E-AUCTION SYSTEM USING MOBILE AGENT TECHNOLOGY

By

NURAFIDAH BINTI ABDUL RAZAK

Dissertation submitted in partial fulfillment of
the requirements for the
Bachelor of Technology (Hons)
(Information Technology)

JUNE 20, 2005

Universiti Teknologi PERONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

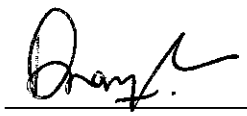
E-Auction System Using Mobile Agent Technology

by

Nurafidah Binti Abdul Razak

A project dissertation submitted to the
Information Technology Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirements for the
BACHELOR OF TECHNOLOGY (Hons)
(INFORMATION TECHNOLOGY)

Approved by,

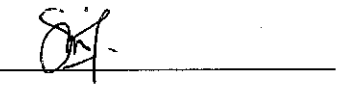


(Mr. Anang Hudaya bin Muhamad Amin)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
JUNE 2005

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.



NURAFIDAH BTE ABDUL RAZAK

ABSTRACT

Mobile agent technology has known an important rise these last years. This project will present 3 objectives in constructing an E-Auction system which are; to study on how mobile agent system is said to be useful to reduce communication cost in e-auction system as well as to accept the use of limited local resources, and to have an asynchronous computing. 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 buy. Once they have found the destination or server, the piece of software or the latest hit single which user looking for, the agent can send the result of the best requested item on behalf of the user. This project will be focusing on E-Auction business on PDA's and will be using mobile agent platform which is aglets, Java runtime, and J2sdk1.4.1_03. This project will undergo 5 stages which are; plan the hardware or software required for the project, then analyze and get Product Requirement, next is design the process of implementing PDA's E-Auction 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 satisfy 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, this project is aimed to create a simple E-Auction 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

Bismillah Ar-Rahmani Ar-Raheem

In the Name of Allah, The Most Compassionate, the Most Merciful

In order to complete this report, I have been doing researches over internet and reading materials over mobile agent technology in this The E-Auction System Using Mobile Agent Technology project, I would like to thank:

1. Mr. Anang Hudaya bin Muhamad Amin, my supervisor (for giving me the guidelines and ways in producing a good output and full support in terms of knowledge input along this project)
2. The Backbone of FYP Committee – Mr. Mohd Nor Ibrahim and Ms. Vivian, and all IT/IS lecturers, (for giving full commitment in term of providing information about the final year project)
3. My parents, Mr. Abdul Razak Bin Dolan, and Mrs Sabariah binti Md. Din and family who supports me financially and mentally
4. My twin, Nurhayati binti Abdul Razak, who gives me full support and shared her knowledge mobile agent technology
5. Mr. Rasky, the technician who responsible to provide cooperation for Final Year student to do the Final year Project.
6. Friends, for their support and information over my Final Year project
7. Those who involved directly and indirectly towards the project.

TABLE OF CONTENTS

ABSTRACT. i

ACKNOWLEDGEMENT ii

TABLE OF CONTENT iii

LIST OF FIGURES v

ABBREVIATIONS AND NOMENCLATURES . . . vi

CHAPTER 1:INTRODUCTION **1**

1.1 Background of Study 1

1.1.1 Mobile Agents . . . 2

1.1.2 Aglets 2

1.1.3 E-Auction . . . 3

1.2 Problem Statement 3

1.2.1 Problems Identification . . 4

1.2.2 Significant of the Project . . 4

1.3 Objectives and Scope of Study . . 5

1.3.1 The Relevancy of the Project . . 5

1.3.2 Feasibility of the Project within the Scope and Time Frame . . 6

CHAPTER 2:LITERATURE REVIEW **7**

2.1 Supporting Information . . . 7

2.1.1 Mobile Agents Technology . . 7

2.1.2 Agents. 8

2.1.3 Aglets 9

2.1.4 The Interesting of Mobile Agent Technology 10

2.1.5 The Advantages and Benefits of Mobile Agents. 11

2.1.6 E-Auctions 12

2.1.7 Assumptions 13

2.1.8 Current Project . . . 13

CHAPTER 3: METHODOLOGY 15

3.1 Procedure Identification . . . 15

3.1.1 Planning Phase . . . 16

3.1.2 Analysis Phase . . . 16

3.1.2.1 Tahiti Server Functionality 19

3.1.3 Design Phase 20

3.1.4 Implementation Phase. . . . 20

3.1.5 Testing. 21

3.2 Tool 21

CHAPTER 4: RESULTS AND DISCUSSION. . . . 22

4.1 Findings 22

4.1.1 How the E-Auction System Work Generally 22

4.1.2 Agent Life Cycle . . . 23

4.1.3 Result Comparison . . . 25

4.2 Discussion 26

4.2.1 Shopper Agent . . . 27

4.2.2 Auctioneer Agent . . . 27

CHAPTER 5: CONCLUSION AND RECOMMENDATION . 29

5.1 Conclusion 29

5.2 Recommendation 30

REFERENCES 32

APPENDICES 33

LIST OF FIGURES

Figure 2.1: Aglet Object Model

Figure 3.1: The selected Methodology Model

Figure 3.2: The Command Prompt

Figure 3.3: The Tahiti Login Window

Figure 3.4: The Tahiti Main Window

Figure 3.5: Interface to Create an Agent

Figure 3.6: The Result of Creating an Agent

Figure 3.7: Dispose an Aglet

Figure 4.1: Agent Life Cycle

LIST OF TABLES

Table 2.1: Auction Protocols

Table 4.1: Agent Life Cycle Description

ABBREVIATION AND NOMENCLATURES

1]	DAI	:	Distributed Artificial Intelligent
2]	JRE	:	Java Runtime
3]	MA	:	Mobile Agent
4]	MAS	:	Multi-agent System
5]	PAI	:	Parallel Ai
6]	PDA	:	Personal Digital Assistant
7]	DPS	:	Distributed Problem Solving
8]	RPC	:	Remote Procedure Call
9]	WWW	:	World Wide Web

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

This introduction for dissertation report explains the fundamental information of the project, which consist background of study – explain more thoroughly about the evolving of mobile agent technology as well as the background and history of mobile agent technology, problem statement(s), objective(s) and scope of study.

1.1 Background of Study

Software agents have been a popular topic in the area of information engineering for several years. Lots of expectations have been laid on them, but large-scale usage of agents is still waiting [Wallin2004]. Look back at the starting point of evolving agent technology, which discovered that software agents evolved from Multi-agent Systems (MAS). MAS, along with Distributed Problem Solving (DPS) and Parallel AI (PAI), branched from Distributed Artificial Intelligent (DAI). There is no consensus in the industry to the definition of what an agent is. As defined in one online research paper, mobile agent as a program with special characteristics that gives it certain capabilities [KhaDia]. Mobile agents have attracted significant attention recently. In addition to mobile code (such as applets), agents consist of data and non-transient system state that can travel between the nodes in a distributed system (Intranet or Internet) [MiLaChau]. Mobile agents also represent someone; they can perform autonomous actions on behalf of a user or another agent and migrate from machine to machine in a heterogeneous network. A number of academic systems (such as Agent Tcl, Mole, Ara and Tacoma)

and industrial systems (such as Telescript, Aglets, Concordia and Voyager) exist. The products using mobile agents have started to appear, such as Guideware. The deeper explanations about mobile agents nowadays will be presented in the next subtopic.

1.1.1 Mobile Agents

According to lecture notes from Monash University [Kendall], mobile agent is a software program that moves from machine to machine under its own control. It can suspend execution at any pointing time, transport itself to a new machine and resume execution. Once the agent is created, a mobile agent autonomously decides which locations to visit and what instructions to perform. The continuous interaction with the agent's originating source is not required as it implicitly specified through the agent code and specified through a run-time modifiable itinerary. As to deeper understand the mobile agent technology, get the concept of mobile agent first. A mobile agent is an object which can autonomously move via the network from one host to another along with its code and execution state and perform some task on behalf of a user.

1.1.2 Aglet

In the Oxford dictionary definition, an aglet is "a metal tag attached to each end of a shoelace", "metal tubes at the end of your shoe lace!", and also "light weight agent". Actually, aglet refers to agent plus applet. According to the lecture notes from Monash University, aglet is defined as java based autonomous software agent. It also added that mobile agent is supports the ideas of autonomous execution and dynamic routing on its itinerary. Besides that, aglet is a java object that can be transported along with state information and it can receive request from external sources, but each individual aglet decides whether or not to comply with external requests. It can be execute in any aglet environment and operating within any Java environment. In addition, it is a Java programs that can halt execution, travel across the network (with both code and state in tact) and continue execution at another host.

An agent is hosted by an Aglet Server i.e. Tahiti Server. Let's get clear idea about an aglet by compare the aglet with applet. Like an applet, the class file for an aglet can migrate across a network. Unlike applets, when and aglet migrates it also carries with it its state. In java technology, an applet is code that can move across a network from a server to a client. However, an aglet is a running java program (code and state) that can move from one host to another on a network. In addition, aglet carries its state wherever it goes, it can travel sequentially to many destinations on the network, including eventually returning back to its original host.

1.1.3 E-Auction

Source from internet [WurWelWal98] indicates that e-auction have rapidly achieved enormous popularity on the Internet. EBay, one of several commercial sites that run user-created auctions, claims to be transacting nearly $2 million a week. The industry has rapidly spawned sub industries, such as newsletters, auction software providers, and specialized search engines. In addition to their use in online retail, automated auctions are also found at the core of systems for market-based resource allocation.

Now, it is the year of 2000 century and of course the developments of e-auction system through internet become more sophisticated. In order to complete the final year project course, the title of e-auction system using mobile agent technology has been selected and using the information given at the above paragraph as the guidelines to build an e-auction system. Moreover, the main objective of this project is only to create a simple e-auction system that can be implemented using mobile agent technology which known as the new evolving technology in the industry.

1.2 Problem Statement

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 buy. Once they have found

the destination or server, the piece of software or the latest hit single which user looking for, the agent can send the result of the best requested item on behalf of the user.

1.2.1 Problem Identification

- A transaction or query between a client and the server may require many round trips over the wire to complete. Each trip creates network traffic and consumes bandwidth. In a system with many clients and/or many transactions, the total bandwidth requirements may exceed available bandwidth, resulting in poor performance for the application as a whole. This is the result of **nagging client/server network bandwidth problem.** The problem arise above is always involved with the current auction system through internet.
- Moreover, in the design of traditional client/server architecture, the architect spells out the roles of the client and server pieces very precisely -- up front, at design time. The architect makes decisions about where a particular piece of functionality will reside based on network bandwidth constraints (remember the previous problem), network traffic, transaction volume, number of clients and servers, and many other factors. If these estimates are wrong, or the architect makes bad decisions, the performance of the application will suffer. Unfortunately, once the system has been built and the performance measured, it's often difficult or impossible to change the design and fix the problems.
- Furthermore, there have problems created by intermittent or unreliable network connections. (Todd Sundsted, 1998)

1.2.2 Significant of the Project

Reduce communication costs – there may be a lot of raw information that need to be examined to determine their relevance in PDA's auction system. Transferring this raw information can be very time-consuming and clog of the networks. Imagine having to transfer many images of PDAs just to pick one out. It is much more natural to get your agents to "go" to that location, do a local searching/pruning of the suitable PDA and

only transfer the chosen compressed image or a piece of result back across the network2. It obviates the need for costly network connections between remote computers as required in remote procedure calls (RPC).

Limited local resources – the processing power and storage on the local machine may be very limited (only perhaps for processing and storing the results of a PDA), thereby necessitating the use of mobile agents.

Asynchronous computing – you can set off your mobile agents and do something else and the results of your favorites PDA will be back in your mailbox, say, at some later time. They may operate when you are not even connected.

1.3 Objectives and Scope of Study

This proposed project is to develop a simple e-auction system using mobile agent technology. This project will present 3 objectives in constructing an e-auction system which are; to study on how mobile agent system is said to be useful to reduce communication cost in e-auction system as well as to accept the use of limited local resources, and to have an asynchronous computing. The scope of this project is focusing on e-auction system for PDA's business tools. E-auction markets of PDA's can centralize supplies and demands on a single virtual market place. Furthermore, it will be less need for mobile agents as suppliers and buyers can see the potential trading partners.

1.3.1 The Relevancy of the Project

This project is relevant to the organization that handle business information online and serving their customer via 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 agents can be introduced in intelligent PDA's auction networks to reduce the complexity of the business process. The works in intelligent networks area allowed separating services from the network itself, which brings 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 and Time Frame

This project is feasible by using mobile agent platform – aglets, Java runtime (JRE), and Java 2 Software Development Kit - J2SDK1.4.2_03. The project will take five (5) months starting from Jan 2005 until May 2005. 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 PDA's E-auction 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

This section contains the acknowledged findings on this study, consisting of relevant concept, facts and data that are collected from books, journals and online researches. To have thorough understanding on e-auction system using mobile agent technology, the supporting information given below illustrated deeply the ideas and theory that have been adopted previously by some organizations around the world.

2.1 Supporting Information

2.1.1 Mobile Agents Technology

Mobile agent technology is the next logical step following the explosive growth of the internet. One of the major growth potentials of mobile software agents is in electronic auction. Such environments are to exist in the form of electronic markets where time and space limitations are eliminated and advanced mechanisms are deployed for the realization of safe and reliable trading [KhaDia]. **Mobile** agents are defined in formal terms by computer scientists as objects that have *behavior*, *state*, and *location* [Sommers97]. The set of events varies a bit from model to model, but the following is a list of the most common ones:

- Creation - Analogous to the constructor of an object. A handler for this event should initialize state and prepare the agent for further instructions.

- Disposal - Analogous to the destructor of an object. A handler for this event should free whatever resources the agent is using and prepare the agent for burial.
- Dispatch - Signals the agent to prepare for departure to a new location. This event can be generated explicitly by the agent itself upon requesting to migrate, or it can be triggered by another agent that has asked this agent to move.
- Arrival - Signals the agent that it has successfully arrived at its new location and that it should commence performing its duties.
- Communication - Notifies the agent to handle messages incoming from other agents and is the primary means of interagent correspondence.

2.1.2 Agents

According to some points resulted from studying the lecture notes from Monash University, agents are an incremental development in Network Computing and Software Engineering as well as an incremental development on object. It is a bit of intelligent systems, can say that it is interesting, and is a real application. When derived to the definition of an agent, it stated that agents are autonomous programs that can handle tasks on behalf of their owner. It also suited to operating within wireless network because they can save bandwidth by traveling to the source of the request, as well as can operate even when their owner is disconnected from the network. There is no consensus on a single definition for an agent. It also a new Buzzword and everybody wants to call their software "Agent". Thus, the concept of agents presents that agents are software system which acts "intelligently" on user behalf. Agent is convenient metaphor that situated in an environment and exhibits behavior which can be viewed as: pro-active; autonomous; communication; persistent; mobile; benevolent; adaptive/learning; collaborative; reactive; deliberative; and etc. Agent also has stronger notions of agency: mentalistic notions such as knowledge, beliefs, desire, intention, goals, and logic for reasoning with them. Agent is cutting out the hype as it is a kind of software and an approach to software design. Below are the important types of agents:-

- Collaborative
- Interface
- Mobile
- Information / internet
- Others – Reactive, Hybrid, Smart

2.1.3 Aglets

Aglets are **autonomous mobile software agents**, which are programs that can be dispatched from one computer and **transported to a remote computer** for execution. Arriving at the remote computer, they present their credentials and **obtain access to local services and data**. The remote computer may also serve as a broker by bringing together agents with similar interests and compatible goals, thus providing a meeting place at which agents can interact.

An aglet has the following characteristics:

- **Object-passing capability.** It is a complete program object with its own methods, data states, and travel itinerary that can send other aglets or pass itself along in a network as an entity.
- **Autonomous.** An aglet has the ability to decide on its own what actions to take and where and when to go elsewhere.
- **Interaction with other program objects.** It can interact locally with other aglets or stationary objects. When necessary, it can dispatch itself or other aglets to remote locations to interact with other objects there.
- **Disconnected operation.** If a computer is currently disconnected from the network, the aglet can schedule itself to move when the computer is reconnected.
- **Parallel execution.** Multiple aglets can be dispatched to run concurrently in different computers.

Below is the aglet object model as stated in the lecture notes from Monash University:

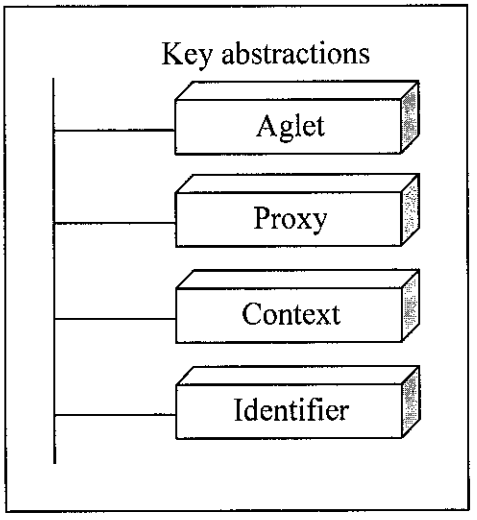


Figure 2.1 Aglet Object Model

2.1.4 The Interesting of Mobile Agent Technology

From the findings that have been gathered through multiple journals and online researched, mobile agent's definition can be summarized as programs that are capable of being transferred to remote hosts in order to carry out different tasks on behalf of their users as well as can halt its execution, move to another host on the network while maintaining their state, and resume execution on the destination host [LaAri97]. From the findings that have been gathered through multiple journals, *Mobile agent* use specialized servers to interpret the agent's behavior and communicate with other servers.

A *Mobile Agent* has inherent navigational autonomy and can ask to be sent to some other nodes. *Mobile Agents* should be able to execute on every machine in a network and the agent code should not have to be installed on every machine the agent could visit. Therefore *Mobile Agents* use mobile code systems like Java and the Java virtual machine where classes can be loaded at runtime over the network [Cetus-Link2002]. *Mobile agents* are agents that can physically travel across a network, and perform tasks on machines that provide agent hosting capability. This allows processes to migrate from computer to computer, for processes to split into multiple instances that execute on different machines, and to return to their point of origin. *Mobile agent* require a scenario

where a process invokes procedures of a remote host, process migration allows executable code to travel and interact with databases, file systems, information services and other agents.

The technology behind mobile agents is examined, and an analysis of its uses and implications is offered [Reilly]. The appeal of mobile agents is quite alluring - *mobile agents* roaming the Internet could search for information, find user great deals on PDA's, and interact with other agents that also roam networks (and meet in a gathering place) or remain bound to a particular machine.

2.1.5 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 [Versteeg97]. 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:

- 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 operations

- Mobile agents combine the strengths of techniques such as RPC, java applets etc. into a single, convenient framework.

2.1.6 E-Auction

A negotiation mechanism is essentially a protocol within which agents interact to determine a contact. Auctions constitute a general class of such protocols, as characterized in the standard definition expressed by McAfee and McMillan (1987):

"An auction is market institutions with an explicit set of rules determine resource allocation and prices on the prices on the basis of bids from the market participants. The above points are findings from a result of research through internet" [WurWelWal98]. It presents the idea of the auction environment in the industry.

Besides that, there are also some points related in the online research about the auction mechanism as well as the auction protocols [Kendall]. Valuation is one of the mechanisms in auction which derived to the definition on how an agent's value of the item is formed in three auctions which are:

- Private value auction
  - Value depends on agents own preferences (e.g. buying a cake)
- Common value auctions
  - Value depends completely on other agents values (e.g. buying treasury bonds)
- Correlated value auctions
  - Value depends on both self and other

Now, talked about the auction protocols, it have 4 protocols which are an English, sealed bid, Dutch and Vickery. The details about each protocol have discussed in the table below:

English	First-price open-cry ▣ Each bidder is free to raise his bid ▣ When no bidder is willing to raise, the auction ends with the highest bidder winning
Sealed Bid	First-price sealed bid ▣ Each bidder submits one bid without knowing the other bids ▣ The highest bidder wins and pays the amount of his own bid
Dutch	Descending ▣ Seller continuously lowers the price until one of the bidders takes price at the current price
Vickery	Second-price sealed bid ▣ Each bidder submits one bid without knowing the other bids ▣ The highest bidder wins, but pays the price of the second highest bid

Table 2.1: Auction Protocols

2.1.7 Assumptions

For this e-auction system using mobile agent technology's project, it will be assumed to use the sealed bid auction protocol which is the first-price sealed bid. As explained in the next chapter, this e-auction system is assumed that the Agent created will be doing the bidding by it owns automatically and that working areas are not the major consequences of this project.

2.1.8 The current project

Electronic auction house first to offer mobile agents is the current example of e-auctions using mobile agent technology. According to the research journal [Fitz], the booming world of electronic commerce now provides bidders using Internet auction houses such as eBay and OnSale with virtual agents to do their bidding. They explained that those agents are programs that literally do a buyer's bidding as he or she goes about another

task, from cleaning the house to closing a big corporate deal. The agent reports back to the buyer on his or her PC every five minutes or so and is instructed not to bid above certain price. They are the first Internet auction house with mobile agents. They let the user create his or her own agent. They create the bidding based on the game theoretic analysis, they compiled the optimal bidding strategies into different agents. This put a novice user and a game theorist/expert bidder on equal footing for electronic commerce.

CHAPTER 3

METHODOLOGY

3. METHODOLOGY

3.1 Procedure Identification

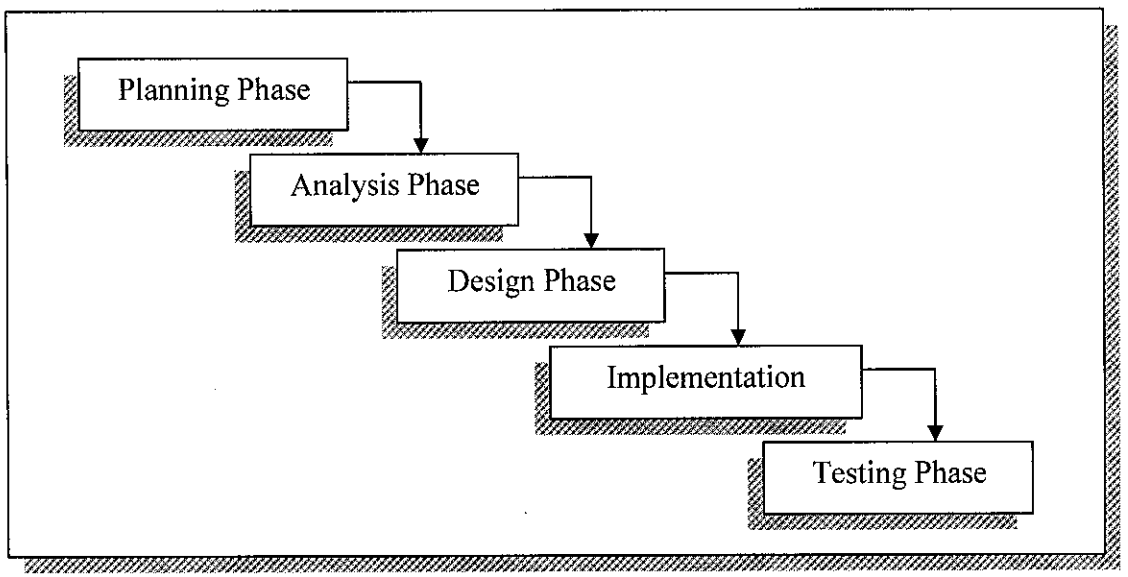


Figure 3.1: The selected methodology model

- Plan the hardware or software required for the project.
- Analyze and get Product Requirement.
- Design the process of implementing PDA's E-auction 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 "e-auction system using mobile agent technology" 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 to 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 continued with the development of class diagram and 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 Aglets 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.

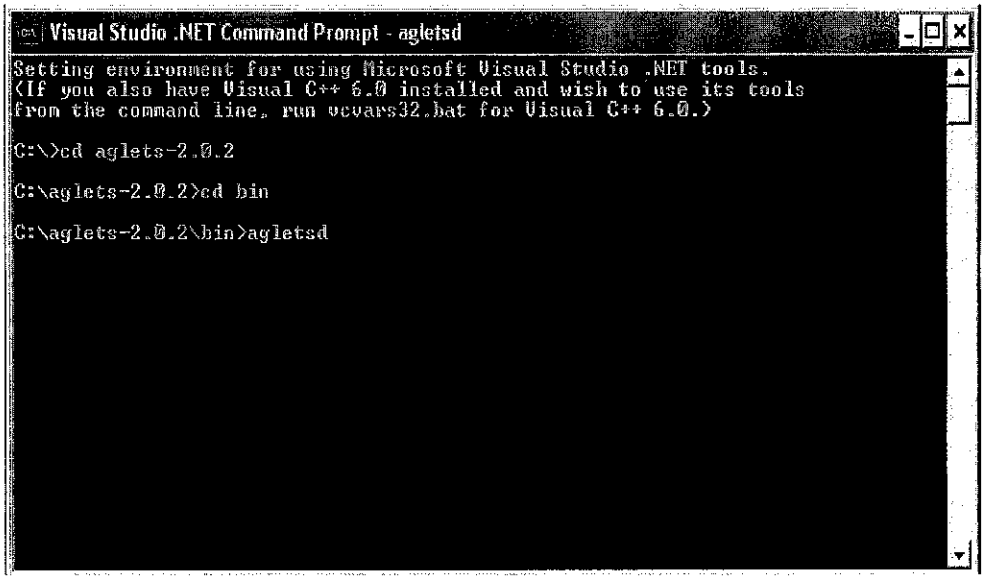


Figure 3.2: The command prompt window to run the agletsd (Tahiti Server)

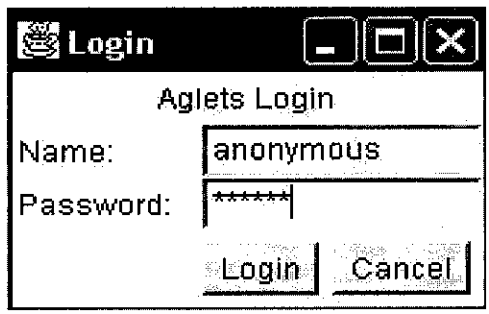


Figure 3.3: The Tahiti Login Window

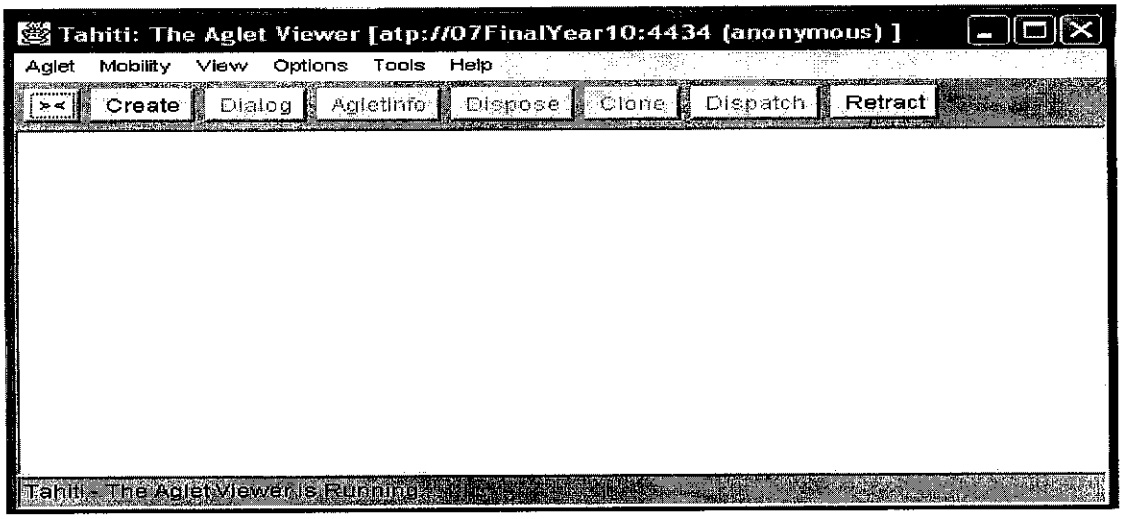


Figure 3.4: The Tahiti Main Window

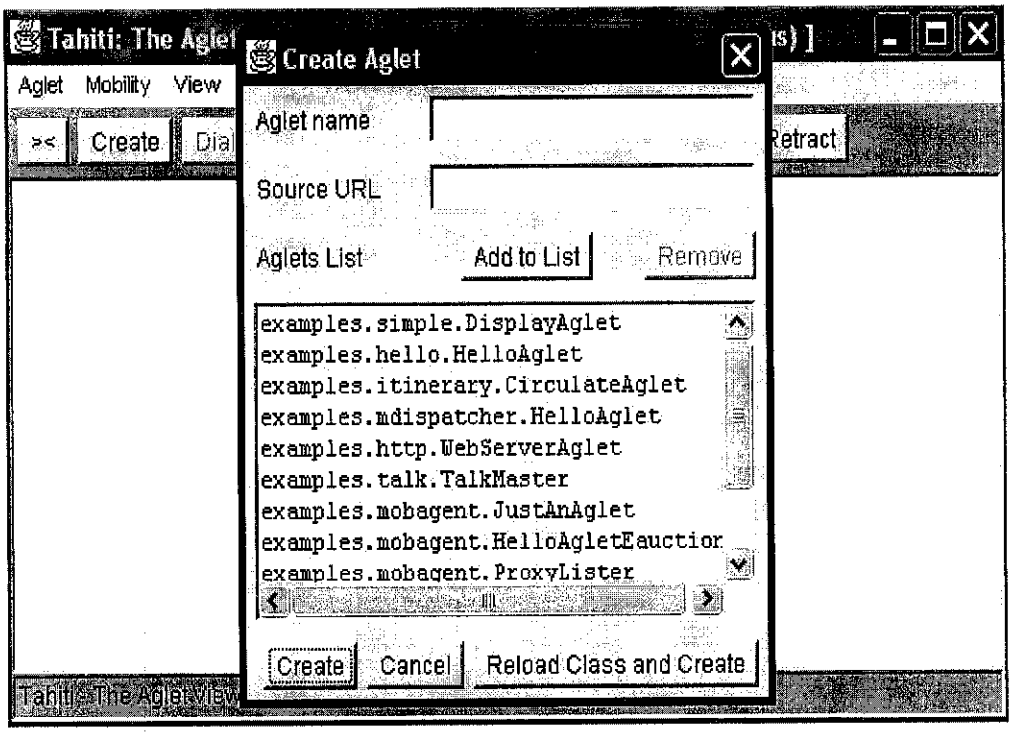


Figure 3.5: Interface to create an agent

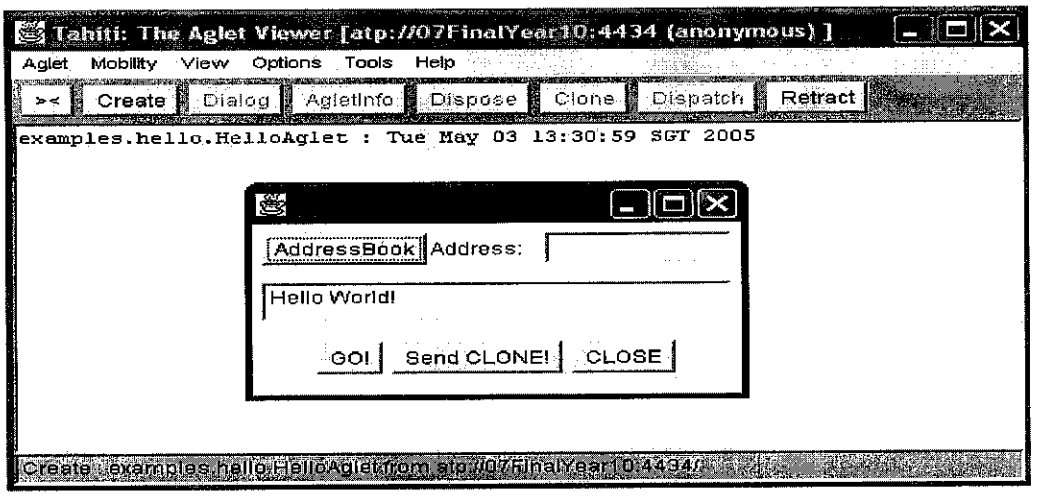


Figure 3.6: The result of creating an agent

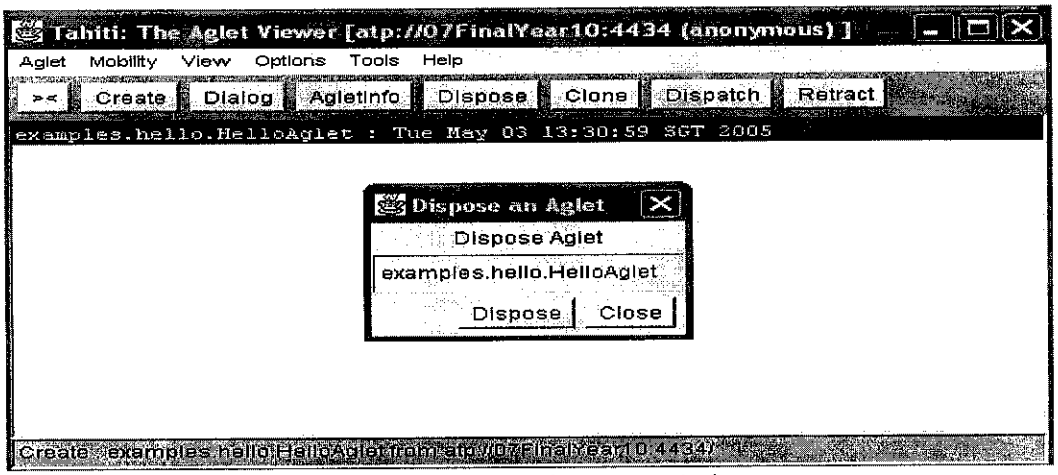


Figure 3.7: Dispose an Aglet

3.1.2.1 Tahiti Server Functionality

First of all, as shown in figure 3.2, the aglets server should be run by executed the command of agletsd. Then, it will appear a Tahiti login window (figure 3.3) and will ask the user the authenticate itself. In this case, the username will be anonymous and the password will be aglets. Figure 3.4, 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 (e.g: "examples.mobagent.HelloAglet") and "Reload Class and Create". Every time user modifies and compiles the program, user need to "Reload Class and Create". Refer to figure 3.5 for details. The figure 3.6 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 buttons in the Tahiti main window. However, figure 3.7 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.

3.1.3 Design Phase

The third phase is design phase, which is design the process of implementing E-auction system. At the previous stage, this project has done the analysis on the idea behind e-auction system. Therefore, at this stage the design for those output from the analysis are to be drawn in an appropriate diagram or graphical representations. 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 e-auction system.

3.1.4 Implementation Phase

The implementation phase is the next phase after design. This phase will come out with the well structured 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 build as simplest as it can as 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 chapter 4.

3.1.5 Testing Phase

For the last phase in this project, all 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 expert who is Mr. Anang himself; react as the supervisor of this topic for final year project.

3.2 Tools Required

3.2.1 Software

- Aglets 2 Platform – Aglets 2.0.2
- Java 2 Run-Time Environment – JRE
- Java 2 Source Development Kit (J2SDK)
- Adobe Photoshop 7.0

3.2.2 Hardware

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

CHAPTER 4

RESULTS AND DISCUSSION

4. RESULTS AND DISCUSSION

This chapter presents the findings or outcome of the project work. By following the methodology as stated in Chapter 3, the outcome and results will be presented in this chapter.

4.1 Findings

Implementation of e-auction system using mobile agent technology was a challenging and time-consuming task. During the time given, this project has successfully produce a smoothly running Aglet either using only one Tahiti server or multiple Tahiti server. the importance points were all the operations in the Tahiti server can be implemented, although the e-auction system did not cope-up as real as it can, but it still did not out of the concept line of auction.

4.1.1 How the E-auction System Work Generally

In order to implement the mobile agent technology (Aglet) in E-Auction system, the user basically needs to interact and create his/her own agent to get the best matching result of the requested good (PDA). The user does not have to login any pages as required at the WWW (World Wide Web). Thus, does not have to wait until the confirmation is successful and valid then return to them back. What the user only needs to do is to participate in the request activities, then assigned an agent and which is more correctly

to say, to create an agent that will be performed all the auction transactions. The auction also will do the bidding and negotiation activities by itself and travel by its own decision and come back to its owner with the best result. However, in this project the bidding and negotiation activities are assumed have been done correctly and successfully without focusing on it formulas and its coding parts.

4.1.2 Agent Life Cycle

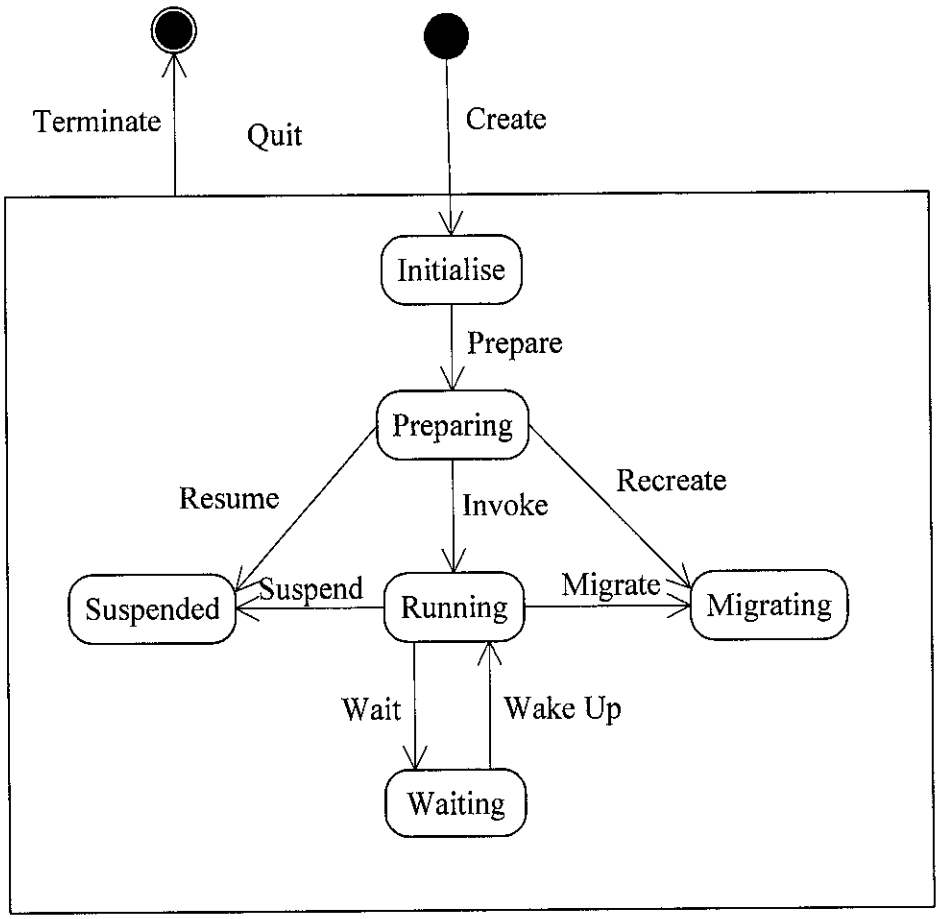


Figure 4.1: Agent Life Cycle

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 deserialised and recreated after migration
Quit	The agent itself chooses to ends its life cycle
Terminate	The management services forces the agent to terminate

Table 4.1: Agent Life Cycle Description

4.1.3 Result Comparison

As we can see in the appendices page, the sequence diagram of e-auction system using mobile agent technology show that user do not have to do many transaction in using the e-auction system. All the transaction is done by the assigned agent. However, for e-auction system that involve with homepage or world wide web show that the user need to do many activities and need to wait for so many approval of transaction by the server. Here shows that e-auction system via World Wide Web is wasting the user time, therefore, by using mobile agent technology, perhaps this problem can be solve. Moreover, by using mobile agent technology, only the best result of the user request will be send to the user by the assigned agent. It is different for e-auction system via World Wide Web where the user request the interested information and the server provide so many related information and it will tend to occur a communication cost in that particular activities. At this point of time, the objectives of to study on to study on how mobile agent system is said to be useful to reduce communication cost in e-auction system can be figured out. In addition, the second objective which is to study on how mobile agent system can accept the use of limited local resources would also be figured out by looking at the first point before. Mobile agent technology is said have created an agent within the transaction and the agent will only send one best result to the user. Then, user does not need to have so many spaces to save the result in its hard disk and this study have prove it. By the way, the e-auction system via World Wide Web failed to achieve this objective. Besides that, using mobile agent technology, whenever user disconnected from internet, the agent will still send back the requested information to the user. This is because the agent still does their job whenever the user is disconnected from the network. However, if user is disconnected from network and the user is using World Wide Web transaction, then the user need to do the activities from the first step. This will show that the third objective which is to study that the mobile agent technology can have asynchronous computing have been figured out.

4.2 Discussion

From all the collected information about auction using mobile agent through online researches, journals, and books, here is the selected topic that can be observed thoroughly. The article discussed about mobile agent technology for dynamic stock exchange and many points have been found are related with e-auction system for PDA's.

Below are some interested findings which can be used in this proposed project. Stock exchange is a type of trading where time and price become part of the same equation. The stock market rotates around stocks, which are the objects being traded, and people who are the consumers of these objects. The whole exchange operation focuses around the time when a trade should take place. What the objects stand for has far less importance than how the objects behave. The whole trading system is based upon the interaction between the objects and the consumers. The objects' behavior triggers specific events according to which the consumers take action. For a trade to take place the following requirements have to be observed:

- **Location**: Both the objects and the consumers must come together in a location where the trading occurs. This location acts as a hub or a container within which transactions take place.
- **Event**: Object behavior triggers events that generate a potential for a trade.
- **Decision**: The trade potential goes through a an intelligent decision making process where it is translated into a resolution of whether or not to conclude the trade.
- **Agreement**: The terms of the transaction are agreed upon by the parties involved.
- **Trade**: The transaction takes place.

As the above terms are fulfilled, the transaction is concluded. Each transaction loops back into the system as an event and as a trigger for other events. The above discussion show that e-auction system using mobile agent technology must also be familiar with those terms and the use of intelligent mobile agents in the construction of an electronic

auction environment. In addition, agents are built as an extension of the consumers as they are given the power to negotiate, make decisions, and conclude transactions.

There is another selected article or journal that closely related to this project interested to be discussed as it gives more deeply understanding to produce the e-auction system. Entitled secure auction for mobile agent, add on one case study which derived to auction for mobile agent. The idea behind the article is there exist two main agent types, referred to as a shopper agent and an auctioneer agent, which are used in auction events. Both agent type are descendants of the agent super class and have similar functions, for example, the same communication interfaces. Both of the agents also use the same agent language that enables co-operation between them.

4.2.1 Shopper Agent

A shopper agent is the most intelligent agent in the system. Its main goal is to buy PDA, which is author is interested in. When a user creates a shopper agent, he/she is requested to fill up a profile of his/her own interest. After a shopper agent has analysed the profile, it uses the information received to decide if the user is interested in a particular PDA that has been offered by the auctioneer agent. Thus, the agent can filter out the entire PDA offers are forwarded to the agent's author, who is asked to conform his/her interest in the PDA. If the user accepts the request, a shopper agent will migrate to the auction place to take part in the auction. An agent's ability to autonomously make decisions can also be used in the bidding situation, where the agent itself decides, according to the user profile, how high bids it should place. If the author does not feel confident on an agent's ability to decide on the bidding price, he or she can set an absolute limit price that will not be exceeded by the agent.

4.2.1 Auctioneer Agent

The purpose of the auctioneer agent is to sell PDA's to the shopper agents and try to get as high a price as possible. The auctioneer agent knows the PDA's that it is auctioning

and knows their value. It may have a minimum price for PDA's so that if shoppers are not willing to bid above that, it will buy the PDA's back for itself. The auctioneer can receive assignments from clients containing PDA' lists with numerous PDA are on them. Those PDA's are then auctioned one by one from the list and the results of the auction are transmitted to the client. In addition to managing the auction event, the auctioneer sends adverts of PDA's, which are going under the hammer. Adverts are only sent to those shopper agents that are registered with the auction in order to reduce network traffic. The auctioneer agent is a stationary agent and it does not have such a complex profile as shopper agents have. Therefore, implementation of the agent is simpler and it is smaller in size.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

This chapter discuss about the relevancy to the 4 main objectives of this project. Later in this section will also include the suggestion future work for expansion and continuation.

5.1. CONCLUSION

Mobile agent technology is a concept that stretches the limits of distributed computing. This paper discusses the utilization of intelligent mobile agents in e-auction system. It presents the construction of the Aglets which is a software entity that allows for real-time PDA's e-auction through the use of agentified traders. The system contains all the components necessary to carry out successful and fair trading for different types of PDA's. Agents are developed externally then sent to the e-auction where they are verified, registered and admitted to the system. An agent is an autonomous entity with the ability to control its own actions and make decisions. Such authority makes trading possible between traders in this virtual market. This e-auction system is including a mobile agent which is defined as software program that moves from machine to machine under its own control, it can suspend execution at any point in time, transport itself to a new machine and resume execution. Once created, a mobile agent autonomously decides which locations to visit and what instructions to perform Continuous interaction with the agent's originating source is not required by implicitly specified through the agent code and specified through a run-time modifiable itinerary

As a conclusion, I can say that developing the e-auction system using mobile agent technology was a challenging task for me. Without any guidance from my supervisor

(Mr. Anang), I think it is difficult for me to manage to complete this project successfully. I really need to put into considerations of the design and implementation phases. Both phases really force me to put much effort hardly. However, this project gave me so many new experiences about the new technology as well as helped in consolidate my java programming skills.

This project goal was to create a simple e-auction system using mobile agent technology. The research on this project has been done, but sincerely I still need some guidelines from the experts. Many aspects can be improved in order to complete the project goals and objectives. By the end of this project, the system should be able to:

- Produce a simple e-auction system applying the concept of using mobile agent technology
- Complete the study of mobile agent technology that can reduces communication cost
- Complete the study of mobile agent technology that allow to use limited resources
- Complete the study of mobile agent technology that have an asynchronous computing

5.2. RECOMMENDATION (suggested future work for expansion and continuation)

This E-auction system using mobile agent technology is new for me and maybe also new to University Technology of Petronas's students. Nowadays, technology emerges so fast and we as the user must cope-up with that emerging technology. This application or system can be enhanced better as well as can has a more detail research work on the mobile agent technology together with its application. Here are some potential further works for future expansion and continuation:-

1] **More research on how e-auction system implemented with mobile agent technology**

Besides aglets, there are many others mobile agent toolkits to be used in implementing the e-auction system. With all the type of toolkits provided, there will have the comparison issues in many areas in producing a better and enjoyment e-auction system.

2] **Further research and implementation of the real e-auction system**

The current product that has been produced only show the concept or basis of e-auction system. If the system could be as real as possible, of course it will be more interesting.

3] **Create another system rather than e-auction system using mobile agent technology**

As stated in the literature review, mobile agent technology can be applied in many industry areas. Therefore it will be beneficial to make full used of it by implementing the mobile agent technology with other program or system.

4] **Research on security of mobile agent technology**

Because mobile agent technology involved with network, it is exposes to many risk such as in security issue. This security issue is very broad area to be discussed. Therefore it will be a chance to other researchers to involve with this research project

REFERENCES

[1] [KhaDia] Khaled El-Sawi, Dr. Dia Ali, "Mobile Agent Technology for Dynamic Stock Exchange". URL: http:www.elsawi.com/research/stock.htm

[2] [Wallin2004] Arto Wellin, "Secure Auction for Mobile Agents", VTT Publications 538, Espoo 2004. URL: http://www.vtt.fi/inf/pdf/publications/2004/P538.pdf

[3] [MiLaChau] Dejan S. Milojicic, William LaForge, Deepika Chauhan, "Mobile Objects and Agents (MOA)", The Open Group Research Institute. URL: http://www.jxml.com/papers/coots2.pdf

[4] [Sommers97] Bret Sommers, "Agents: Not Just for Bond Anymore", April 1997. URL: http://www.javaworld.com/javaworld/jw-04-1997/jw-04-agents.html
Reuven Lerner, *Content Management*, Linux Journal, 2003

[5] [LaAri97] Danny B.Lange, Yariv Aridor, "Agent Transfer Protocol – ATP/01" March 19. 1997. URL : http://www.trl.ibm.com/aglets/atp/atp.htm

[6] [Cetus-Link2002] Cetus-Link, "Distributed Objects & Components: Mobile Agents", 2002. URL: http://www.cetus-links.org/oo_mobile_agents.html

[7] [Reilly] David Reilly, "Mobile Agents – Process migration and its implications". URL: http://www.davidreilly.com/topics/software_agents/mobile_agents/

[8] [Versteeg] Steven Versteeg, "433-463 Thesis – Languages for Mobile Agents", 25 August, 1997. URL: http://www.cs.mu.oz.au/~scv/thesis.html

[9] [WurWelWal98] Peter R.Wurman, Michael P.Wellman, William E.Walsh, "The Michigan Internet AuctionBot: A Configurable Auction Server for Human and Software Agents".

APPENDICES

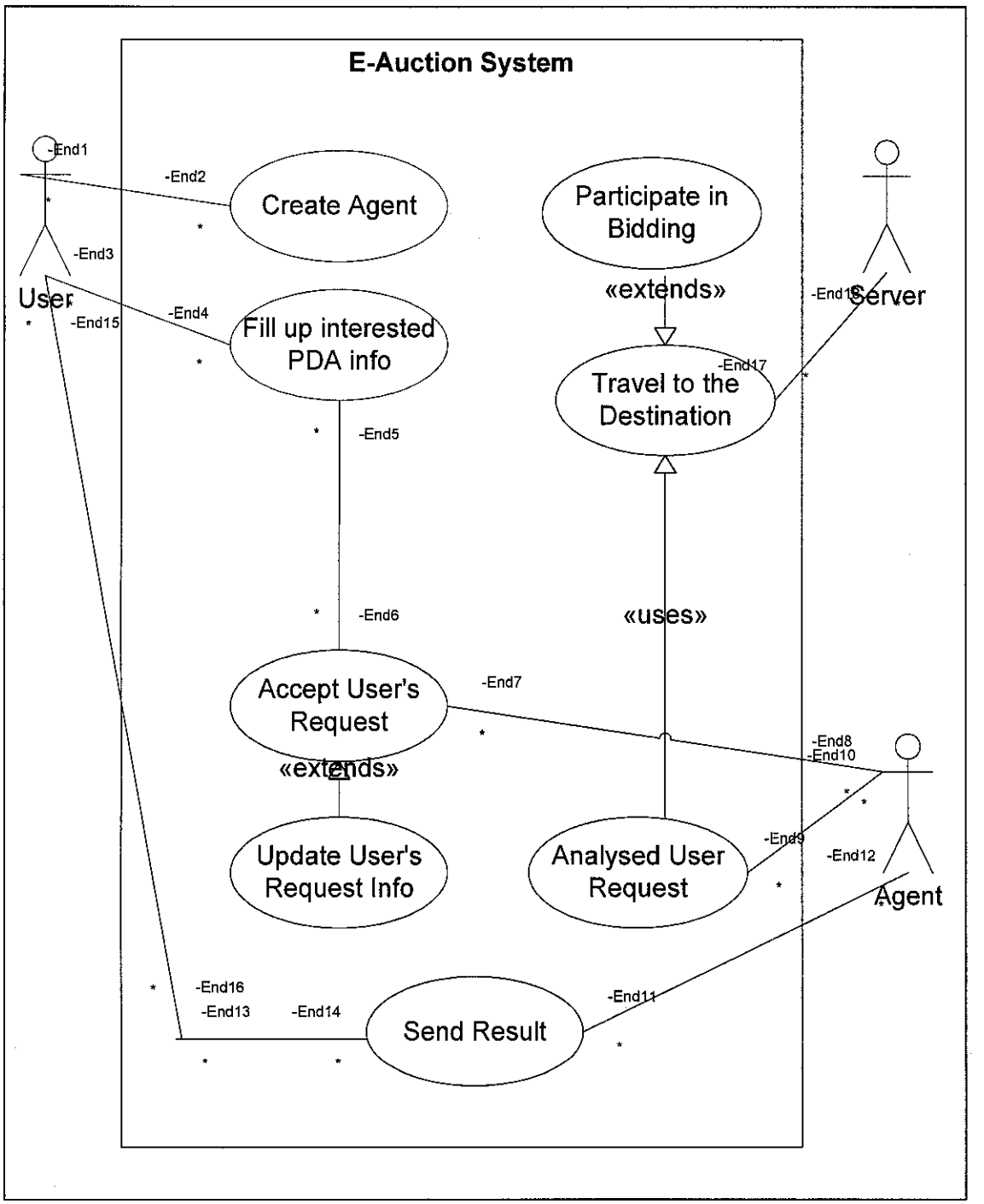


Figure: Use Case of E-Auction System

Sequence Diagram

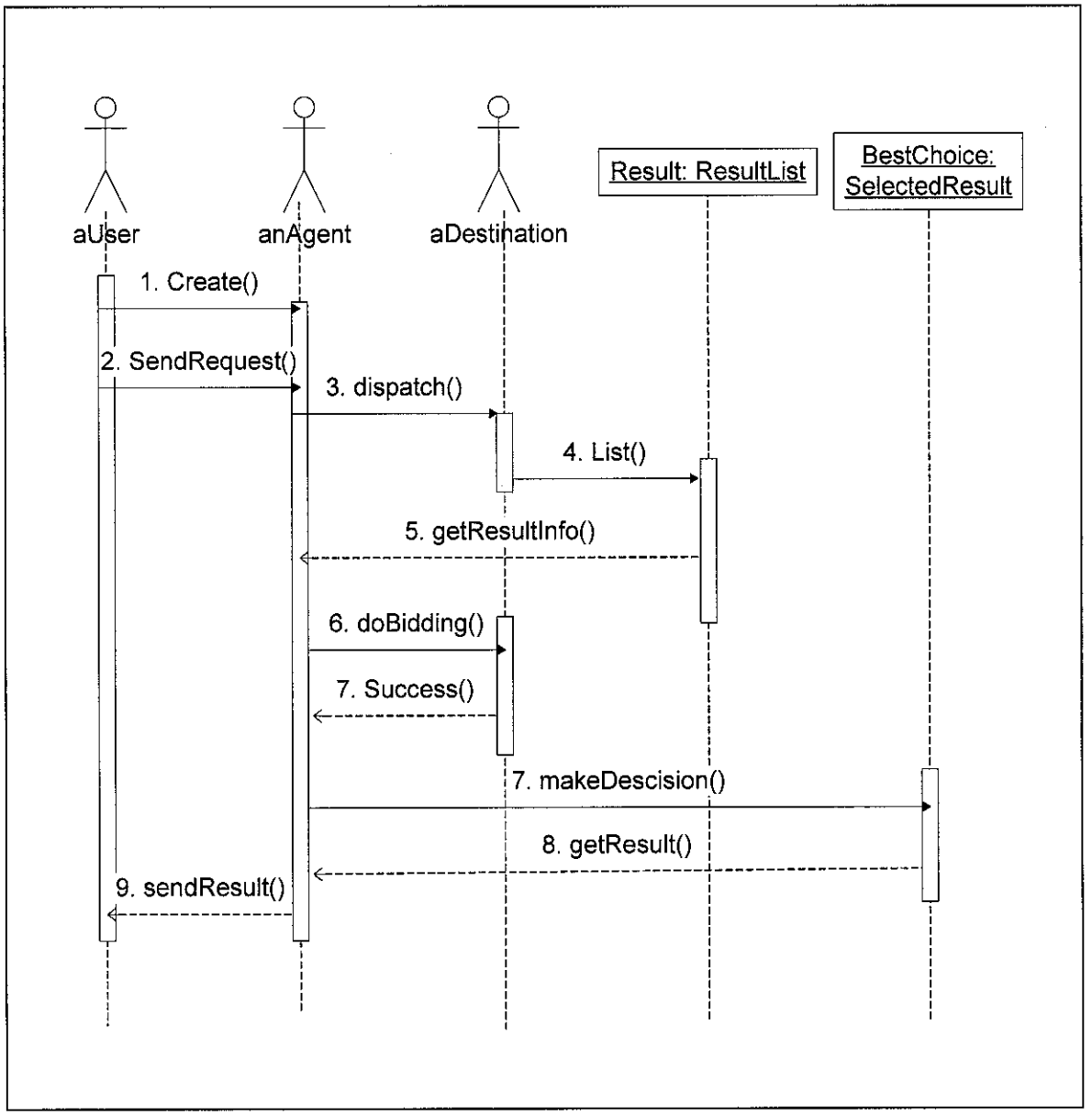
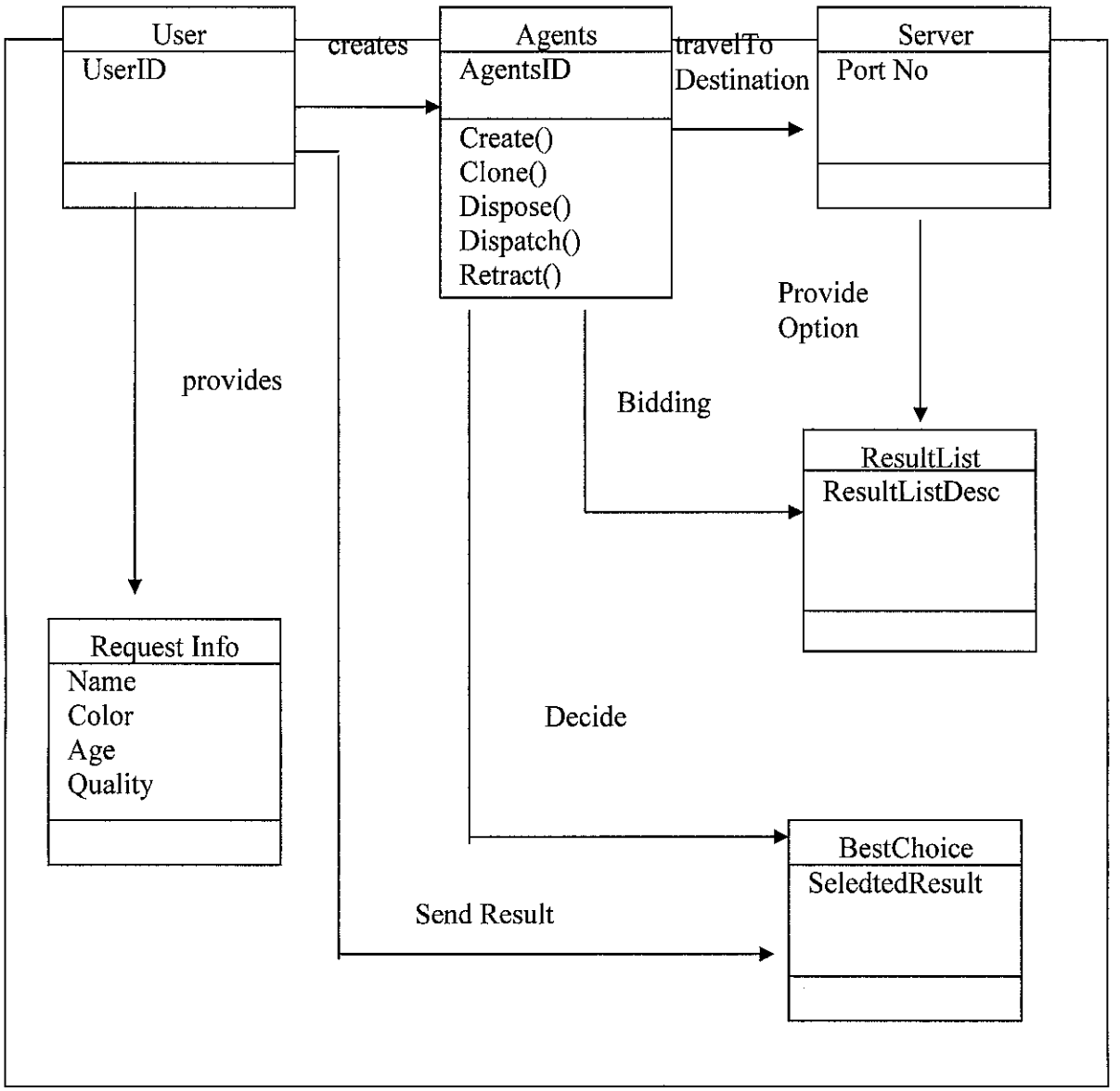
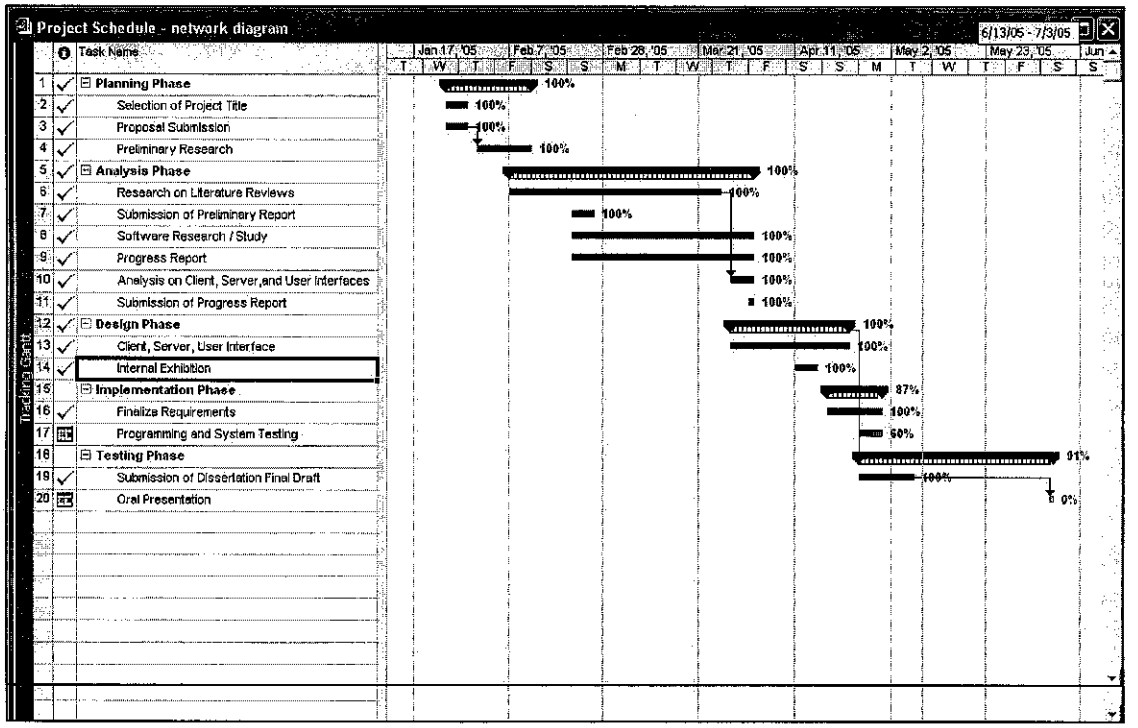


Figure : Sequence Diagram

Class Diagram



Gantt Chart



E-Auction System's Project Schedule

```
// USER INTERFACE FOR USER REQUEST INFO TABLE IN ACCESS WITH INSERT, UPDATE,
DELETE, VIEW,
// FIRST, NEXT, PREVIOUS. LAST, SAVE, AND EXIT BUTTONS.


package UserReq.agents;


import java.awt.*;

import java.awt.event.*;

import java.io.*;

import java.lang.*;

import java.lang.InterruptedException;

import java.net.*;

import java.util.*;

import java.util.Enumeration;

import java.io.Externalizable;

import java.io.ObjectInput;

import java.io.ObjectOutput;

import java.io.IOException;

import javax.swing.*;

import com.ibm.aglet.*;

import com.ibm.aglet.event.*;

import com.ibm.aglet.util.*;

import com.ibm.agletx.util.SimpleItinerary;

onArrival()


       //IMPORTING THE REQUIRED PACKAGES.
       import java.awt.*;
       import java.sql.*;
       import java.util.Vector;
       import javax.swing.*;
       import java.awt.event.*;



          public void run() {
       if (!_theRemote) {
                     System.out.println(who() + "\'run()\' is starting...");
                  pause();
                  System.out.println(who() + "Dispatching \'" + PREFIX +
"\'...");
             try {
```

```
                    String host = getAgletContext().getHostingURL().toString();
                    URL destination = new
URL((String)getAgletContext().getProperty("location", host));
                    System.out.println(who() + "Destination is \'" +
destination.toString() + "\'");
                    dispatch(destination);
                     System.out.println(who() + "You should never see this on
your console!");
              } catch (Exception e) {
                    System.out.println(who() + "Failed to dispatch \'" + PREFIX
+ "\'.");
                    System.out.println(e.getMessage());
              }
                    pause(); pause();
                    System.out.println(who() + "\'run()\' is finishing.");
       } else {
                    System.out.println(who() + "\'run()\' is starting...");
                    pause();
                     System.out.println(who() + "\'run()\' is finishing.");
              }
        }
       UserReq userReq1 = new UserReq()

       public class UserReq1 extends JPanel
       {
              // DECLARATION OF VARIABLES
              String pname,pcolor,pprice,pq,pldate;
              String pnam;

              //DEFINITIONS OF JBUTTONS.
              JButton      btnInsert;
              JButton      btnDelete;
              JButton      btnUpdate;
              JButton      btnView;

              JButton      btnFirst;
              JButton      btnPrev;
              JButton      btnNext;
              JButton      btnLast;

              JButton      btnEnter;
              JButton      btnExit;

              // DEFINITIONS OF JLABELS.
              JLabel              lblHead;
              JLabel              lblName;
              JLabel              lblColor;
              JLabel              lblPrice;
              JLabel              lblPq;

              // DEFINITIONS OF JTEXTFIELDS.
              JTextField   txtName;
              JTextField   txtColor;
              JTextField   txtPrice;
              JTextField   txtPq;


              //CONSTRUCTOR OF THE CLASS USERREQ
              public UserReq()
              {
                     btnInsert    = new JButton("Insert");
                     btnDelete    = new JButton("Delete");
                     btnUpdate    = new JButton("Update");
```

```
                    btnView      = new JButton("View");
                    btnFirst     = new JButton("First");
                    btnPrev      = new JButton("<<");
                    btnNext      = new JButton(">>");
                    btnLast            = new JButton("Last");
                    btnEnter     = new JButton("Enter");
                    btnExit      = new JButton("Exit");

                    // ADDING SHORTCUTS TO THE BUTTON
                    btnInsert.setMnemonic('I');
                    btnInsert.setToolTipText("Adds a new Record to the Table");

                    btnDelete.setMnemonic('D');
                    btnDelete.setToolTipText("Deletes an Existing Record from
the Table");

                    btnUpdate.setMnemonic('U');
                    btnUpdate.setToolTipText("Updates a new Record to the
Table");

                    btnView.setMnemonic('V');
                    btnView.setToolTipText("To View the Records in the Table");

                    btnFirst.setMnemonic('F');
                    btnFirst.setToolTipText("To View the First Records in the
Table");

                    btnPrev.setMnemonic('P');
                    btnPrev.setToolTipText("To View the Previous Records");

                    btnNext.setMnemonic('N');
                    btnNext.setToolTipText("To View the Next Records in the
Table");

                    btnLast.setMnemonic('L');
                    btnLast.setToolTipText("To View the Last Records in the
Table");

                    btnEnter.setMnemonic('E');
                    btnEnter.setToolTipText("Enter");

                    btnExit.setMnemonic('X');
                    btnExit.setToolTipText("Close");

                    lblHead      = new JLabel(" USER REQUEST ");
                    lblName      = new JLabel(" Product Name ");
                    lblColor     = new JLabel(" Product Color ");
                    lblPrice     = new JLabel(" Product Price ");
                    lblPq        = new JLabel(" Product Quality ");
                    //lblLdate   = new JLabel(" Product Last Date ");

                    txtName            = new JTextField(70);
                    txtColor     = new JTextField(10);
                    txtPrice     = new JTextField(10);
                    txtPq        = new JTextField(20);
                    //txtLdate   = new JTextField(10);

                    setLayout(null);

                    // PLACING ALL THE OBJECTS ON THE PARENT CONTAINER I.E
FRAME AT DESIRED POSITIONS

                    //HEADING LABEL
```

```
Font font = new Font("COMIC CANS SERIF",Font.BOLD, 18);
setFont(font);
lblHead.setBounds(230,5,375,25);
lblHead.setFont(font);
add(lblHead);

//INSERT BUTTON
btnInsert.setBounds(160,40,75,25);
add(btnInsert);

//UODATE BUTTON
btnUpdate.setBounds(240,40,75,25);
add(btnUpdate);

//DELETE BUTTON
btnDelete.setBounds(320,40,75,25);
add(btnDelete);

//VIEW BUTTON
btnView.setBounds(400,40,75,25);
add(btnView);

//FIRST BUTTON
btnFirst.setBounds(160,40,75,25);
add(btnFirst);

//PREVIOUS BUTTON
btnPrev.setBounds(240,80,75,25);
add(btnPrev);

//NEXT BUTTON
btnNext.setBounds(320,80,75,25);
add(btnNext);

//LAST BUTTON
btnLast.setBounds(400,80,75,25);
add(btnLast);

//PRODUCT NAME
lblName.setBounds(160,120,100,25);
add(lblName);
txtName.setBounds(280,120,200,25);
txtName.setToolTipText("Brand");
add(txtName);

//PRODUCT COLOR
lblColor.setBounds(160,155,100,25);
add(lblColor);
txtColor.setBounds(280,155,200,25);
txtColor.setToolTipText("Color");
add(txtColor);

//PRODUCT PRICE
lblPrice.setBounds(160,190,100,25);
add(lblPrice);
txtPrice.setBounds(280,190,200,25);
txtPrice.setToolTipText("Price Range");
add(txtPrice);

//PRODUCT QUALITY
lblPq.setBounds(160,225,100,25);
add(lblPq);
txtPq.setBounds(280,225,200,25);
```

```
            txtPq.setToolTipText("Quality Range");
            add(txtPq);

            //PRODUCT Last Date
            //lblPq.setBounds(160,225,100,25);
            //add(lblPq);
            //txtPq.setBounds(280,225,200,25);
            //txtPq.setToolTipText("Quality Range");
            //add(txtPq);

            //ENTER BUTTON
            btnEnter.setBounds(200,335,75,25);
            add(btnEnter);

            //EXIT BUTTON
            btnExit.setBounds(350,335,75,25);
            add(btnExit);
      }
      // THE MAIN FUNCTION WHICH INVOKES THE USERREQ.JAVA FILE

      public static void main(String args[])
      {
            // INSTANTIATING AN OBJECT OF USERREQ
            UserReq Req   = new UserReq();
            JFrame frm    = new JFrame("User Request");
            frm.setContentPane(Req);
            frm.setVisible(true);
            frm.setSize(650,400);
            frm.setResizable(false);

            WindowListener winListener = new WindowAdapter()
                  {
                        public void windowClosing(WindowEvent winEvt)
                        {
                              System.exit(0);
                        }
                  };    // END OF WINDOWADAPTER() METHOD

                        frm.addWindowListener(winListener);
      }
}     // END OF USER INTERFACE PROGRAM....
```