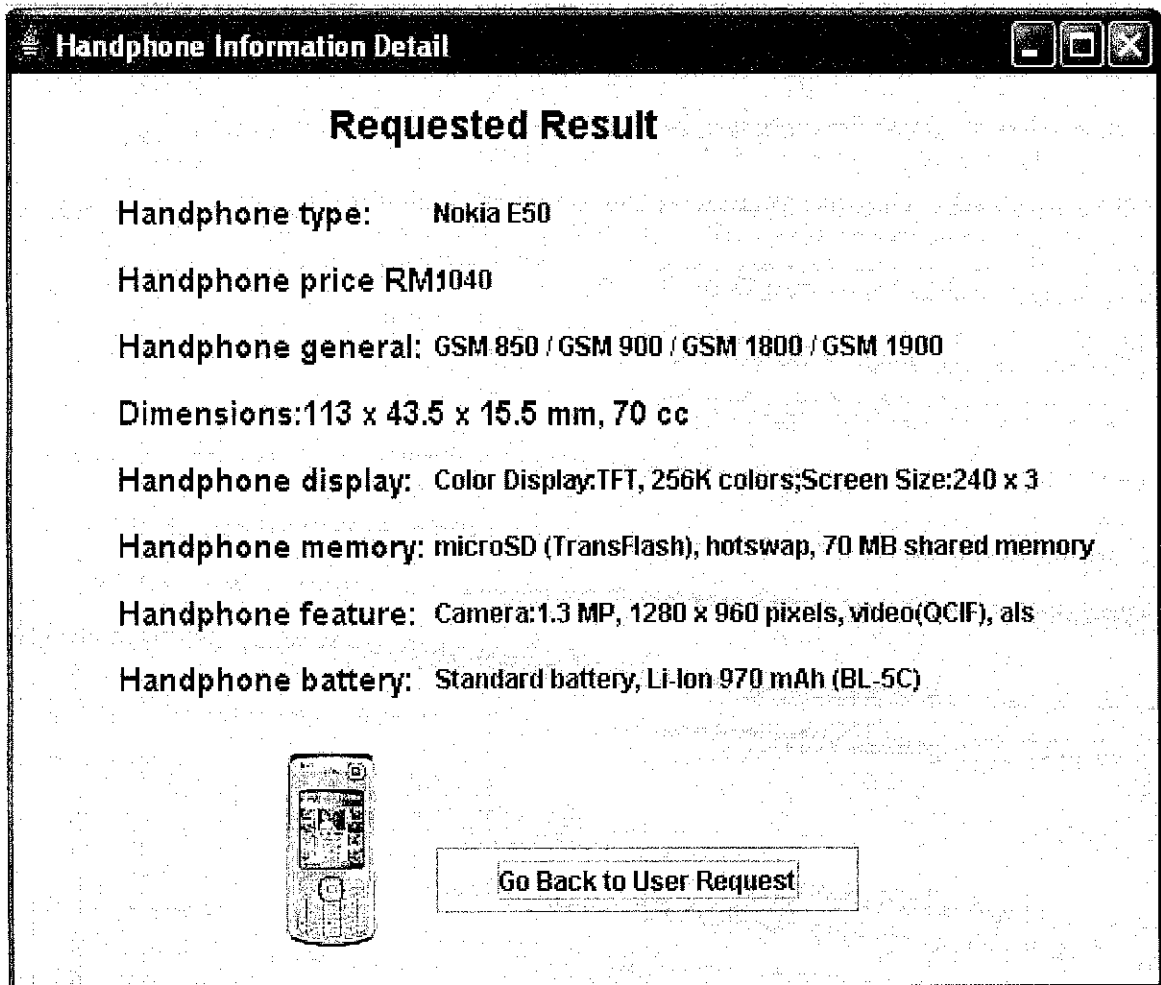


A screenshot of a web application window titled "User Request". It contains two dropdown menus. The first is labeled "Product Brand" and has "Nokia" selected. The second is labeled "Product Type" and has "Nokia E50" selected. The window has standard minimize, maximize, and close buttons in the top right corner.

User Request

Product Brand:

Product Type:



A screenshot of a web application window titled "Handphone Information Detail". It displays the requested information for a Nokia E50. The text includes the handset type, price, general specifications, dimensions, display details, memory, features, and battery information. At the bottom left is a small image of the Nokia E50 handset, and at the bottom center is a button labeled "Go Back to User Request". The window has standard minimize, maximize, and close buttons in the top right corner.

Requested Result

Handphone type: Nokia E50

Handphone price RM1040

Handphone general: GSM 850 / GSM 900 / GSM 1800 / GSM 1900


Dimensions: 113 x 43.5 x 15.5 mm, 70 cc

Handphone display: Color Display: TFT, 256K colors; Screen Size: 240 x 320

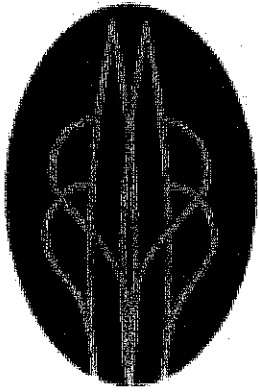
Handphone memory: microSD (TransFlash), hotswap, 70 MB shared memory

Handphone feature: Camera: 1.3 MP, 1280 x 960 pixels, video (QCIF), also

Handphone battery: Standard battery, Li-Ion 970 mAh (BL-5C)



[Go Back to User Request](#)



UNIVERSITI
TEKNOLOGI
PETRONAS

TAB4014 Final Year Project II

Information Retrieval Using Mobile Agent (Aglet)

by

Ma sochheat (4682)

Copy right@ Ma Sochheat (4682)

Information Communication Technology

Outline (1/2)

Page 2

- **Problem Statement**
- **Objectives & Scope of Study**
- **Methodology & Tools**
- **Aglet Life Cycle**
- **Agent Transfer Protocol (ATP)**
- **System Architecture**

Outline (2/2)

Page 3

- **Use Case Diagram**
- **Sequence Diagram**
- **Result**
- **Conclusion**
- **Future Enhancement**

Copy right@ Ma Sochheat (4682) Information Communication Technology

Problem Statement

Page 4

User wants to have the detail information about the type of the Handphone. User will create an agent to look for the detail information of the Handphone type that was Selected from user interface.

Copy right@ Ma Sochheat (4682) Information Communication Technology

Objectives & Scope of Study

Page 5

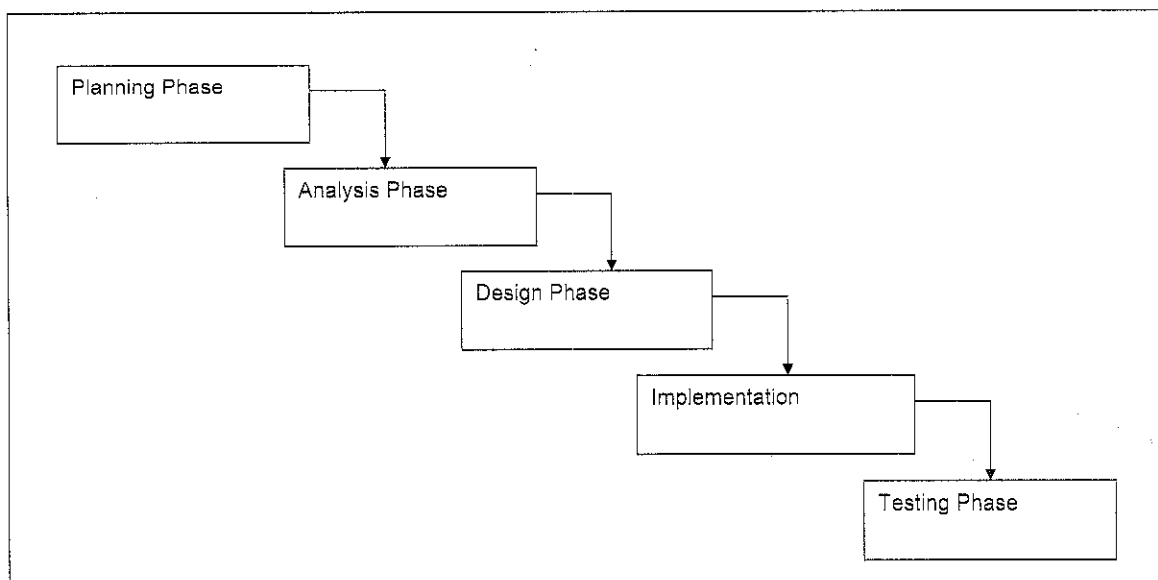
- √ User can brows to see the provided version of hand phone
- √ Help the user to experience the capability of Mobile Agent when the information being requested
- √ Help user to get the result faster
- √ Provide user more option: help user to search for hand phone detail information by selecting the criteria that they want:
 - ❖ *Brand*
 - Nokia
 - Samsung
 - Sony Ericsson
 - ❖ *Type of the hand phone*

Copy right@ Ma Sochheat (4682) Information Communication Technology

Methodology & Tools (1/3)

Page 6

√ Software process model



Copy right@ Ma Sochheat (4682) Information Communication Technology

Methodology & Tools (2/3)

Software process model detail

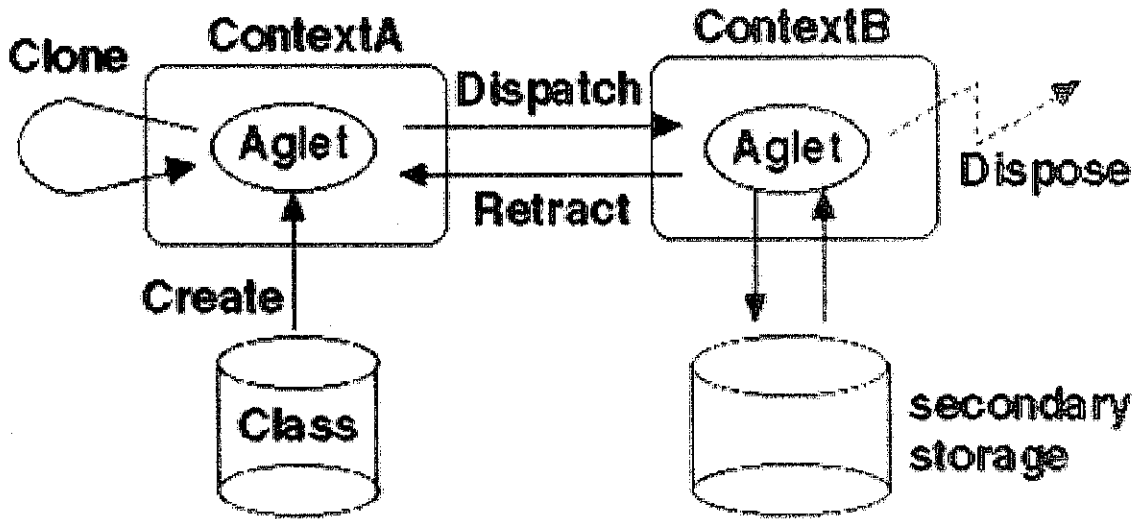
- ❖ Plan the hardware or software required for the project.
- ❖ Analyze and get Product Requirement
- ❖ Design the process of implementing the system
- ❖ Implement the project by using mobile agent platform – aglets that involves programming activity
- ❖ Test the product to ensure the product successfully completed and fulfill the objectives.

Methodology & Tools (3/3)

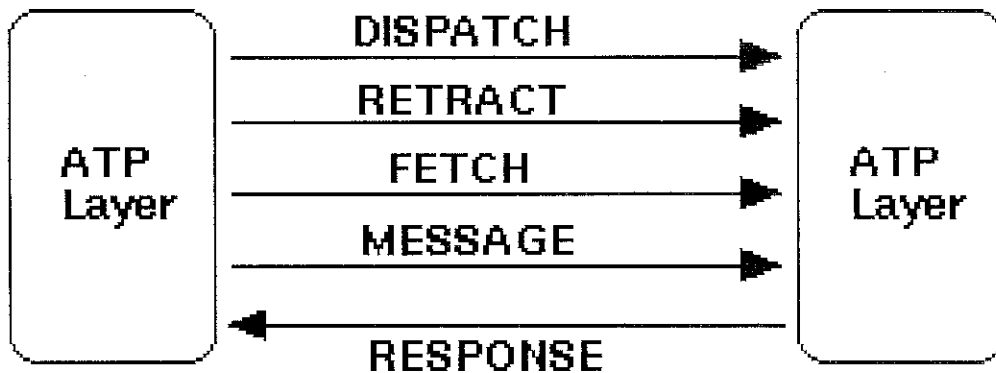
Tools

- I. Java Development Kit
(JDK1.5.0_08)
- II. Aglets Software Development Kit
(ASDK - aglets-2.0.2.jar)
- III. MySQL Database System
- IV. Microsoft Visual J++ 6.0 (Java editor)

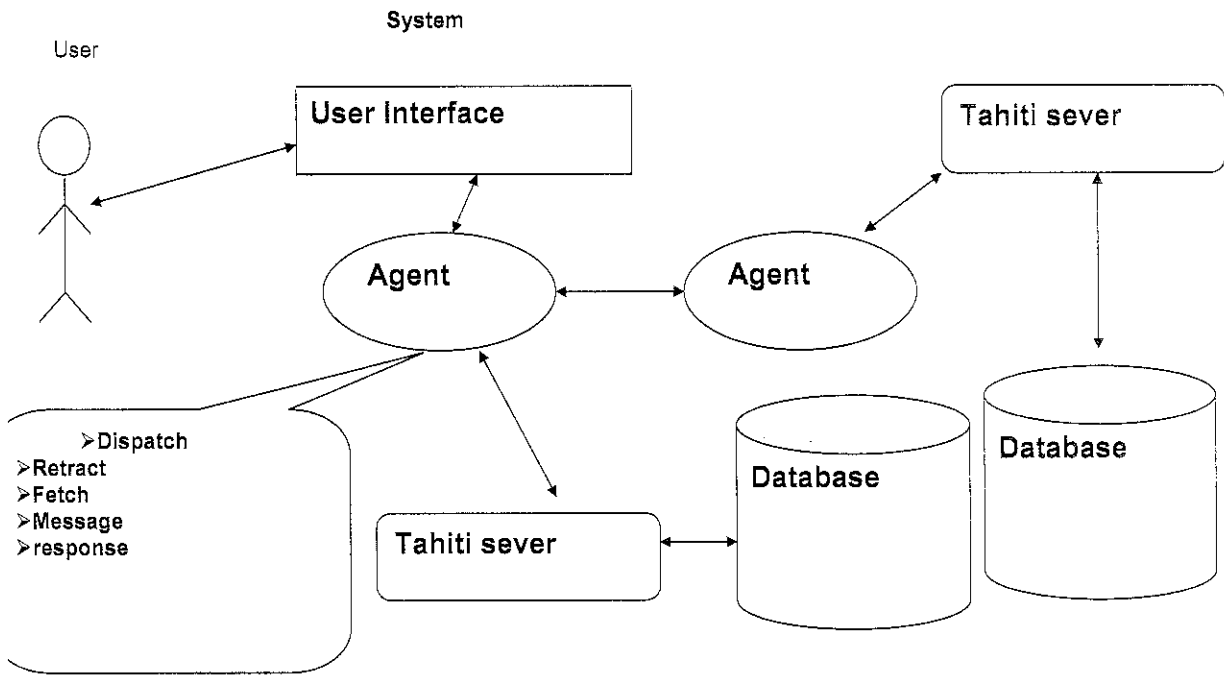
Aglet Life Cycle



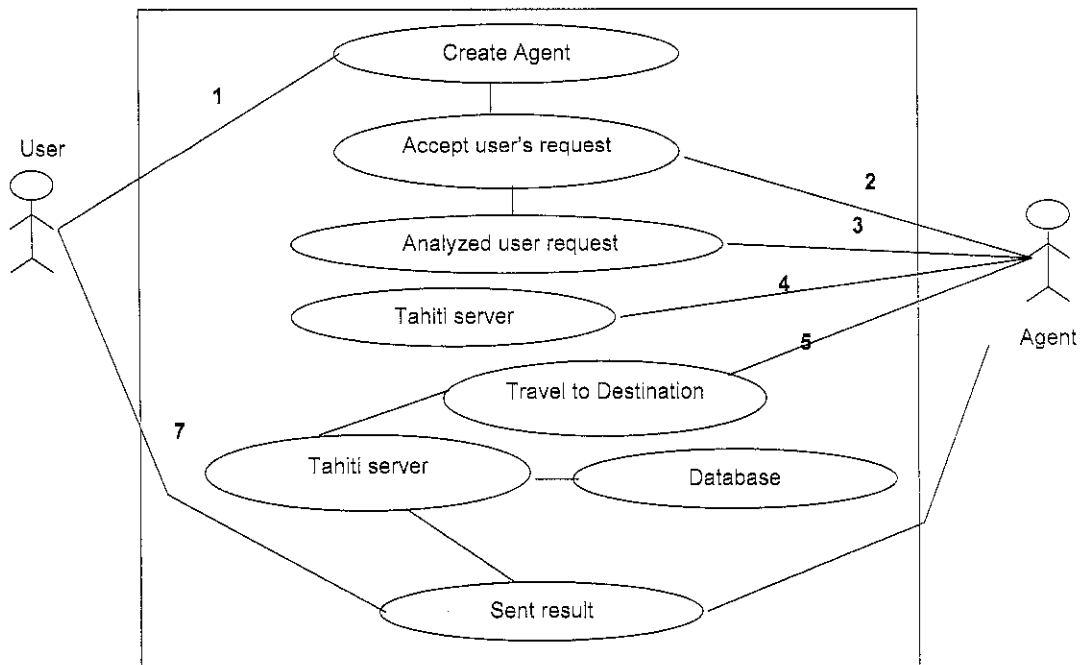
Agent Transfer Protocol



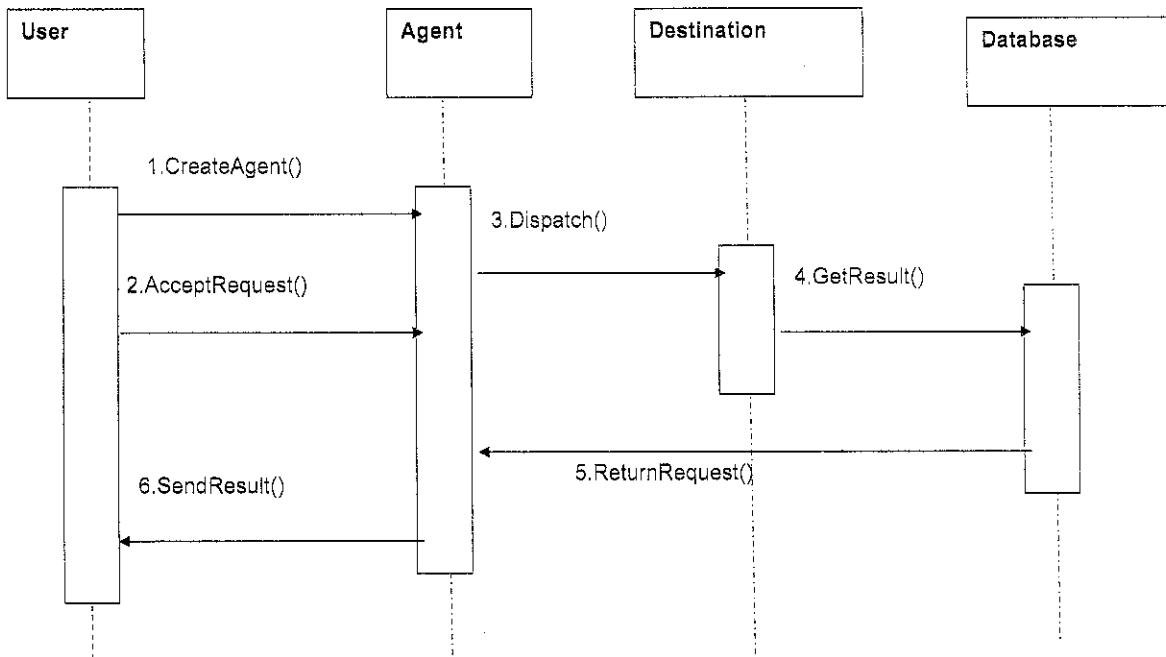
System Architecture



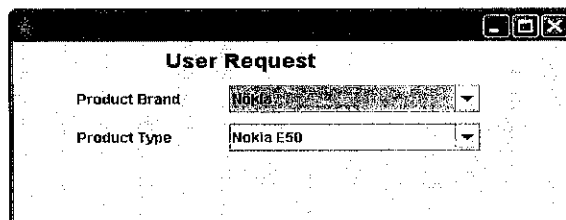
Use case Diagram



Sequence Diagram



Result (1/2)




There are 3 Nokia Type(s) in Nokia table.
Nokia E50
Nokia N75
Nokia N95
Handphone Type: Nokia E50
Handphone Price: 1040
Handphone General: GSM 850 / GSM 900 / GSM 1800 / GSM 1900
Handphone Size: Dimensions:113 x 43.5 x 15.5 mm, 70 cc
Handphone Display: Color Display:TFT, 256K colors;Screen Size:240 x 3
Handphone Memory: microSD (TransFlash), hotswap, 70 MB shared memory
Handphone Features: Camera:1.3 MP, 1280 x 960 pixels, video(QCIF), als
Handphone Battery: Standard battery, Li-Ion 970 mAh (BL-5C)

Result (2/2)

Handphone Information Detail

Requested Result

Handphone type: Nokia E50
Handphone price RM1040
Handphone general: GSM 850 / GSM 900 / GSM 1800 / GSM 1900
Dimensions: 113 x 43.5 x 15.5 mm, 70 cc
Handphone display: Color Display:TFT, 256K colors;Screen Size:240 x 3
Handphone memory: microSD (TransFlash), hotswap, 70 MB shared memory
Handphone feature: Camera:1.3 MP, 1280 x 960 pixels, video(QCIF), als
Handphone battery: Standard battery, LI-Ion 970 mAh (BL-5C)



[Go Back to User Request](#)

Conclusion

Aglet has been serve as the useful agent to retrieve the information needed by the user. Up on the arrival at the other server, user can request the information and the agent will go back home after the retract method is performed.

Future Enhancement

Page 17

- Develop the same application by using different mobile agent toolkits.
- Research on Distributed Database using Mobile agent
- Develop the same system and host it on the web
- Provide the Download Function for the user to download the ring tone, mobile phone application, multimedia image etc

Copy right@ Ma Sochheat (4682) Information Communication Technology

Page 18

Thank You!!

Demo

Q&A

Copy right@ Ma Sochheat (4682) Information Communication Technology