WXES 3182: PROJECT LATIHAN ILMIAH TAHAP AKHIR II

"INTELLIGENT MARKING SYSTEM FOR STRUCTURED QUESTION"

MASTURA ARIFIN

WEK 98151

SUPERVISED BY MISS NORISMA IDRIS

MODERATED BY DR SYED MALEK FAKAR DUANI

Dissertation submitted in partial fulfillment of the requirement for the Degree of Bachelor in Computer Science ,Faculty of Computer Science and Information Technology University Malaya.

Submission date : 22 February 2002

Session : 2001/2002

TABLE OF CONTENT

Abstract

As part of the graduation exercise, it is required to develop as system for the Degree of Bachelor in Computer Science and Information Technology, University of Malaya. The system is intended to develop an ***Intelligent Marking System for Structured Question***.The Intelligent Marking System hopes to overcome the manual marking that are done for many years. The Intelligent Marking System aims to give a fast response and marks to the answer that are answered

The literature review gives an overview of the marking system and findings of various researches and technology in the effectiveness of the intelligent marking system. Some of the literature surveys done are examined and their features compared. In this project, sources were collected from various part including the Internet ,books and the senior project.

The Intelligent Marking System developed through a modified version combining the benefits of both waterfall model and prototyping approach. The Artificial Intelligence technique,such as The Expert System- The Rule Based System,The Information Retrieval and keyowrd matching used for the developing tools for the Intelligent Marking System while the software used is JBuilder 5.

The selection of both hardware and software are vital to ensure the success of a system. Criteria such as capability, credibility, cost speed and size of memory have been emphasized to ensure the hardware-selected meet the requirements of the system to be developed.

System design is very essential in a system development process as it plays a major role in determining the success of the system.The Intelligent Marking System would be maneuvered through 2 kinds of user interface- the students and the marking system (the lecturers).

Exhaustive testing was done on the application when it was completed. The testing strategies used were unit testing, integrating testing and system testing.

Finally, some of the problems faced during the development of the application and the solutions found are describes. Some suggestions for the future enhancements on the system are also suggested.

Acknowledgment

The development of ***The Intelligent Marking System***, is developed through the advice, assistance and contribution of many individuals. Firstly, I would like to express my utmost gratitude to my project supervisor, ***Miss Norisma bt Idris,*** who has provided me with full support and guidance throughout the whole development phase of my system.

To my moderator, ***Dr Syed Malek Fakar Duani bin Syed Mustapha***, for giving me suggestion regarding the appropriate tools to choose for the system.

Sincere thanks to all my friends especially to ***Mahathir bin Abang Abang*** that always be there to help and guide me throughout this project development.Not forgetting to my dearest friend,***Zahini Lokman*** ,that always be there for me through my thick and thin .

Last but not least, my sincere gratitude to both my parents and also my family members for being supportive, motivate and caring throughout my project development.

Your Sincerely,

Mastura Bt. Arifin

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

Chapter 1: Introduction

1.0 Introduction to the project

This introductory chapter gives a description or purpose of the project and problems to be solved. The significance and rationale of the project will be discussed here. Furthermore the system functions, limitation and its assumption will also be dwelt into later.

1.1 Introduction to the Marking System

As we have move towards the future ,time is very important for each and everyone of us.Due to limited teaching resources and the large number of students, instructors would welcome an IT-tool that can be used any time to mark the students' coursework.[ipdweb.np.edu.sg/lt/feb00/innov.htm]

Writing is an essential part of the educational process, many lecturers find it difficult to incorporate large numbers of writing assignments in their courses due to the effort required to evaluate them. The liability to convey information verbally is an important educational achievement in its own right, and on that is nor sufficiently well assessed by other kinds of tests. In addition the testing is thought to encourage a better conceptual understanding of the material and to reflect a deeper, more useful level of knowledge and application by students.

Thus grading and commenting on written texts is important not only as an assessment method, but also as a feedback device to help students better learn both content and the skills of thinking and writing.[http://imej.wfu.edu/articles]

Buliding a system that can mark the structured answer question will benefits both the students and lecturer.

1.2 Scope

The function of this system is to provide an agent to help the users to do the marking based on their requirement.Basically the system will help the user to do all the checking and marking on the short answer question.

The user of the system maybe a teacher,lecturer ,tutor or even the student itself.This project typically only covers one particular subject,which is ***Biology***, but it can be implemented by other subject that have similiraity with this .

1.3 Objective

The broad objective of this thesis is to describe a collection of knowledge based that is constructed for the marking.This collection of knowledge known as intellignet marking system will assist user in knowing how to do the marking process of the structured question given.The objective of the intelligent marking system are the following:

- To assist the teachers,lecturer,tutor and student to do the marking process

- To reduce the time marking the paper manually
- To suggest the marks that will be given to the marked paper.
- Be user friendly:easy to use and helpful.
- To speed up the generation of the marking process.
- The make the system more be reliable and not proned to crashes.
- The system should be able to handle and report error,since many applications involve uncertainty information.In education,uncertainty may mean something which has not been discovered.

1.4 Schedule of Activities

A project schedules was prepared at the beginning stage of the project to insure that this project will be completed in time. Basically, this project is divided into 8 stages and the activities for each stage are as follows:

Stage	**Activity**
1. Preliminary investigation	• Identify project objective • Identify system scope
2. Study programming tools	• Visual Prologue or Visual Basic
3. System planning	• Determine system requirement • Prepare project schedule
4. System Design	• Design data dictionary • Design screen formats

5. System implementation	• Code system module
6. System testing	• Design test data
7. System maintenance	• Make necessary changes to the system
8. Documentation and report	• Prepare project report • Prepare user manual

Table 1.1: Activities in each stage

A Gantt Chart is an easy way to schedule tasks. It is essentially is a chart on which bars represent each task or activity. The length of each bar represents the relative length of the task. The Gantt Chart for this project is shown in Figure 1.1 below. This schedule was re-estimated from time to time during the development life cycle.

Gantt Chart

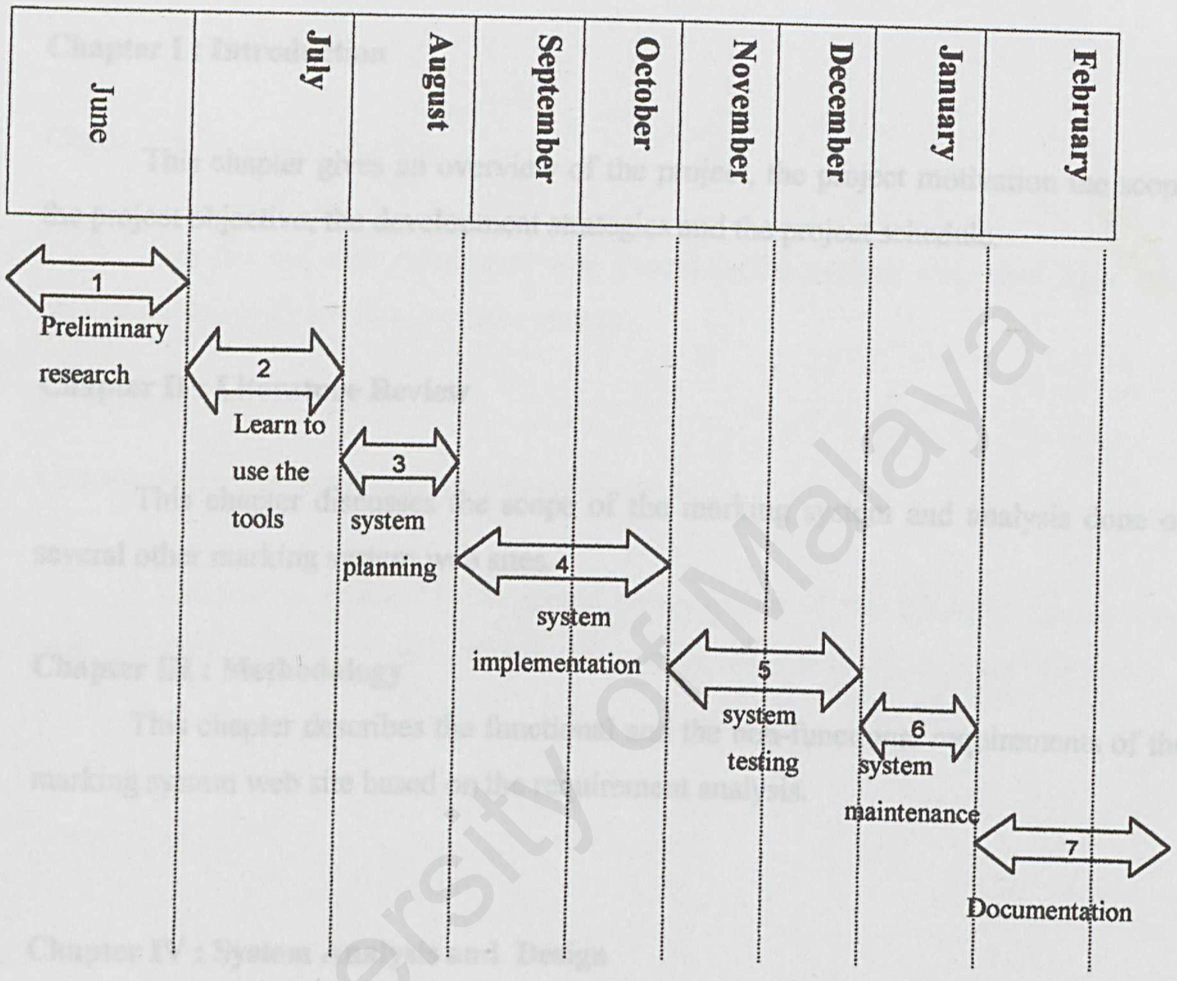


Figure 1.1 : Gantt Chart for the Marking System Tasks

1.5 Chapter Organization

This thesis is divided into the following chapters :

Chapter I : Introduction

This chapter gives an overview of the project, the project motivation the scope, the project objective, the development strategies and the project schedule.

Chapter II : Literature Review

This chapter discusses the scope of the marking system and analysis done on several other marking system web sites.

Chapter III : Methodology

This chapter describes the functional and the non-functional requirements of the marking system web site based on the requirement analysis.

Chapter IV : System Analysis and Design

This chapter describes the consideration of the web design, and the techniques used to develop the marking system.

Chapter V : System Development and Implementation

The development environment and development tools are explained in details in this chapter

Chapter VI : System Testing

This chapter also documents and describes the various tests that have been carried out to verify the development system.

Chapter VII : System Evaluation and Conclusion

Limitation encountered and future enhancement of the Marking System web page is highlighted here. Finally, the overall conclusion of the project paper included.

Summary

This chapter gives and overview of the project that includes the aims, relevance, significance and the scope of the development of an intelligent agent to the marking system. In conclusion, intelligent agent has been around for years, but the actual implementation is still in its early stages. As agents gain a wider acceptance and become more sophisticated, they will become a major factor in the future of the Internet. Intelligent agents will not completely replace surfing altogether, but they will make information gathering much easier for the users or consumer. Instead of searching through lists and lists of unwanted sites, the user could ask their agent to start searching, and in a few moments, it come back with the information that is needed immediately. Besides that, it also covers the importance and feasibility of the project as well as the project limitations and expected income.

CHAPTER 2

LITERATURE REVIEW

Chapter 2:Literature review

2.0 INTRODUCTION TO THE INTELLIGENT SYSTEM

An intelligent agent is a software that assists people and act on their behalf. Intelligent agents work by allowing people to delegate work that they could have done, to the agent software. Agents can perform repetitive tasks, remember things you forgot, intelligently summarize complex data, learn from you and even make recommendations to you.

In other word, intelligent system is a computerized program tools that mimic the humans mind. It uses the knowledge from education and experience to solve the problem. Although a computer cannot have experience or study and learn humans as the human mind do,but it can apply the knowledge given to it by human experts. It differ from expert system because expert system is vary domain specific.

Intelligent system has several capabilities and features that distinguish it from conventional computer program. Perhaps this can be viewed from their respective goals in which the basic on eis to capture and distribute the expertise of human expert.[Prevau,1990].In contrast,the goal of a conventional program is merely to implement a set of algorithm.

Recent development have shown that the capabilities of intelligent system could be improved when multiple knowledge engineering technique under hybrid environment are incorporated.[Silverman,1987].

Other conventional characteristic of intelligent system remain as are less influential in the development of the system.

2.0.1 What kind of problems Intelligent Agents can solve ?

To understand how intelligent agents work, it is best to examine some of the practical problems that intelligent agent can help solve. An intelligent agent can help you find and filter information when you are looking at corporate data or surfing the Internet and don't know where the right information is. It could also customize information to your preferences, thus saving you time of h andling it as more and more new information arrived each day on the Internet.

2.0.2 Applications of Intelligent agent

Here are some of the examples that use intelligent agent which illustrate some of the important ways intelligent agents can help solve real problems and make today's computer system easier to use.

2.0.3 Characteristic of Intelligent Agent

All agents are autonomous, which means that an agent has control over its own actions. All agents are also goal-driven. Agents have a purpose and act accordance with that purpose. There are several ways of making goals known to an agent, and are listed below:

- An agent could be driven by a script with pre-defines action which would then define the agent's goals.

- An agent could also be a program and as long as the program is driven by goals and has other characteristics of agents.
- An agent could also be driven by rules, and the rules would define the agent's goals.
- There is also embedded agent goals, such as "planning" methodologies, and in some cases the agent could change its own goals over time.

An agent could also senses changes in its environment and responds to these changes. This characteristic of the agent is at the core of delegation and automation. For example, you tell your assistant "when x happens, do y" and the agent is always wai ting for x to happen. An agent continue to work even when the user is gone, which means that an agent could run on a server, but in some cases, an agent run on the user systems.

In a Multi-Agent System, agents are social, this means that they communicate with other agents. Some agents learn or change their behavior base on their previous experiences. Some agents are mobile, meaning they move from machine to machine to be clo ser to data they may need to process and do so without network delays. Finally, some agents attempt to be believable, such that they are represented as an entity visible or audible to the user and may even have aspects of emotion or personality.

2.0.4 TYPES OF PROBLEM SOLVE BY INTELLIGENT SYSTEM

- **CONTROL**

  Control system adaptivelly governs the behaviour of a given system to meet its specification.For instance ,controlling a manufacturing process.The control system obtains data on the system's operation,intepret data to a form an understanding of the state of the system or prediction its future state,determines and executes needed adjustment,It also performs monitoring and interpretation tasks to track system's behaviour over time.

- **DESIGN**

  Design system configures objects under a set of problems constraints.The system usually performs the its tasks following a seriesof steps,each with its own specific constraints.

- **DIAGNOSIS**

  Diagnosis system infers system malfunctions or faults from observable information.Normally the diagnosis system has knowledge of possible fault conditions with means to infer whether the fault exist from information on the system observable behavior.

- **INSTRUCTION**

  Instruction system guide the education a student in a given topic. They treat the student as a system that must be diagnosed and repaired. Typically it is an interactive topic to form the understanding of the behavior.

- **INTERPRETATION**

Interpretation system produces an understanding of a situation from available information. Typically this information consists of data from such sources as sensors, instruments, test result, etc.

- **MONITORING**

Monitoring system compares observable information on the behavior of a system.

- **PLANNING**

Planning system form actions to achieve a given goal under problem constraints. Planning system must have the flexibility to change the series of planned tasks when new problem information. Planning system usually require non-monotonic reasoning.

- **PREDICTION**

Prediction system infers likely consequences from a given situation. This system attempts to predict future events using available information and a model of the problem.

- **PRESCRIPTION**

Prescription system recommends solution to a given system malfunction. This system usually first incorporate a diagnostic task to determine the nature of malfunction.

- **SELECTION**

  Selection system identifies the best choice from a list of possibilities. It work from problem specifications defined by the user and attempts to find solution that most closely matches the specifications. It is used for decision support system.

- **SIMULATION**

  Simulation systems model a processor or system to permit operational studies under various conditions. They model the various components of the system and their interaction.

2.1 Purpose of Literature Review

This review of literature describes the findings of various researches and technology in the effectiveness of the intelligent marking system. The idea knowledge and experience gained during the survey will be used in the development of the Intelligent Marking System for short answer question. Various good and relevant features are to be noted during the survey, particularly the design and interfaces methods used by vary system. Thus, the literature review provides pertinent information and validity to the research and the environment will be necessary to determine and implement the best solution.

2.2 Approach

There are several approaches to literature review. The survey include several Internet search engines and senior student projects from the document room and discussion with local consultant who was wide knowledge and experience in developing applications in the stated field.

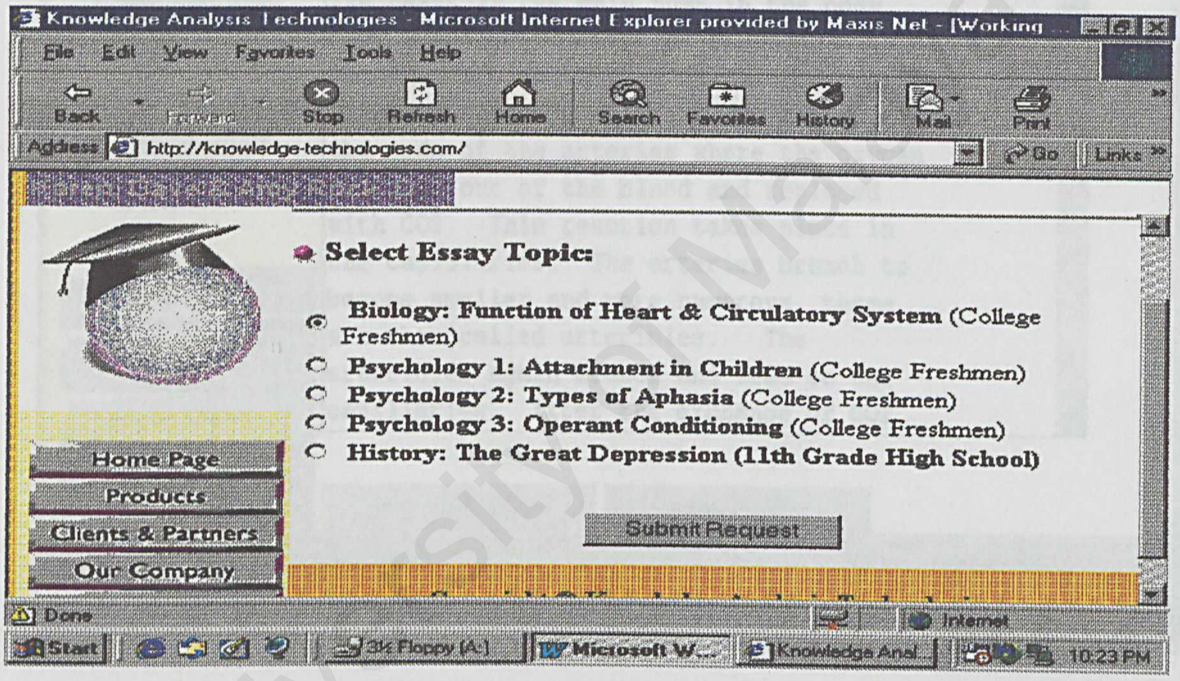
2.3 Findings

The fact-finding technique is the specific method for collecting data and relevant information pertaining to system requirements. These include interview, questionnaires and observation. In this project, research of the marking system has been done.All the data about the function of the marking system has been collected from the Internet. The data and the similar system were then stored in softcopy and hardcopy forms.Some of the literature surveys done is as follows:

2.3.1 knowledge-technologies.com

knowledge-technologies.com is one of the web-based marking system that we can found in the Internet. The basic feature in the knowledge -technologies.com includes trial part of how the user can mark the question paper.Besides that the user can also give their own answer to be marked by the system. Shown below are some example of the Intelligent Essay Accessor interface:

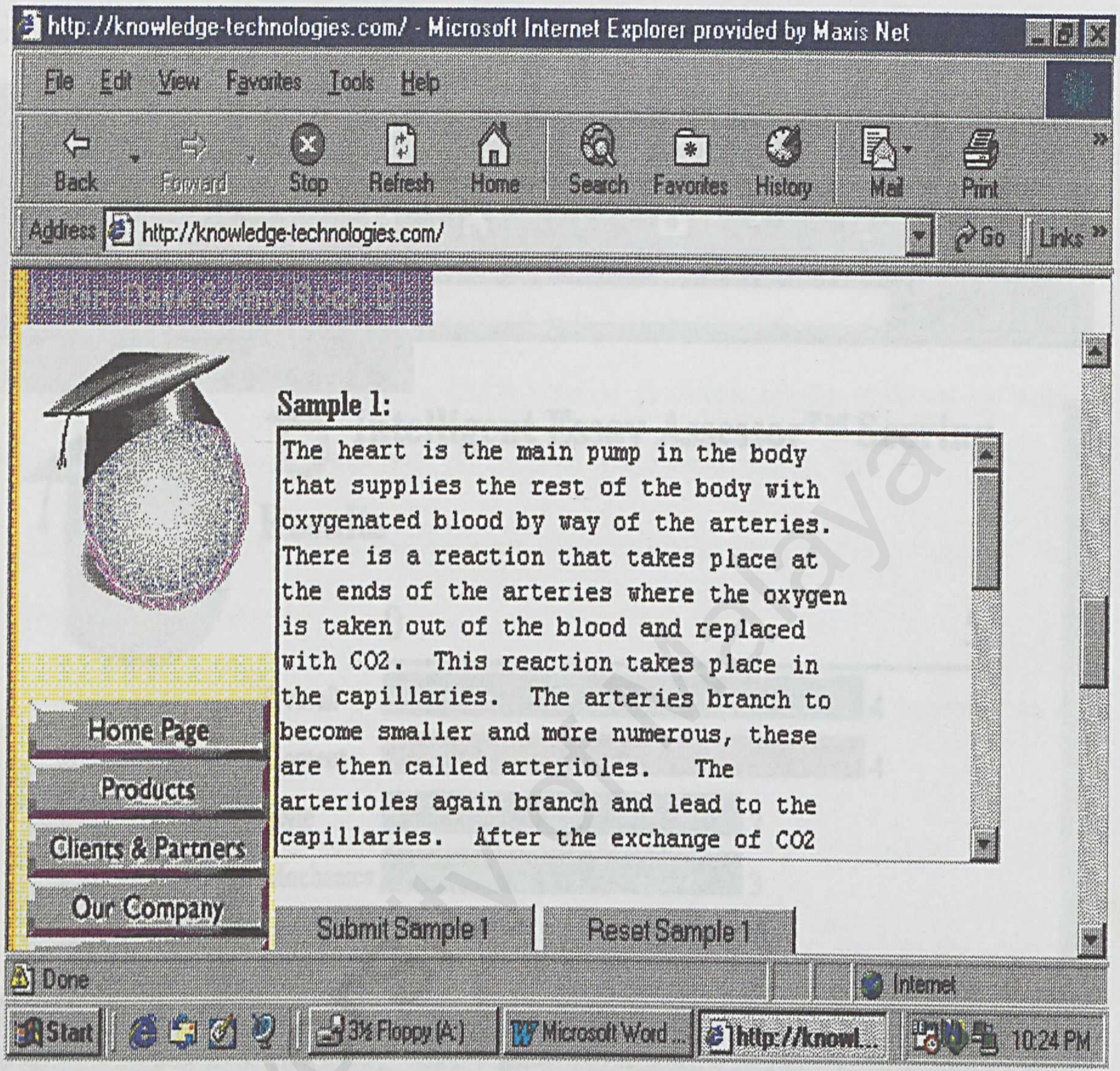
Figure 2.1-Screenshot of IEA Essay Topics



Knowledge Analysis Technologies - Microsoft Internet Explorer provided by Maxis Net - [Working ...
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address http://knowledge-technologies.com/
Go Links
Home Page
Products
Clients & Partners
Our Company
Submit Request
Done
Internet
Start
3½ Floppy (A:)
Microsoft W...
Knowledge Anal...
10:23 PM

Select Essay Topic:

- [✓ Biology: Function of Heart & Circulatory System (College Freshmen)]
- Psychology 1: Attachment in Children (College Freshmen)
- Psychology 2: Types of Aphasia (College Freshmen)
- Psychology 3: Operant Conditioning (College Freshmen)
- History: The Great Depression (11th Grade High School)


http://knowledge-technologies.com/ - Microsoft Internet Explorer provided by Maxis Net
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address http://knowledge-technologies.com/
Go Links
Home Page
Products
Clients & Partners
Our Company
Submit Sample 1
Reset Sample 1
Done
Internet
Start
3½ Floppy (A:)
Microsoft Word ...
http://knowl...
10:24 PM

Sample 1:

The heart is the main pump in the body that supplies the rest of the body with oxygenated blood by way of the arteries. There is a reaction that takes place at the ends of the arteries where the oxygen is taken out of the blood and replaced with CO2. This reaction takes place in the capillaries. The arteries branch to become smaller and more numerous, these are then called arterioles. The arterioles again branch and lead to the capillaries. After the exchange of CO2

Figure 2.2 – Screenshot Essay Sample for IEA

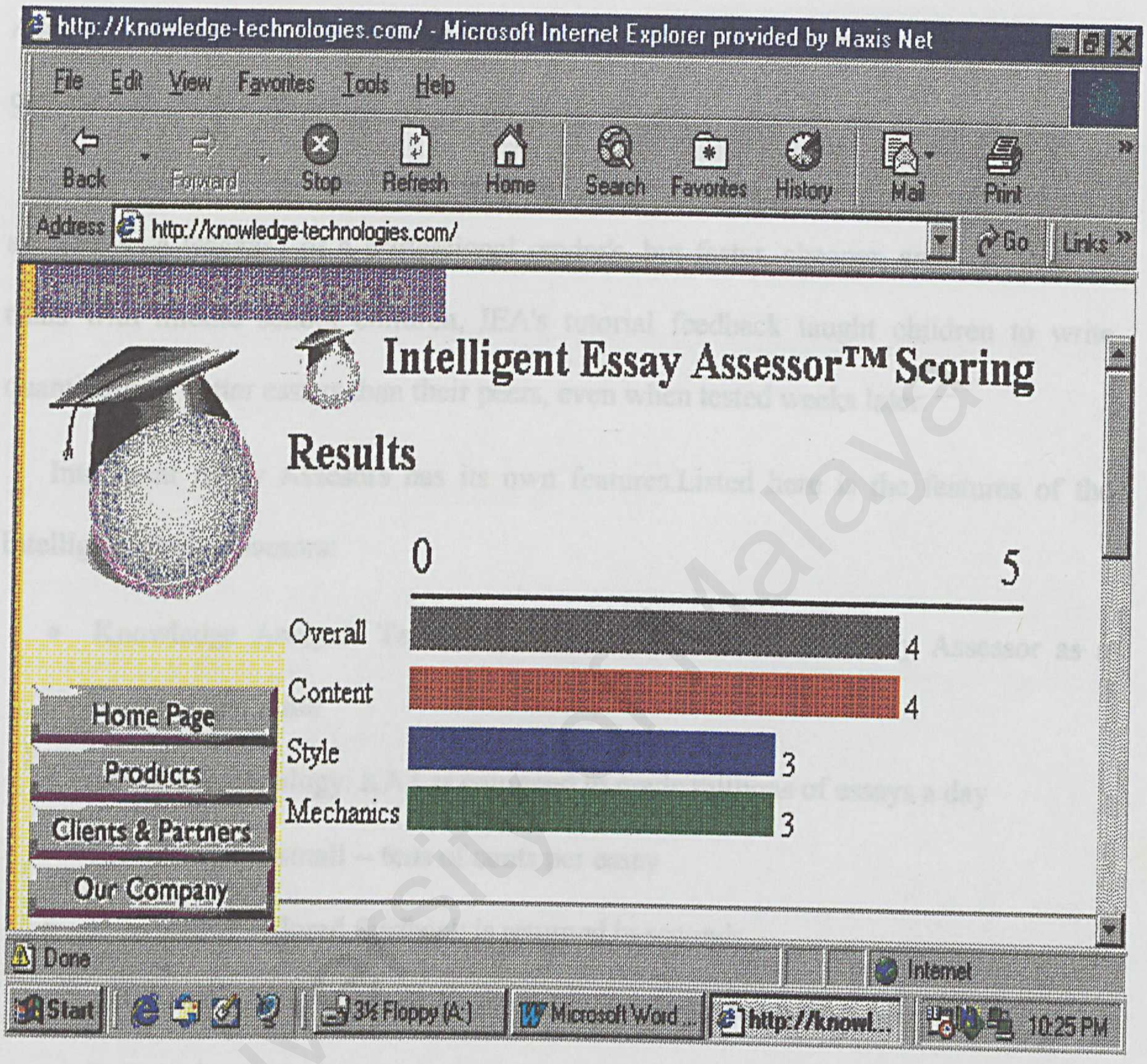

http://knowledge-technologies.com/ - Microsoft Internet Explorer provided by Maxis Net
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address http://knowledge-technologies.com/
Go Links
Intelligent Essay Assessor™ Scoring Results
0
5
Overall
4
Content
4
Style
3
Mechanics
3
Home Page
Products
Clients & Partners
Our Company
Done
Internet
Start
3½ Floppy (A:)
Microsoft Word ...
http://knowl...
10:25 PM

Figure 2.3 – Screenshot Marks for IEA

Basically the Intelligent Essay Assessor (IEA) works automatically by assessing and critiques electronically submitted text essays, giving teachers and students a new learning tool, useful in almost every subject. IEA helps to gives teacher interaction with an independent learning tool for knowledge acquisition. Students in the other hands can put what they know into words and find out how well they've done. Intelligent Essay Assessor operates as a web-based service, and supply instantaneous feedback on the content and conceptual quality of the student's writing.

Extensive research and testing of IEA's assessment capability has proven it to be as valid as a teacher's or a professional grader's, but faster, cheaper, and effortless. In trials with middle school children, IEA's tutorial feedback taught children to write quantitatively better essays than their peers, even when tested weeks later.

Intelligent Essay Assesors has its own features.Listed here is the features of the Intelligent Essay Assesors:

- Knowledge Analysis Technologies offers the Intelligent Essay Assessor as a hosted application
- Scalable technology: KAT is equipped to grade millions of essays a day
- Service cost is small -- tens of cents per essay
- Individually tailored feedback is returned in seconds
- Able to provide practice and drill with a textbook and study guide
- Built in detectors for: plagiarism, abnormal English, highly unusual essays

To develop the Intelligent Essay Assessor it must use some technology.For Developing The IEA, it uses Latent Semantic Analysis, a machine-learning algorithm that accurately mimics human understanding of language. This patented and proprietary technology is based on over 10 years of corporate and university research and development. IEA analyzes the body of text from which people learn to derive an understanding of essays on that topic. The algorithm is highly compute intensive, requiring over a gigabyte of RAM, which is why IEA is offered as a web-based service.

For the purpose of the Intelligent Marking System IEA features two means of scoring essays:

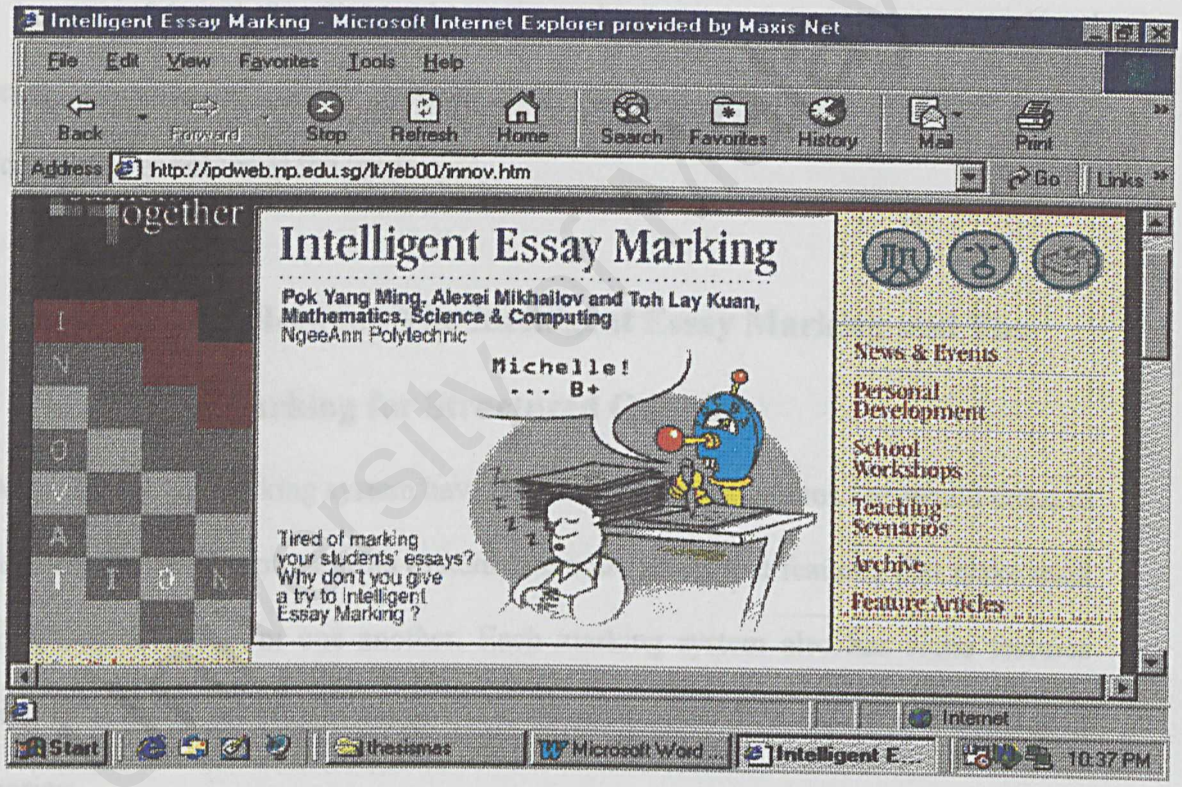
- Comparison of a new essay with samples of human graded essays for a particular question
- Comparison of the new essay with electronic versions of textbooks or other material used by students

The Intelligent essay Assesors has its own advantage:

- IEA is the only grading application on the market that measures meaning and content. Others just use surface features of the essay, such as length, spelling, grammar, and lists of keywords.

2.3.2 INTELLIGENT ESSAY MARKING BY THE STUDENTS OF NGEE ANN POLYTECHNIUQE

The Intelligent Essay Marking System is based on the Pattern Indexing Neural Network (the Indextron) developed at NGEE ANN Polytechnic. Shown below is the figure of the Intelligent Essay Marking by the NGEE ANN Polytechnique.


Intelligent Essay Marking - Microsoft Internet Explorer provided by Maxis Net

Address http://ipdweb.np.edu.sg/lt/feb00/innov.htm

Intelligent Essay Marking

Pok Yang Ming, Alexei Mikhailov and Toh Lay Kuan, Mathematics, Science & Computing
NgeeAnn Polytechnic

Michelle! ... B+

Tired of marking your students' essays? Why don't you give a try to Intelligent Essay Marking ?

Figure 2.4 – Screenshot for Intelligent essay by NGEE ANN Polytechnic

This Marking System has its own advantages.The advantage of this system is that it can run on an ordinary PC.Besides saving time for lecturers, both systems can be used as an intelligent tutoring system that will help students to write better by grading a paper fast and providing the feedback quickly.

The essay grading is based on qualitative type of questions rather than numerical type. Hence, the system can be used for exams in biology, psychology, history, anatomy and other non-mathematical subjects. The paper discusses the approach employed to build the Intelligent Essay Marking System and reflects the preliminary results obtained through marking a number of students' essays.

Besides being a tool for assessment, the intelligent essay marking system can also be a useful tool for diagnostic and tutoring purposes in many content-based subjects. Students can be given immediate feedback and can learn where and why they had done well or not made the grade.

2.4 Comparison Between The Intelligent Essay Marking and The Intelligent Marking for Structured Question.

Although all the marking system have basically the same features one would come to expect from an intelligent marking system, there are also some features that make each marking system out from one another. Each marking system also has some features missing from other marking system . Comparisons can be done based on a number of categories:

- Style of marking
- Interface
- Words

- **Style of Marking**

  When doing the marking paper for the student, the marks should be shown at the end of the answer. This can make the students know what is the mark the scored for each of the question they have answered.

- **Words**

  Basically the essay marking system has too many words to be checked,but the marking system that will be implement later will only cover only a short answer.This means that the answered that the student will answered will be basically about 20-40 words.

- **Interface**

  There will be two kind of interface for this structured marking system:

  - The student interface
  - The marking system interface-that will be used by the lecturers.

  The esay marking system that have been searched earlier has an interface that basicallly connect directly to the marking system,but for the marking system that will be developed later,it will be kept in the database before the lecturer do the marking process.

Summary

This chapter generally explains the literature study of the project, which carried out since the beginning of the semester. This chapter reviewed the design and architecture of the intelligent system. It show how the intelligent system works with the system.

The findings from this literature review will be used in the next stage of developing The Intelligent Marking System for Structured Question to help determine what tools and techniques should be use to develop the system.

The next chapter regards project planning and methodology which has helped to identify the functional and non-functional requirements of the system .Hardware requirements are listed in detailed based on the software that has been carefully analyzed and evaluated, such as server consideration ,database consideration web server consideration and other developing tools.

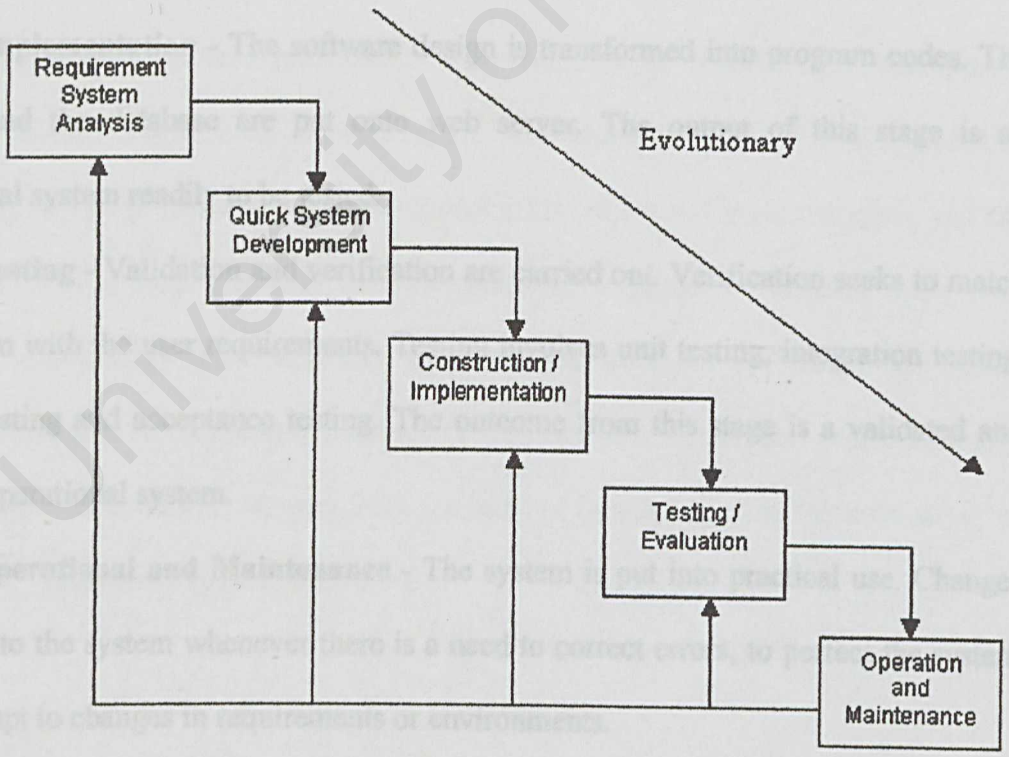
CHAPTER 3

METHODOLOGY

Chapter 3 : Methodology

3.0 Methodology for this Project

Each methodology has its own strength and weakness. The traditional waterfall model is most useful because it provides a framework of stages to ease project management and it has been well used and tested in the industry. To solve its inherent problems, prototyping needs to be integrated to recognize the iterative nature of software development. The Figure 3.1 below depicts the system development model adopted for this project.


Requirement System Analysis
Quick System Development
Construction / Implementation
Testing / Evaluation
Operation and Maintenance
Evolutionary

The Figure 3.1 : System development model adopted for this project

The methodology adopted for this project is a modified version combining the benefits of both waterfall model and prototyping approach. It is recognize that the nature of distinctive stages in the waterfall model will help determine project milestone and thus ease project management. The following explains the stages in the methodology:

1. **System Analysis** - The problem and the current manual or computerized system is analyzed. Information is collected from user. User requirements are defined and specified. A requirement specification is the output of this stage.
2. **System Design** - The requirements are mapped into system design. Design issues are decided upon here. This involves input, output, database, user interface and design in other aspects. A design specification is the output of this stage.
3. **Implementation** - The software design is transformed into program codes. The system and the database are put onto web server. The output of this stage is an operational system readily to be tested.
4. **Testing** - Validation and verification are carried out. Verification seeks to match the system with the user requirements. Testing involves unit testing, integration testing, system testing and acceptance testing. The outcome from this stage is a validated and verified operational system.
5. **Operational and Maintenance** - The system is put into practical use. Changes are made to the system whenever there is a need to correct errors, to perfect the system and to adapt to changes in requirements or environments.

An evolutionary prototype is adapted to compensate the limitations of the waterfall model. In the stage of system design, it seeks to evaluate different design approaches. During the stage of implementation, it is to evolve into a full scale-final system.

Since the prototype is evolutionary in nature, it is built knowing that it will become the final system ultimately. Hence, the prototype is built in a modular fashion. Codes are reused whenever possible. Software quality is stressed and documentation is done throughout programming of the prototype.

3.1 Justification

By developing prototype functions instead of huge specifications, the user is given a real-life feedback of the end result. Accomplished through forth generation languages, this leads to a quick system design intended for full system testing, and can be enhanced over time until it is user-acceptable and optimal in terms of operations. Thus, such languages offer a unique approach that replaces the traditional system development steps in reaching a satisfactory solution in terms of end user computing. The method can be used all the way from the actor or function level to the node structure level to help decide if a system is feasible.

3.1.1 Strength

By using the prototype process, customers or designers are able to try out a requirement before agreeing to it.User can discover requirement errors or oversights early in the software process. Requirement validation process consists of seven factors: correctness, consistency, completeness, realistic, needed, verifiable and traceable.Misunderstanding between software developers and users may be identified while the system functions are demonstrated.

User services which are difficult-to-use or unclear may be found and corrected.Incomplete and/or inconsistent requirements may be found and corrected.In a short time, a working system is available to demonstrate the capabilities and usefulness of the application to management; however, this working system is limited.

The specification for a production-quality system can be derived from the prototype.Prototyping can be viewed as a risk reduction technique. Experiments have shown that prototyping reduces the number of problems connected to requirement specifications and the overall development cost may be lower if a prototype is developed.

3.1.2 Weakness

Progress is difficult to measure since it is not visible. Also, if the systems are developed rapidly, documentation is usually not created to reflect each version of the system.

The system may be poorly structured. The constant change is harmful to the software structure.Special skills are often required. Small teams of talented and motivated individuals are needed to succeed with prototyping.

Major technical problems revolve around the need for rapid software developments.The prototype may not correspond with the way in which the final system is used, especially with throw-away prototyping.

Many software project managers are inexperienced in the areas of planning, cost, and estimating a prototyping project. Change procedures may not suitable for controlling rapid changes or development.

Prototype users or evaluators may be pressured by managers to draw quick conclusion concerning the prototype.The cost of prototyping represents a large portion of the total development costs. However, effective prototyping can increase the software quality.

3.2 Introduction to Information Retrieval

Information retrieval (IR) systems deal mainly with textual records written in a particular language. The records may be full-text documents, abstracts or just titles with -bibliographic information. Even multimedia databases contain a substantial amount of text. Natural language text expresses concepts and the relations that hold between the concepts. It is therefore surprising that the retrieval methods used in most IR systems focus on keywords or concepts, and largely ignore the relations between concepts.

Since the 1940s the problem of information storage and retrieval has attracted increasing attention. It is simply stated: we have vast amounts of information to which accurate and speedy access is becoming ever more difficult. One effect of this is that relevant information gets ignored since it is never uncovered, which in turn leads to much duplication of work and effort. With the advent of computers, a great deal of thought has been given to using them to provide rapid and intelligent retrieval systems. In libraries, many of which certainly have an information storage and retrieval problem, some of the more mundane tasks, such as cataloguing and general administration, have successfully been taken over by computers. However, the problem of effective retrieval remains largely unsolved.

In principle, information storage and retrieval is simple. Suppose there is a store of documents and a person (user of the store) formulates a question (request or query) to which the answer is a set of documents satisfying the information need expressed by his question. He can obtain the set by reading all the documents in the store, retaining the relevant documents and discarding all the others. In a sense, this constitutes 'perfect' retrieval. This solution is obviously impracticable. A user either does not have the time or does not wish to spend the time reading the entire document collection, apart from the fact that it may be physically impossible for him to do so.

When high speed computers became available for non-numerical work, many thought that a computer would be able to 'read' an entire document collection to extract the relevant documents. It soon became apparent that using the natural language text of a document not only caused input and storage problems (it still does) but also left unsolved the intellectual problem of characterising the document content. It is conceivable that future hardware developments may make natural language input and storage more feasible. But automatic characterisation in which the software attempts to duplicate the human process of 'reading' is a very sticky problem indeed. More specifically, 'reading' involves attempting to extract information, both syntactic and semantic, from the text and using it to decide whether each document is relevant or not to a particular request. The difficulty is not only knowing how to extract the information but also how to use it to decide relevance. When the characterisation of a document is worked out, it should be such that when the document it represents is relevant to a query, it will enable the document to be retrieved in response to that query. Human indexers have traditionally characterised documents in this way when assigning index terms to documents. The indexer attempts to anticipate the kind of index terms a user would employ to retrieve each document whose content he is about to describe. Implicitly he is constructing queries for which the document is relevant. When the indexing is done automatically it is assumed that by pushing the text of a document or query through the same automatic analysis, the output will be a representation of the content, and if the document is relevant to the query, a computational procedure will show this.

Intellectually it is possible for a human to establish the relevance of a document to a query. For a computer to do this we need to construct a model within which relevance decisions can be quantified. It is interesting to note that most research in information retrieval can be shown to have been concerned with different aspects of such a model.

3.2.1 An information retrieval system

Information Retrieval have three components: input, processor and output. Starting with the input side of things.. Rather than have the computer process the natural language, an alternative approach is to have an artificial language within which allqueries and documents can be formulated.

When the retrieval system is on-line, it is possible for the user to change his request during one search session in the light of a sample retrieval, thereby, it is hoped, improving the subsequent retrieval run. Such a procedure is commonly referred to as *feedback*.

Secondly, the processor, that part of the retrieval system concerned with the retrieval process. The process may involve structuring the information in some appropriate way, such as classifying it. It will also involve performing the actual retrieval function, that is, executing the search strategy in response to a query. In the diagram, the documents have been placed in a separate box to emphasise the fact that they are not just input but can be used during the retrieval process in such a way that their structure is more correctly seen as part of the retrieval process.

Finally, we come to the output, which is usually a set of citations or document numbers. In an operational system the story ends here. However, in an experimental system it leaves the evaluation to be done.

3.2.2 IR in perspective

Although information retrieval can be subdivided in many ways, it seems that there are three main areas of research which between them make up a considerable portion of the subject. They are: content analysis, information structures, and evaluation. Briefly the first is concerned with describing the contents of documents in a form suitable for computer processing; the second with exploiting relationships between documents to improve the efficiency and effectiveness of retrieval strategies; the third with the measurement of the effectiveness of retrieval.

Thus a list of what might be called 'keywords' was derived for each document. In addition the frequency of occurrence of these words in the body of the text could also be used to indicate a degree of significance. This provided a simple weighting scheme for the 'keywords' in each list and made available a document representative in the form of a 'weighted keyword description'.

At this point, it may be convenient to elaborate on the use of 'keyword'. It has become common practice in the IR literature to refer to descriptive items extracted from text as *keywords* or *terms*. Such items are often the outcome of some process such as, for example, the gathering together of different morphological variants of the same word.

The development in information structures has been fairly recent. The main reason for the slowness of development in this area of information retrieval is that for a long time no one realised that computers would not give an acceptable retrieval time with a large document set unless some logical structure was imposed on it. In fact, owners of large data-bases are still loath to try out new organisation techniques promising faster and better retrieval. The slowness to recognise and adopt new techniques is mainly due to the scantiness of the experimental evidence backing them. The earlier experiments with document retrieval systems usually adopted a serial file organisation which, although it was efficient when a sufficiently large number of queries was processed simultaneously in a batch mode, proved inadequate if each query required a short real time response. The popular organisation to be adopted instead was the inverted file.

Evaluation of retrieval systems has proved extremely difficult.
Today effectiveness of retrieval is still mostly measured in terms of precision and recall or by measures based thereon.

3.2.3 Effective and Efficiency

Much of the research and development in information retrieval is aimed at improving the effectiveness and efficiency of retrieval. Efficiency is usually measured in terms of the computer resources used such as core, backing store, and C.P.U. time. It is difficult to measure efficiency in a machine independent way. In any case, it should be measured in conjunction with effective-ness to obtain some idea of the benefit in terms of unit cost. Effectiveness is commonly measured in terms of precision and recall. The reason for emphasising these two measures is that frequent reference is made to retrieval effectiveness

3.3 Knuth-Morris-Pratt string matching

The problem: given a (short) pattern and a (long) text, both strings, determine whether the pattern appears somewhere in the text

We look at a naïve solution. Suppose the text is in an array: char T[n] and the pattern is in another array: char P[m].

One simple method is just to try each possible position the pattern could appear in the text.

3.3.1 Naive string matching:

```
for (i=0; T[i] != '\0'; i)
{
for (j=0; T[i] != '\0' && P[j] != '\0' && T[i]==P[j]; j) ;
if (P[j] == '\0') found a match
}
```

There are two nested loops; the inner one takes O(m) iterations and the outer one takes O(n) iterations so the total time is the product, O(mn). This is slow; and have to speed it up.

In practice this works pretty well -- not usually as bad as this O(mn) worst case analysis. This is because the inner loop usually finds a mismatch quickly and move on to the next position without going through all m steps. But this method still can take O(mn) for some inputs. In one bad example, all characters in T[] are "a"s, and P[] is all "a"'s except for one "b" at the end. Then it takes m comparisons each time to discover that you don't have a match, so mn overall.

Here's a more typical example. Each row represents an iteration of the outer loop, with each character in the row representing the result of a comparison (X if the comparison was unequal). Suppose we're looking for pattern "nano" in text "banananobano".

```
 0 1 2 3 4 5 6 7 8 9 10 11

 T: b a n a n a n o b a n o

i=0: X
i=1:   X
i=2:     n a n X
i=3:       X
i=4:         n a n o
i=5:           X
i=6:             n X
i=7:               X
i=8:                 X
i=9:                   n X
i=10:                    X
```

Some of these comparisons are wasted work! For instance, after iteration i=2, we know from the comparisons we've done that T[3]="a", so there is no point comparing it to "n" in iteration i=3. And we also know that T[4]="n", so there is no point making the same comparison in iteration i=4.

3.3.2 Skipping outer iterations

The Knuth-Morris-Pratt idea is, in this sort of situation, after you've invested a lot of work making comparisons in the inner loop of the code, you know a lot about what's in the text. Specifically, if you've found a partial match of j characters starting at position i, you know what's in positions T[i]...T[i1].

We as a user can use this knowledge to save work in two ways. First, skip some iterations for which no match is possible. Try overlapping the partial match you've found with the new match you want to find:

```
i=2: n a n
i=3:   n a n o
```

Here the two placements of the pattern conflict with each other -- we know from the i=2 iteration that T[3] and T[4] are "a" and "n", so they can't be the "n" and "a" that the i=3 iteration is looking for. We can keep skipping positions until we find one that doesn't conflict:

```
i=2: n a n
i=4:     n a n o
```

Here the two "n"'s coincide. Define the *overlap* of two strings x and y to be the longest word that's a suffix of x and a prefix of y. Here the overlap of "nan" and "nano" is just "n". (We don't allow the overlap to be all of x or y, so it's not "nan"). In general the value of i we want to skip to is the one corresponding to the largest overlap with the current partial match:

3.3.3 String matching with skipped iterations:

```
i=0;
while (i<n)
{
for (j=0; T[i] != '\0' && P[j] != '\0' && T[i]==P[j]; j) ;
if (P[j] == '\0') found a match;
i = i  max(1, j-overlap(P[0..j-1],P[0..m]));
}
```

3.3.4 Skipping inner iterations

The other optimization that can be done is to skip some iterations in the inner loop. Let's look at the same example, in which we skipped from i=2 to i=4:

```
i=2: n a n
i=4:     n a n o
```

In this example, the "n" that overlaps has already been tested by the i=2 iteration. There's no need to test it again in the i=4 iteration. In general, if we have a nontrivial overlap with the last partial match, we can avoid testing a number of characters equal to the length of the overlap.

This change produces (a version of) the KMP algorithm:

3.3.5 KMP, version 1:

```
i=0;
o=0;
while (i<n)
```

```
{
  for (j=o; T[i] != '\0' && P[j] != '\0' && T[i]==P[j]; j) ;
  if (P[j] == '\0') found a match;
  o = overlap(P[0..j-1],P[0..m]);
  i = i  max(1, j-o);
}
```

The only remaining detail is how to compute the overlap function. This is a function only of j, and not of the characters in T[], so we can compute it once in a *preprocessing* stage before we get to this part of the algorithm. First let's see how fast this algorithm is.

3.3.6 KMP time analysis

We still have an outer loop and an inner loop, so it looks like the time might still be O(mn). But we can count it a different way to see that it's actually always less than that. The idea is that every time through the inner loop, we do one comparison T[i]==P[j]. We can count the total time of the algorithm by counting how many comparisons we perform.

We split the comparisons into two groups: those that return true, and those that return false. If a comparison returns true, we've determined the value of T[i]. Then in future iterations, as long as there is a nontrivial overlap involving T[i], we'll skip past that overlap and not make a comparison with that position again. So each position of T[] is only involved in one true comparison, and there can be n such comparisons total. On the other hand, there is at most one false comparison per iteration of the outer loop, so there can also only be n of those. As a result we see that this part of the KMP algorithm makes at most 2n comparisons and takes time O(n).

3.3.7 KMP and finite automata

If we look just at what happens to j during the algorithm above, it's sort of like a finite automaton. At each step j is set either to j (in the inner loop, after a match) or to the overlap o (after a mismatch). At each step the value of o is just a function of j and doesn't depend on other information like the characters in T[]. So we can draw something like an automaton, with arrows connecting values of j and labeled with matches and mismatches.

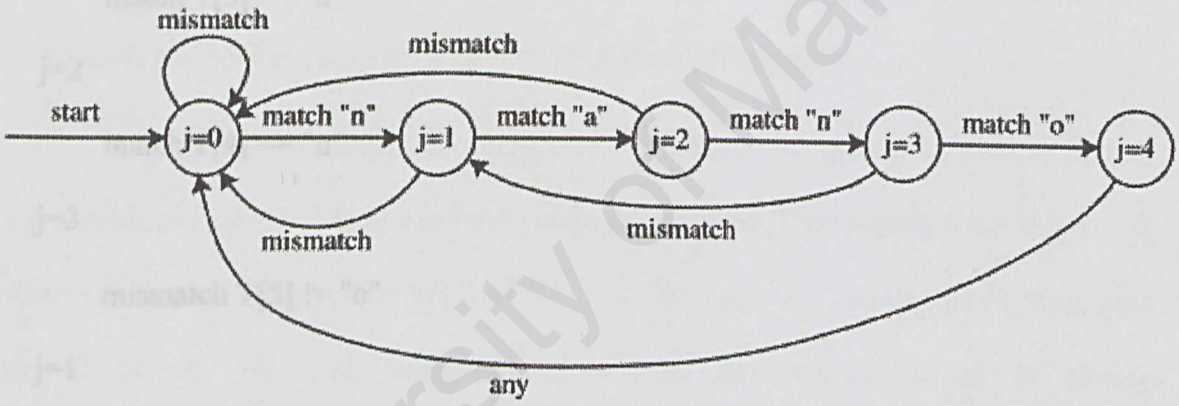


Figure 3.3 KMP and Fnite autamata technique

The difference between this and the automata we are used to is that it has only two arrows out of each circle, instead of one per character. But we can still simulate it just like any other automaton, by placing a marker on the start state (j=0) and moving it around the arrows. Whenever we get a matching character in T[] we move on to the next character of the text. But whenever we get a mismatch we look at the same character in the next step, except for the case of a mismatch in the state j=0.

So in this example (the same as the one above) the automaton goes through the sequence of states:

```
j=0
    mismatch T[0] != "n"
j=0
    mismatch T[1] != "n"
j=0
    match T[2] == "n"
j=1
    match T[3] == "a"
j=2
    match T[4] == "n"
j=3
    mismatch T[5] !="o"
j=1
    match T[5] == "a"
j=2
    match T[6] == "n"
j=3
    match T[7] == "o"
j=4
    found match
j=0
```

```
        mismatch T[8] != "n"
    j=0
        mismatch T[9] != "n"
    j=0
        match T[10] == "n"
    j=1
        mismatch T[11] != "a"
    j=0
        mismatch T[11] != "n"
```

This is essentially the same sequence of comparisons done by the above. So this automaton provides an equivalent definition of the KMP algorithm.

If we want to find all occurrences of the pattern, we should be able to find an occurrence even if it overlaps another one. So for instance if the pattern were "nana", we should find both occurrences of it in the text "nanana". So the transition from j=m should go to the next longest position that can match, which is simply j=overlap(pattern,pattern). In this case overlap("nano","nano") is empty (all suffixes of "nano" use the letter "o", and no prefix does) so we go to j=0.

3.3.8 Alternate version of KMP

The automaton above can be translated back into pseudo-code, looking a little different from the pseudo-code we saw before but performing the same comparisons.

KMP, version 2:

```
j = 0;
for (i = 0; i < n; i)
for (;;) {   // loop until break
  if (T[i] == P[j]) { // matches?
    j;       // yes, move on to next state
    if (j == m) {   // maybe that was the last state
      found a match;
      j = overlap[j];
    }
    break;
  } else if (j == 0) break;   // no match in state j=0, give up
  else j = overlap[j];    // try shorter partial match
}
```

. One advantage of this version of the code is that it tests characters one by one, rather than performing random access in the T[] array, so (as in the implementation) it can be made to work for stream-based input rather than having to read the whole text into memory first.

The overlap[j] array stores the values of overlap(pattern[0..j-1],pattern), which we still need to show how to compute.

Since this algorithm performs the same comparisons as the other version of KMP, it takes the same amount of time, O(n). One way of proving this bound directly is to note, first, that there is one true comparison (in which T[i]==P[j]) per iteration of the outer loop, since we break out of the inner loop when this happens. So there are n of these total. Each of these comparisons results in increasing j by one. Each iteration of the inner loop in which we don't break out of the loop results in executing the statement j=overlap[j], which decreases j. Since j can only decrease as many times as it's increased, the total number of times this happens is also O(n).

3.3.9 Computing the overlap function

Recall that we defined the *overlap* of two strings x and y to be the longest word that's a suffix of x and a prefix of y. The missing component of the KMP algorithm is a computation of this overlap function: we need to know overlap(P[0..j-1],P) for each value of j>0. Once we've computed these values we can store them in an array and look them up when we need them.

To compute these overlap functions, we need to know for strings x and y not just the longest word that's a suffix of x and a prefix of y, but all such words. The key fact to notice here is that if w is a suffix of x and a prefix of y, and it's not the longest such word, then it's also a suffix of overlap(x,y). (This follows simply from the fact that it's a suffix of x that is shorter than overlap(x,y) itself.) So we can list all words that are suffixes of x and prefixes of y by the following loop:

```
while (x != empty) {
x = overlap(x,y);
output x;
}
```

Overlap(x,y) by adding one more character to some word that's a suffix of shorten(x) and a prefix of y and find all such words using the loop above, and return the first one for which adding one more character produces a valid overlap:

3.3.10 Overlap computation:

```
z = overlap(shorten(x),y)
while (last char of x != y[length(z)])
{
if (z = empty) return overlap(x,y) = empty
else z = overlap(z,y)
}
return overlap(x,y) = z
```

So this gives us a recursive algorithm for computing the overlap function in general. If we apply this algorithm for x=some prefix of the pattern, and y=the pattern itself, we see that all recursive calls have similar arguments.

Replacing x by P[0..j-1] and y by P[0..m-1] in the pseudocode above and replacing recursive calls by lookups of previously computed values gives us a routine for the problem we're trying to solve, of computing these particular overlap values.. The value in overlap[0] is just a flag to make the rest of the loop simpler. The code inside the for loop is the part that computes each overlap value.

KMP overlap computation:

```
overlap[0] = -1;
for (int i = 0; pattern[i] != '\0'; i) {
overlap[i  1] = overlap[i]  1;
while (overlap[i  1] > 0 &&
       pattern[i] != pattern[overlap[i  1] - 1])
```

```
    overlap[i  1] = overlap[overlap[i  1] - 1]  1;
  }
  return overlap;
```

The outer loop executes m times. Each iteration of the inner loop decreases the value of the formula overlap[i], and this formula's value only increases by one when we move from one iteration of the outer loop to the next. Since the number of decreases is at most the number of increases, the inner loop also has at most m iterations, and the total time for the algorithm is O(m).

The entire KMP algorithm consists of this overlap computation followed by the main part of the algorithm in which we scan the text (using the overlap values to speed up the scan). The first part takes O(m) and the second part takes O(n) time, so the total time is O(m).

3.4 Requirement Analysis

Requirement analysis is an important process the acceptability of the system. After it has been delivered depends on how well it meets the specified needs and support the work to be automated. If the analyst does not discover the real requirement, the delivered system is unlikely to meet their expectation. The intelligent marking system requirement analysis has been done to understand what we expect the system to do.

3.4.1 Functional Requirements Analysis

Functional requirements are statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations. In some cases, the functional requirement may also explicitly state what the system should not do. Marking system functional requirements are :

- **Updateable Database**

  The database of the system must be able to add on the new scheme of answer by the users(teacher,lecturers and tutors).
- **Search capabilities**

  The database must be able to be searched using keywords for the answers given by the students.

3.4.2 Non-functional Requirements

Non-functional requirements are essential definition of the system properties and constrain under which a system should operate. Marking System non-functional requirements are :

- **Reliability**

  A system is considered reliable if it does not produce dangerous or costly failures when used in a reasonable manner. A system might be used in ways that the designer might not expect it to be used and the system must be able to handle these situations.

- **Accuracy**

  Accuracy refers to the precision of the information provided in the homepage It provides various accuracy measures to maintain the accuracy of the information.

- **Efficiency**

  Efficiency in computer terminology means that enables a system to handle or avoid disaster in the face of unexpected data.

- **Robustness**

  Robustness refers to the quality that enables a system to handle or avoid disaster in the face of unexpected data.

- **Correctness**

  Correctness refers to the extent to which a program satisfies its specifications and fulfils the user's requirements.

- **User friendliness**

  User friendliness is achieved in the system by providing hyperlinks, which the user can select by using the mouse. All selection is made in such a way that the user easily sees them.

- **Modularity**

Modularity is a key factor in good program design. The working of the system was broke into modules so that the distinct functions of objects could be isolated from one another. This characteristic makes it easier to perform testing and maintenance.

- **Maintainability**

  Maintainability can be defined as the ease with which software can be understood, corrected, adapted, and/or enhanced in the future.

- **Expandability**

  Expandability measures the degree to which the architecture, data or procedural design can be extended. This system is designed to be expandable in the future.

- **Response Time**

  Web site downloading should be within an acceptable time where retrieving data should be fast enough to make the accessing easy for the web surfer.

3.5 System Requirements of the Proposed Tools

The selection of both hardware and software are vital to ensure the success of a system. The task of choosing the most suitable hardware and software must be done wisely to meet the system requirements.

In selecting hardware devices, a few criteria have been established to ensure the hardware-selected meet the requirements of the system to be developed. Criteria that have been emphasized are capability, credibility, cost speed and size of memory.

- **Hardware Requirements**
    - Intel Pentium II 233MHz or compatible (recommended minimum)
    - 128 MB (recommended minimum)
    - 256 MB (recommended)
    - 115-250 MB hard disk space (depending on features installed)
    - Monitor capable of 16-bit display with 800x600 resolution or more

- **Software Requirements**

Choosing the right software for a particular application requires knowledge of strength and weakness of the specific software products. The following are the software used for the development of the Marking System.

- Jbuilder 5
- Microsoft Windows 98, 2000, or NT 4.0 (Service Pack 3 or higher)

Summary

The waterfall model with the prototyping was been chosen as the system development strategy due to its suitability mentioned earlier.The AI method which was the Expert System using Rule-Based system was introduced in this chapter. Information retrieval and keyword matching that are used for developing the system was also included here.System procedure with functional and non-functional requirements was also included in this chapter. The system design will come next as in the following chapter after the planning of this methodology.

CHAPTER 4

SYSTEM ANALYSIS AND DESIGN

CHAPTER 4 :SYSTEM ANALYSIS AND DESIGN

4.0 System Analysis

System analysis involves analyzing the system needs.Special tools and techniques are used to help the analyst make requirement determinations.One such tool is the use of Data Flow Diagram to chart input ,processes and the output.From the data flow diagram, a data dictionary is developed that lists all of the data items used in the system, as well as their specifications.

During the phase of developing the system,the structured decision made is also analyze.Structured decisions are those for which the condition,condition alternatives,action and action rules can be determined.There are three major methods for analysis of structured decision: structured English,decision tables and decision trees.

The Intelligent Marking System will have the system security .This is to make sure that only the user who have the access can enter the system.This is to make sure that no fraud will be made to this system by the irresponsible people.

System design is very essential in a system development process as it plays a major role in determining the success of the system. The system specifications describe the features of a system and their appearance to users. The requirements that are concluded from the analysis phase are translated into design specifications.

Intelligent Marking System is design to meet the following goals:

- Suitability - the system must fulfil the objective of the project.
- Reliability - the user can rely on the system to work correctly and produce

accurate output.

- Simplicity - the system must be simple enough for all levels of users.
- . User Interface - the system should support user-friendly interface

4.1 Logical Design

The first process in system design process is to describe the logical design of the project being developed. In Automarking System, a user or web surfer will access the internal web server and be able to surf through the web site with ease and works in carousel.

System design is the foundation of software development. Without proper design, even the greatest idea would fail. Care has to be given when designing the system to ensure that the design is robust, expandable and most important of all, free from bugs.

In the design of the Intelligent Marking system care has been given toensure that the design is as modular as possible to make sure debugging and maintenance can be done easily. Each module is design to handle one particular area of the system, so that any future changes to the system will not affect the rest of the design.

Intelligent Marking System is divided into the following categories :

- .introduction page
- .registration and putting data in database
- .question page done static

- .putting answer in database/view in the same page
- .calculation page/marking/together with answer and question
- .skema page for administration

Users would be able to access by algorithm from page to page. The user may navigate in ease. Alert sign will be prompt when command did not done properly.

4.2 Graphical User Interface (GUI)

A GUI is a graphical rather than a purely textual interface for communication between the user and a computer. The term GUI came into existence because the first interactive user interfaces to computer was text-and-keyboard oriented and did not include graphics. Commands had to be typed in brief and responses were received from the computer. The command interface of DOS operating system is an example of the typical user-computer interface before GUI was introduced.

Most of today's major operating systems provide a graphical user interface. Applications typically use the elements of the GUI that come with the operating system and add their own GUI elements and ideas into them. GUI elements include windows, pull-down menus, button, scroll bars, iconic images, and wizards. With the increasing use of multimedia as part of the graphical user interface, other components such as audio, motion video, and virtual reality interface as likely to be integrated into the GUI for many applications.

The Java programming language class library provides a user interface toolkit called the Abstract Windowing Toolkit, or the AWT. The AWT is both powerful and flexible. Newcomers, however, often find that its power is veiled. The class and method descriptions found in the distributed documentation provide little guidance for the new programmer. Furthermore, the available examples often leave many important questions unanswered. Of course, newcomers should expect some difficulty. Effective graphical user interfaces are inherently challenging to design and implement, and the sometimes complicated interactions between classes in the AWT only make this task more complex. However, with proper guidance, the creation of a graphical user interface using the AWT is not only possible, but relatively straightforward.

4.3 User Interface (Physical Design)

The interface is the system for most users. However well or poorly designed, it stands as the representation of the system. The user interfaces of Intelligent Marking System are evaluated to ensure that the system meets the desire objectives. Intelligent Marking user interface objectives are :

- Effective user interface to allow the user to access the system in easy way.
- Efficient user interface to allow the user to access the internal web server with minimum time required and no error.
- The interface should use terms and concepts that are familiar to the

class of users.

- The interface should be appropriately consistent.
- The user should not surprise by the system.

4.4 Database Designs

Database is one of the approaches to the storage of data in a computer-based system. A database is a formally defined and centrally controlled store of data intended for use in many different applications. Databases are not merely a collection of files. Instead, a database is a central source of data meant to be shared by many users for a variety of applications. The heart of a database is the DBMS (database management system), which allows the creation, modification, and updating of the database; the retrieval of data; and the generation of reports.

The database of the Intelligent Marking System was done in MS Access and connected to the web server through ODBC. ODBC is an acronym for Open Database Connectivity. This term describes an interface for accessing data stored in a variety external database. ODBC allows web server to communicate with the MS Access database.

Before the web server can communicate with the database, the database must be configured with the appropriate driver as a data source. A unique data source must be created for each database. The data source can be set through the windows Control Panel.

In java technology, ODBC must communicate with JDBC (java database connectivity) in order to make the connection between database. JDBC technology is an API that lets you access virtually any tabular datasource from the JavaTM programming language. It provides cross-DBMS connectivity to a wide range of SQL databases, and now, with the new JDBC API, it also provides access to other tabular data sources, such as spreadsheets or flat files.

The JDBC API allows developers to take advantage of the Java platform's "Write Once, Run Anywhere" capabilities for industrial strength,cross-platform applications that require access to enterprise data. With a JDBC technology-enabled driver, a developer can easily connect all corporate data even in a heterogeneous environment.

4.5 Data Dictionary

A data dictionary is a specialized application of the kinds of dictionaries used as references in everyday life. The data dictionary is a reference work of data about data (i.e. metadata) compiled by system analyst to guide them through analysis and design. It explicitly represents the relation ships among data objects and the constraints on the elements of data structure. The data requirements had to be represented in the database itself in the form of database tables. Table 4.1 below shows the data dictionary for Intelligent Marking System

yang	**0.1**
merentas	**0.1**
sinaps	**0.5**

Every keywords in the answer will be given marks.

To develop the Intelligent Marking System, the Data Flow Diagram and the decision trees have been chosen .Shown below is the Decision Tree for the Intelligent Marking System:

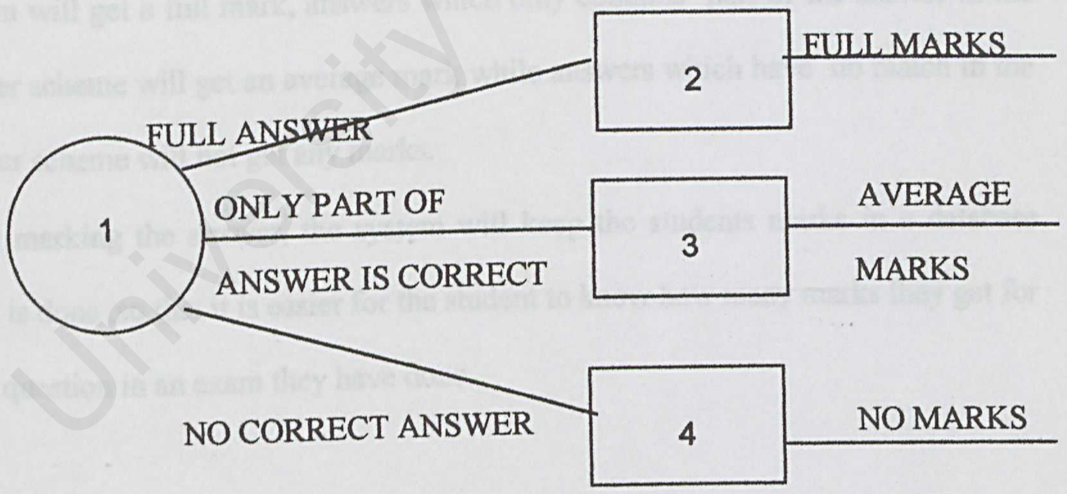


Figure 4.1 The Decision Tree of the Marking System

4.6 HOW THE INTELLIGENT MARKING SYSTEM WORKS?

To get a clear picture of the operation of the system, the following are the summary of processes of the Intelligent Marking System.

- The student will submit their answer and the answer will be stored in the database before the marking is done by the lecturer.
- B y using the rule based system,information retrieval and keyword matching approach,the system will start marking the student's answers one by one and marks will be given to every correct answer based on the answer scheme.
- The application will first read all the answer and then searched for the related keyword that have the same keyword,as the one stored in the marking system
- Answers that have exactly the same keyword as the answer scheme in the marking system will get a full mark, answers which only contains part of the answer in the answer scheme will get an average mark while answers which have no match in the answer scheme will not get any marks.
- After marking the answer, the system will keep the students marks in a database .This is done ,so that it is easier for the student to know how many marks they get for each question in an exam they have done.

4.7 Modules of The Intelligent Marking System for Structured Question

The Marking System will consist of 2 main modules – display module and filtering module. Each module deals with a specific area and functions of the games. Any change that need to be implemented will only involved one or two modules at the most, without the need to modify the entire system. Addition to the current project will also be easy, since only new module needs to be written and added to the system.

The functions of each module is as following:

4.8 Structure of the Design

The figure below gives the structure of the design in term of Data Flow Diagram.

4.8.1 THE STUDENT DATA FLOW DIAGRAM FOR THE INTELLIGENT MARKING SYSTEM

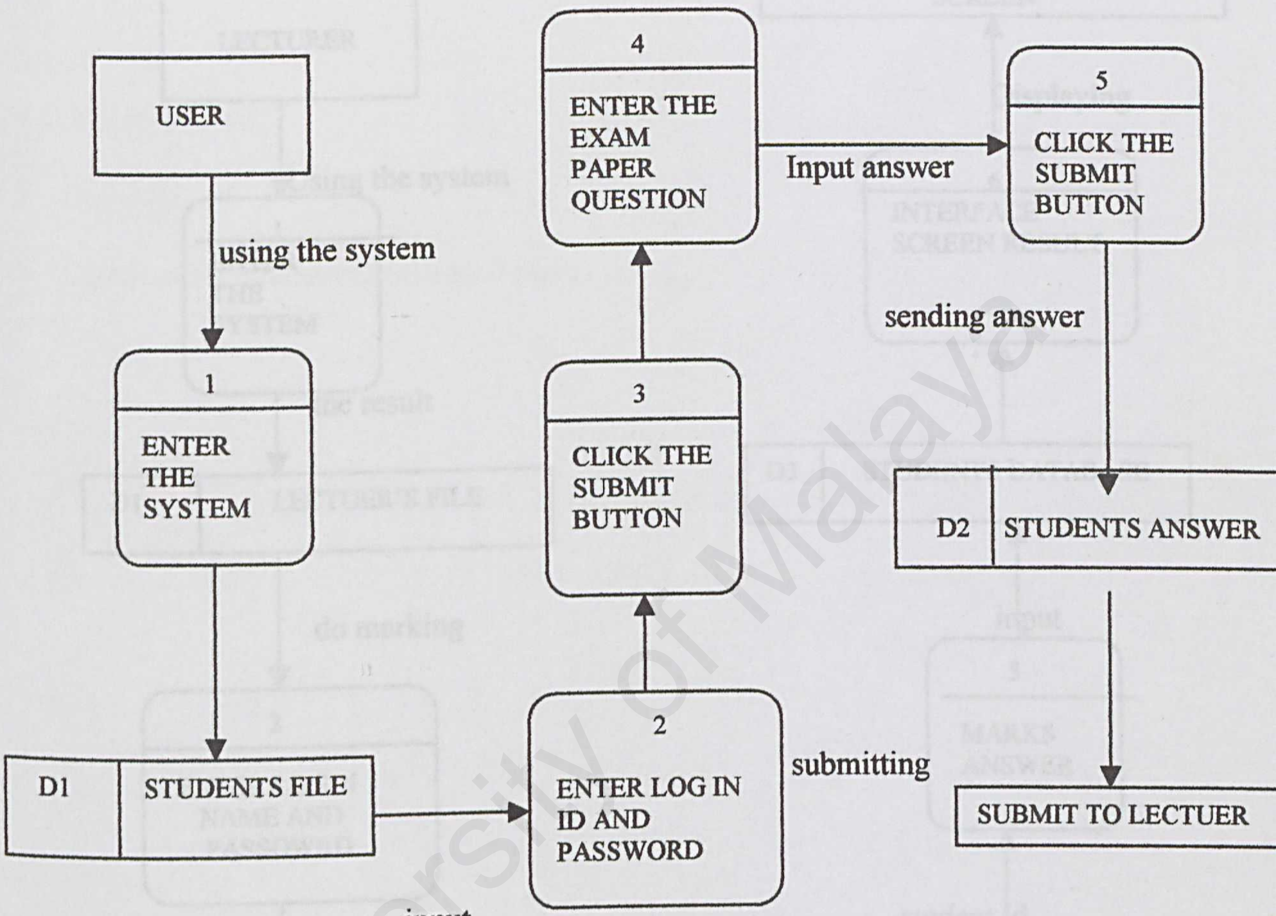


Figure 4.2 Student Data Flow Diagram for The Intelligent Marking System

4.8.2 THE LECTURER DATA FLOW DIAGRAM FOR THE INTELLIGENT MARKING SYSTEM

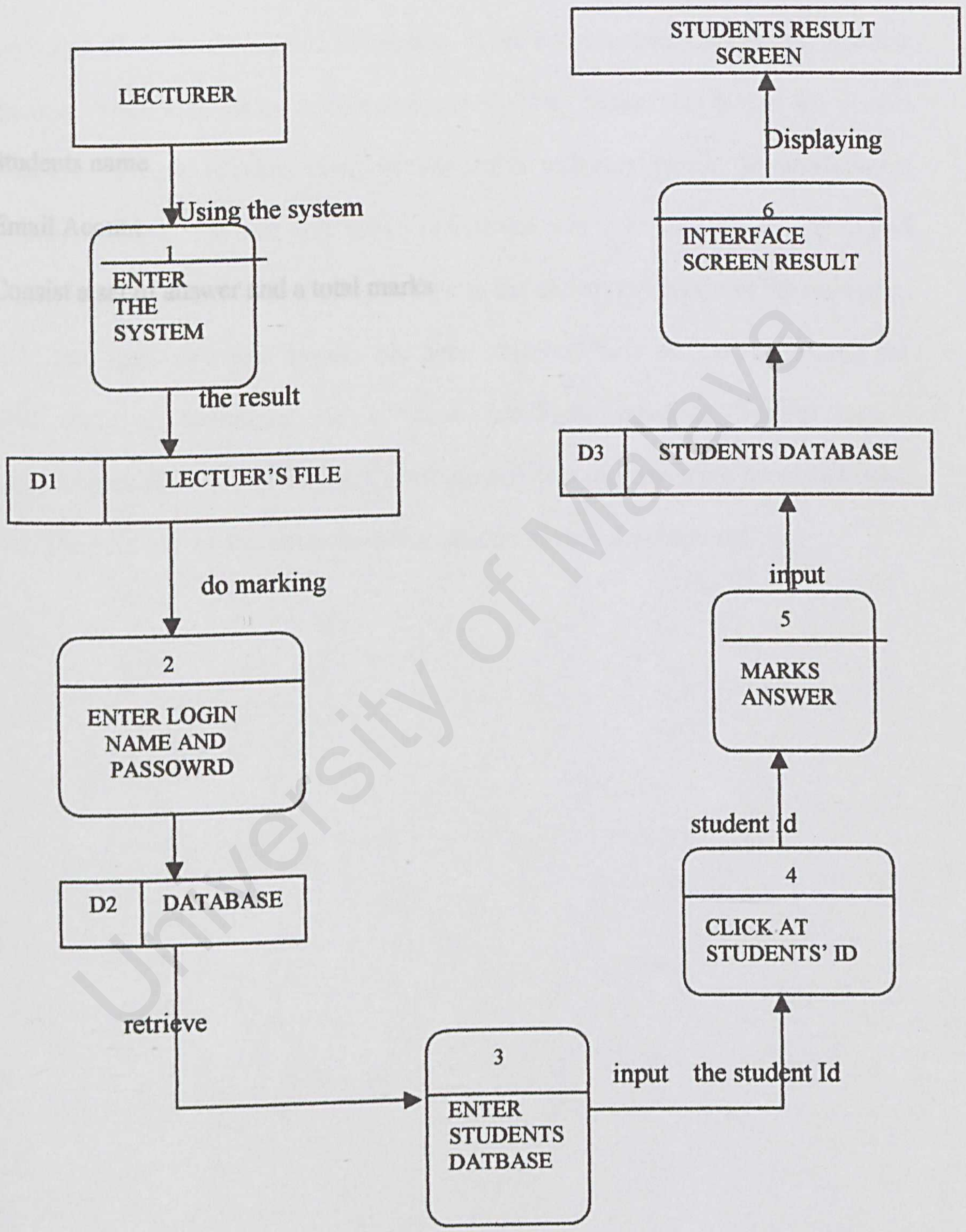


Figure 4.3 Lecturer Data Flow Diagram for Intelligent Marking System

4.9 The element that will be displayed on the screen:

- Students name
- Email Acount
- Consist a set of answer and a total marks

Summary

System analysis and design is an important step when analyzing and designing a system.It is the basis for a solid foundation of the entire system.Choosing a modular design helps when changes or additions occur.Modular deisgn also makes the system more robust and easier to debug.User interface is also important part in the development of the software.Sonce the user only interacts with the user interface,this come to regard that a flaw in the interface design is also a flaw in the underlying deishn of the software.

Intelligent Marking System has been designed base on the logical and the physical designed technique. As a result Intelligent Marking System logical diagram,samples of the used technique, database and data dictionary are presented inthis chapter. The next will be the implementation and the system development.

CHAPTER 5

SYSTEM IMPLEMENTATION

Chapter 5 : System Development and Implementation

5.0 System Development

Usually in prototype model, the requirement analysis, system design and implementation phases do not have a clear boundary. Each phase tends to overlap one another. System implementation is a process that converts the system requirements and design into programming codes. This phase at some times involves some modifications to the previous design.

5.1 Development Strategy

The project development strategy employed in this project is rapid prototyping. Rapid prototyping is used because:

- It helps create a working model of the system
- The trial model is used to decide on the final design

there are six steps in developing a prototype that are shown in the diagram below.

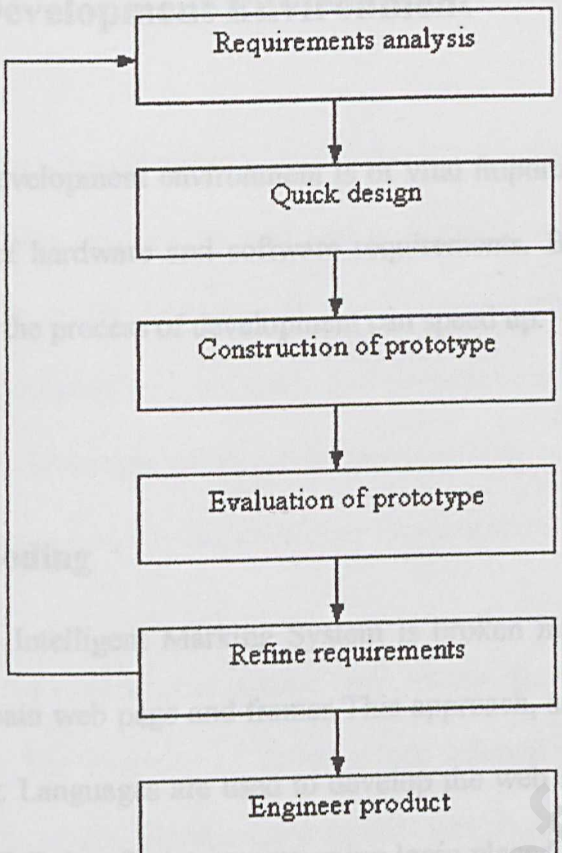


Figure 5.1 Six steps in developing a prototype

The type of prototype model used is evolutionary prototyping. Evolutionary prototyping was used because:

- There are potential for changing the system early in development
- Opportunity to pause development on system that is not workable
- Time is not wasted building prototype a in throw-away prototype

5.2 Development Environment

Development environment is of vital importance to any system development. It consists of hardware and software requirements. By using the suitable hardware and software, the process of development can speed up.

5.3 Coding

Intelligent Marking System is broken into different tasks, which are input, output, main web page and frames.This approach, simplify the problems when it comes to coding. Languages are used to develop the web site are Java, JSP, Servlet, and Java Bean, HTML.JavaScript programming logic placed into HTML lines. This means when the web site downloaded, the script codes is downloaded as well. Frames are inserted and carefully aligned to suit the outlook. To make the web page more attractive, pictures and colorful images are designed and drowned out.Preparation of a HTML, Java, JSP(Java Server Pages), Servlet, Java Bean, cycle of testing and modifying the source codes, loading the file in the browser for viewing and validating and then going back to make further changes when necessary. Refer to the Appendix A for the source codes of the Intelligent Marking System.

5.3.1 Structure Programming

Structure programming is a discipline approach to programming that results in programs that are easy to read and understand and less likely to contain errors. The emphasis is on the following accepted program style guidelines to write code that is clear and readable. Obscure tricks and programming short cuts are strongly discouraged. The main advantage of structure programming is that it is easier to design in the beginning and easier to maintain over the long term.

An easy to read source code makes the system easier to be maintained and enhanced. The elements of style include internal (source code level) documentation, methods for data declaration and approach to statement construction .The following are some of the used coding methods :

1. Selection of meaningful identifier (variable and labels) names.
2. Appropriate comments written in the source codes.
3. Indentation of codes increases the readability.
4. (OOP) Object Oriented Programming is the most important in design with standard.

Summary

Intelligent Marking System has been developed through structure programming to makes the system easier to be maintained and enhanced. System testing will be discussed in the next chapter.

CHAPTER 6

SYSTEM TESTING

Chapter 6 : System Testing

6.0 System Testing

Testing is done to ensure that the system works properly . The system testing has been done to some people with teaching background,the lecturer in Kolej Cermai Jaya,Kuching.

6.1 Testing Objectives

The objective of system testing is as follows :

- To identify, isolate and correct as many bugs as possible. Most programs have bugs, the most insidious of which appear only with unique combinations of data or events.
- To demonstrate that functionality of the system appears to be working properly and performance requirements appear to be met.

6.2 Test Case Design

Any system can be tested using one of two types of test case design. They are white-box testing and black-box testing.

6.2.1 White-Box Testing

White-box testing is carried out at the early stages of the testing process.It is performed to ensure that the internal operation of a system performs according to specifications and all internal components have been adequately exercised. Using these methods, a system engineer can derive cases that :

1. Guarantee that all independent paths within a model have been exercised at least once.
2. Exercise all logical decisions on their true and false sides.
3. Execute all loops at their boundaries and within their operational bounds.
4. Exercise internal data structures to assure their validity.

6.2.2 Black-Box Testing

Black-box testing is conducted to demonstrate that each is fully operational, at the same time searching for errors in each function. This testing approach enables a system engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. Black-box testing is not an alternative to white-box testing. Rather it is a complementary approach that is likely to uncover a different class or errors than those uncovered by white-box methods.

Black-box testing attempts to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structures or external database access
- Performance errors
- Initialization and termination errors

6.3 Testing Strategies

Testing strategies used for testing are unit testing, integration testing and system testing. The objective of unit testing and integration testing is to ensure that the code implemented to design properly; that is the programming part can fulfil what the designing part intended. In system testing, the objective was to ensure that the system does what the user wants it to do.

6.3.1 Unit Testing

In the unit testing, each component is test individually. Each of project features had to be tested individually first. The features that were tested individually include :

1. JavaScript - tested on the browser, Netscape Communicator 4.0 to prove its correctness. After testing over browser, features combined with other feature of the web.

2. Links - tested to ensure that they navigate to the correct screen when clicked upon.

6.3.2 Integrating Testing

This type of testing is the process of reifying the system component work together as described in the system and program design specification. The following approaches are used in integration testing of Intelligent Marking System::

1. JavaScript, HTML, frames are tested on the browser together. Internet Explorer 5 shown no problem when all the components were integrated and tested.

2. All links are tested to ensure that it is not broken or missing. When each links is clicked, connection is established to the desired target and web site to ensure that the user can access the web site.

6.4 System Testing

System testing is to ensure that the system is performing well as a completed system. There are several steps in testing a system :

1. Functional Testing

Functional test checks that the integrated system performs its functions as specified in the requirements.

2. Performance Testing

Performance testing compares the integrated components with the nonfunctional system requirements as it had been described in chapter 3. That it constrains the way in which the system functions are performed.

3. Acceptance Testing

Acceptance test assures the users that the system they requested is the system that was built for them.

4. Installation Testing

A final installation test is run to allow users to exercise system functions and documents additional problems that result from being at the actual site.

Summary

This chapter presents the way of implementing and testing of the Intelligent Marking System using the white and black box testing. The next chapter will be the system evaluation and conclusion.

CHAPTER 7

PROBLEMS AND SOLUTIONS

Chapter 7: System Evaluation and Conclusion

7.0 Problems Definition and Solutions

Through out the system development, problems are encountered and most of them were resolved eventually. The system was evaluated through system testing to identify its strength and limitations and proposals were made for the future enhancement.

7.1 Project Problems and Solution

Many problems were encountered during the development of the system. Efforts were put to solve the problems. Advise from the supervisor and my friends were definitely a big help on problem solving. Besides, surfing the Internet, joining the newsgroup, reading reference books or even interviewing expert to get to the most practical and reasonable solution.

7.1.1 Problems

During this project, there were several problems encountered such as :

1. Difficulties in defining the system scope at the beginning phase.
2. Difficulties in finding the good software to implement the system.From using the Visual Basic,I choosed Java-Jbuilder 5 to implement my system.
3. Lack of programming knowledge in Java-Jbuilder5 and the Artificial tehnique,Knuft-Morris-Algotrithm and Naïve Algorithm
4. Learning the tools in Knuft-Morris,Naïve Algorithm and Jbuilder 5 takes time.
5. Difficulties in linking the database to the system.
6. Difficulties in understanding the real needs of the users.
7. Problems with the faculty equipment such us slow down process of the server and insufficient of software required.

7.1.2 Solutions

The solution to the problems that are stated above is to analyze several marking system web site and senior project to highlight the covered and uncovered features, the system scope and criteria to be used in implementing good user interface.Lack of knowledge in the programming was done by referring to some of the books and also the online tutorials.Websites such as PlanetSourceCode.com provides tips and help in the forms of code snippets and tutorials from users all over the world.

Due to the problems with the faculty equipment's, several alternatives have been taken to still proceed with the system implementation.

Visual Basic that has been proposed earlier is replaced by JBuilder 5. This is because after doing some research,I found that Jbuilder 5 is

- it is an independent source,it can be run anywhwere e.g windows,linux
- Less likely to crash
- The memory cannot be intentinally misused
- It is easier to user –user firendly
- Eventhough it doesn't have its own databse but Jbuilder can use seperatable dfatabase such as MS Access
- Can store many data in the databse

7.2 Features and Strength

The following illustrates the key strength of The Intelligent Marking System

- **User-friendly**

  Intelligent Marking System has a user-friendly and consistent interface for the ease of users. A set of standard GUI has been implemented and has been implemented and has been set to allow the users to browse in a very short time.

- **Cross-browser Compatibility**

  Intelligent Marking System can run on Internet Explorer and Netscape Navigator or even any other serve rthat are compatibible with it.

- **Fast Response**

  Each web page is design to be lightweight. These pages are loaded in a reasonable amount of time to ensure a fast view of the page where heavy graphics been avoided.

- **Consistency**

  Intelligent Marking System navigation maintaining its consistencies.

- **Web Content**

  Intelligent Marking System can covers a wide area of various type of subjects but for this system I covered only the Biology subject.

7.3 System Limitation

This web site has some limitation due to the factors listed below :

1. Due to the time constrains, most of the proposed tools have to be eliminated but still suits the user requirements.

7.4 Future Enhancement

This web site can be further enhanced by adding more functions to it to make the site more informal and useful to the user. Some functions such as step-by-step directions of the recipes were not added into the system because of the time constrain. So, more suggestions are given on how this system can be improved in the future.

7.5 Suggestions

A few suggestions would like to be made on how this project could proceed more smoothly. These improvements would help make the project easier to be completed.

1. Addition of the programming subject

The faculty should add more programming subjects into the syllabus, especially the current popular ones on the market. This would better prepare the students to develop their projects, as they would have more knowledge on the language that would be most suitable to develop their application. The introduction of these programming languages as a subject would also help the students learn he new languages in a systematic way without any rush.

2. Legal software

The faculty should also provide the service of borrowing legal software to the students to enable the software to be installed at home. This would help the students develop their application at home, without depending on the laboratories, which are open only at specific times. This step should also help the students from buying pirated software.

3. Experienced lecturers

Some experienced lecturers should be identified for each of the programming language chosen by the students. This would help the students to seek help in the developing process of the project. The students would be able to refer to those lecturers if they face any problems in using the chosen programming language.

7.6 Knowledge and Experience Gained

A lot of knowledge and experience has been gained throughout the development process of this project. Among them are :

- Experience in studying a new programming languages outside the formal classroom.
- Enhanced creativity and initiative in developing the most appropriate system for the end-user.
- Sharpened problem-solving skill by finding solution to all the problems faced during the development of the project.
- Practicing proper time management to divide the time appropriately among all the subjects taken, and not neglecting any of them.
- Improved confidence in own ability
- Willingness to listen to other opinions and correct oneself.

Summary

The development of this system is not just to fulfil the requirements for a degree but also as an experience for future use when the student is faced with the task of developing a bigger system. This exercise is hoped to inspire confidence in the student or their capability.

A lot of time was spent in choosing the appropriate tools and the suitable interface to make the system as simple and understandable as possible to the end-user. Their needs and interests were taken into considerations so that this system would be beneficial to them.

The Intelligent Marking system was successfully developed within the time frame given. Most of the functions named in the analysis and design phase were incorporated into the system. It is hoped that this system will fulfil the needs of the end-user and help them in marking process.

Appendix A

SOALAN BIOLOGY

1. Apakah Homeotasis? (1m)

   Keupayaan organisma hidup mengekalkan persekitaran dalaman yang stabil.

2. Nyatakan dua fungsi ginjal (2m)

   a.Perkumuhan b.mengawal tekanan osmosis badan

3. Apakah sistem endokrina? (1m)

   Kelenjar tanpa duktus yang merembes hormon secara lansung dalam aliran darah.

3. Terdapat 6 kelenjar endorkrina pada manusia.Nyatakan kelenjar –kelenjar tersebut.

   a.kalenjar pituari

   b.kalenjar tiroid

   c.kalenjar adrenal

   d.pankreas

   e.ovari

   f.testis (3m)

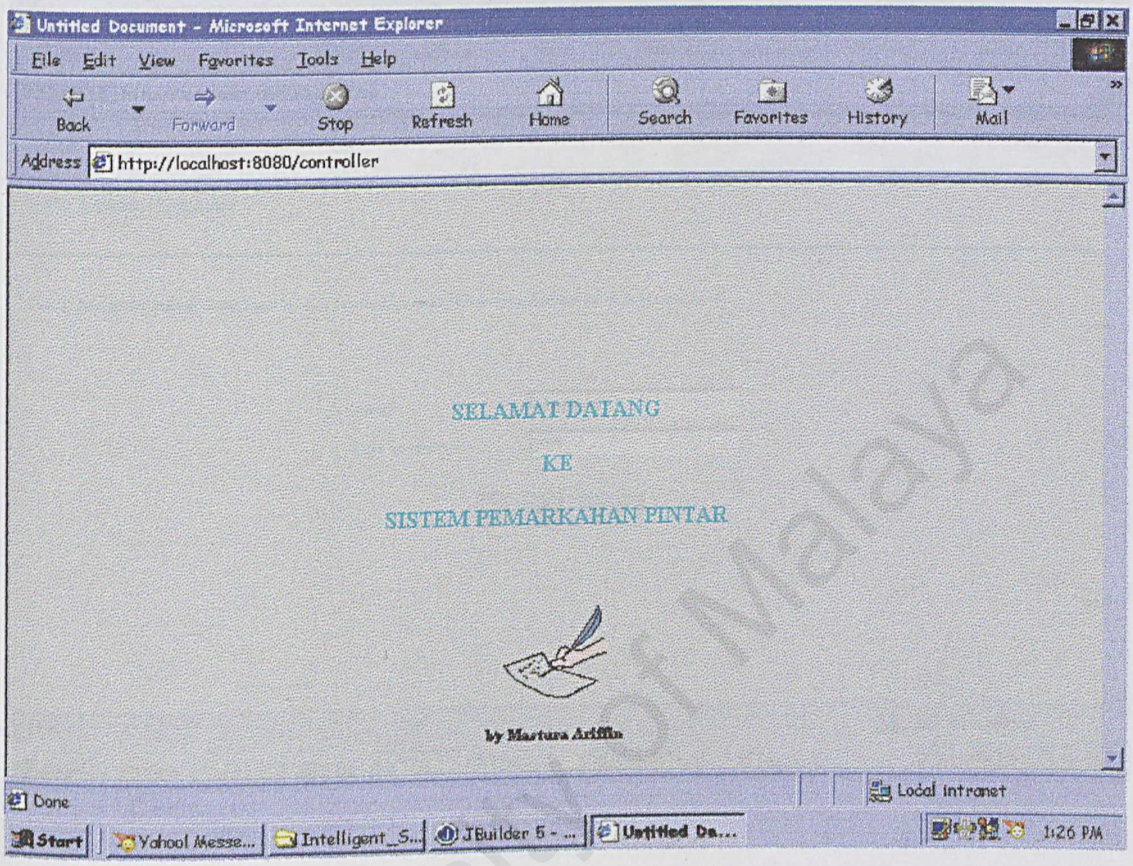
5. Bagaimankah kalenjar endokrina berbeza dari kelenjar air liur dan peluh?(2m)

   Kalenjar endorkrina tidak mempunyai duktus,manakala kalenjar air liur dan peluh berduktus.

User Manual Intelligent Marking System

User Login And Password

Main Page Intelligent Marking System


Untitled Document - Microsoft Internet Explorer

Address http://localhost:8080/controller

SELAMAT DATANG

KE

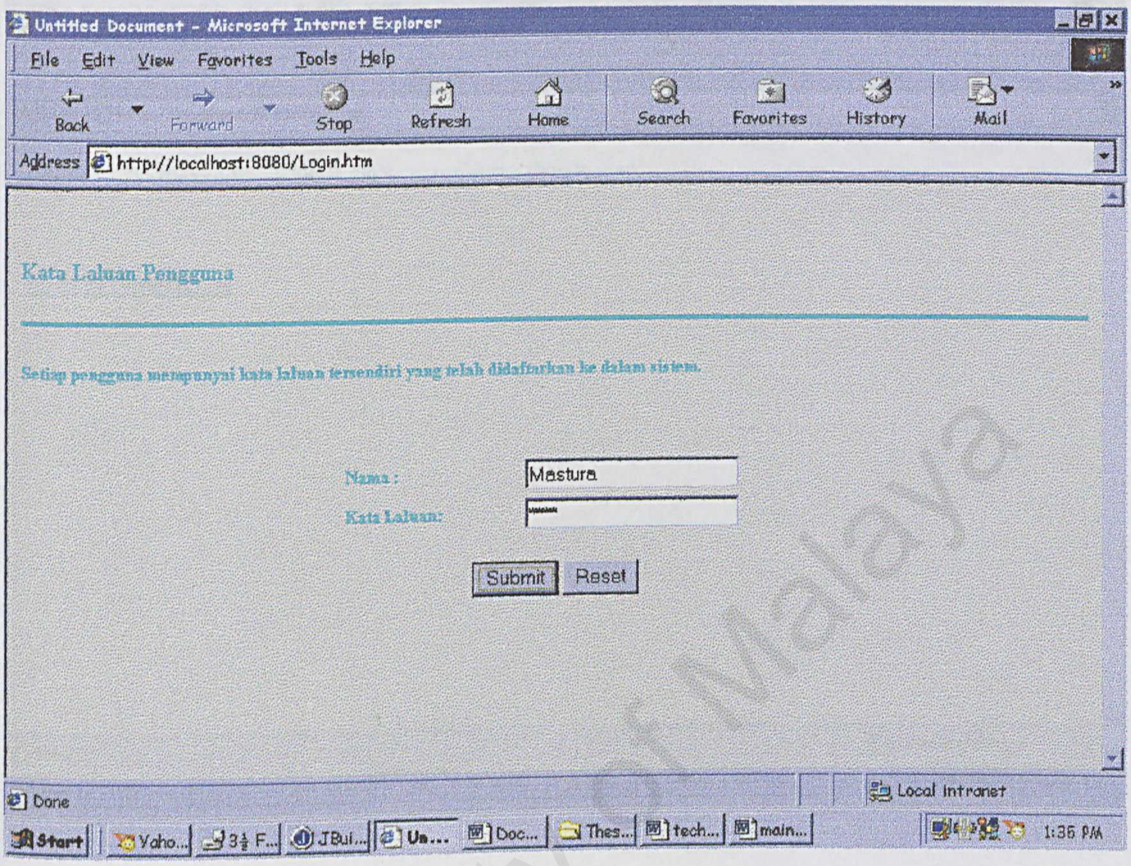
SISTEM PEMARKAHAN PINTAR

by Mastura Ariffin

File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail
Done
Local intranet
Start
Yahoo! Messe...
Intelligent_S...
JBuilder 5 - ...
Untitled Do...
1:26 PM

The Question Paper interface

User Login And Password


Untitled Document - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail
Address

http://localhost:8080/Login.htm

Kata Laluan Pengguna

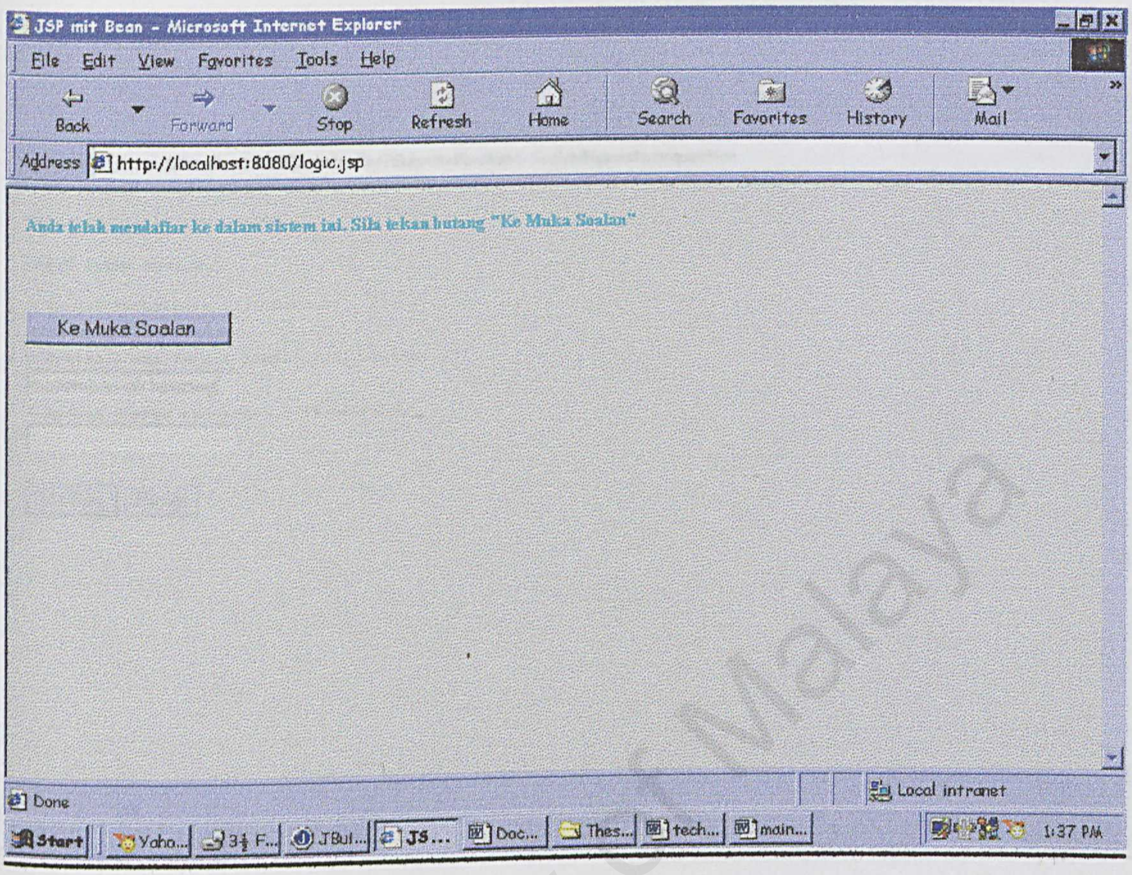
Setiap pengguna mempunyai kata laluan tersendiri yang telah didaftarkan ke dalam sistem.

Nama : Mastura

Kata Laluan: ******

Submit
Reset
Done
Local intranet
Start Yaho... 3½ F... JBui... Un... Doc... Thes... tech... main...
1:35 PM

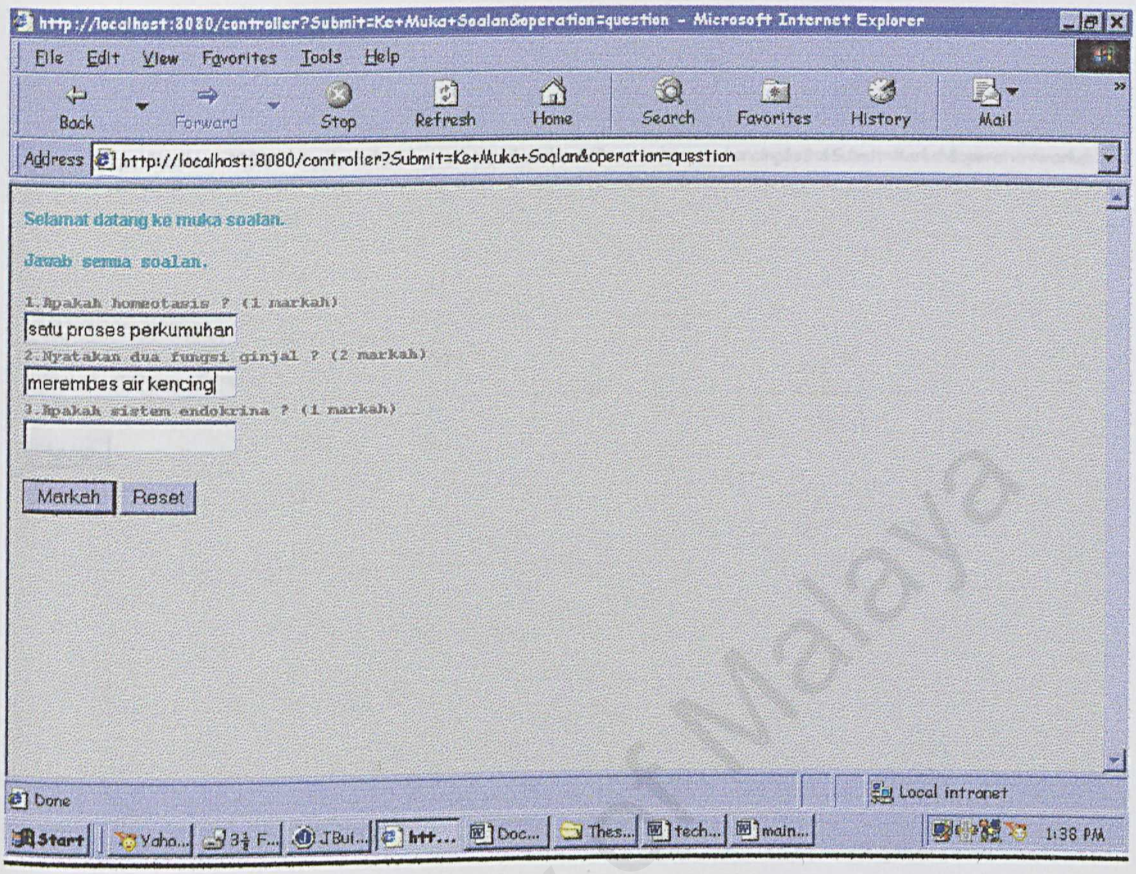
The Question Paper interface


JSP mit Bean - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail
Address http://localhost:8080/logic.jsp

Anda telah mendaftar ke dalam sistem ini. Sila tekan butang "Ke Muka Soalan"

Ke Muka Soalan
Done
Local intranet
Start
1:37 PM

The Question


http://localhost:8080/controller?Submit=Ke+Muka+Soalan&operation=question - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail
Address http://localhost:8080/controller?Submit=Ke+Muka+Soalan&operation=question
Done
Local intranet
Start Yaho... 3½ F... JBui... htt... Doc... Thes... tech... main...
1:38 PM

Selamat datang ke muka soalan.

Jawab semua soalan.

1.Apakah homeotasis ? (1 markah)

satu proses perkumuhan

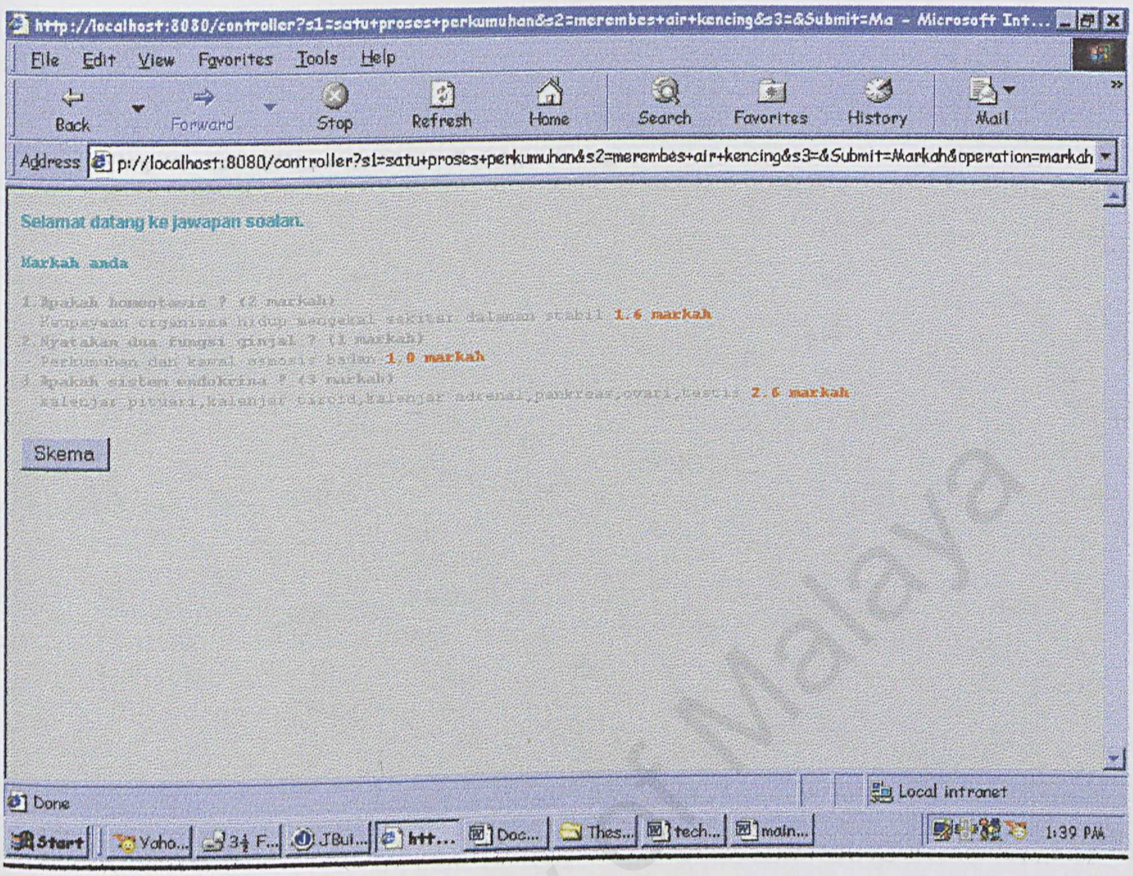
2.Nyatakan dua fungsi ginjal ? (2 markah)

merembes air kencing

3.Apakah sistem endokrina ? (1 markah)

Markah Reset

The Marks given to the Correct Answer



http://localhost:8080/controller?s1=satu+proses+perkumuhan&s2=merembes+air+kencing&s3=&Submit=Ma - Microsoft Int...

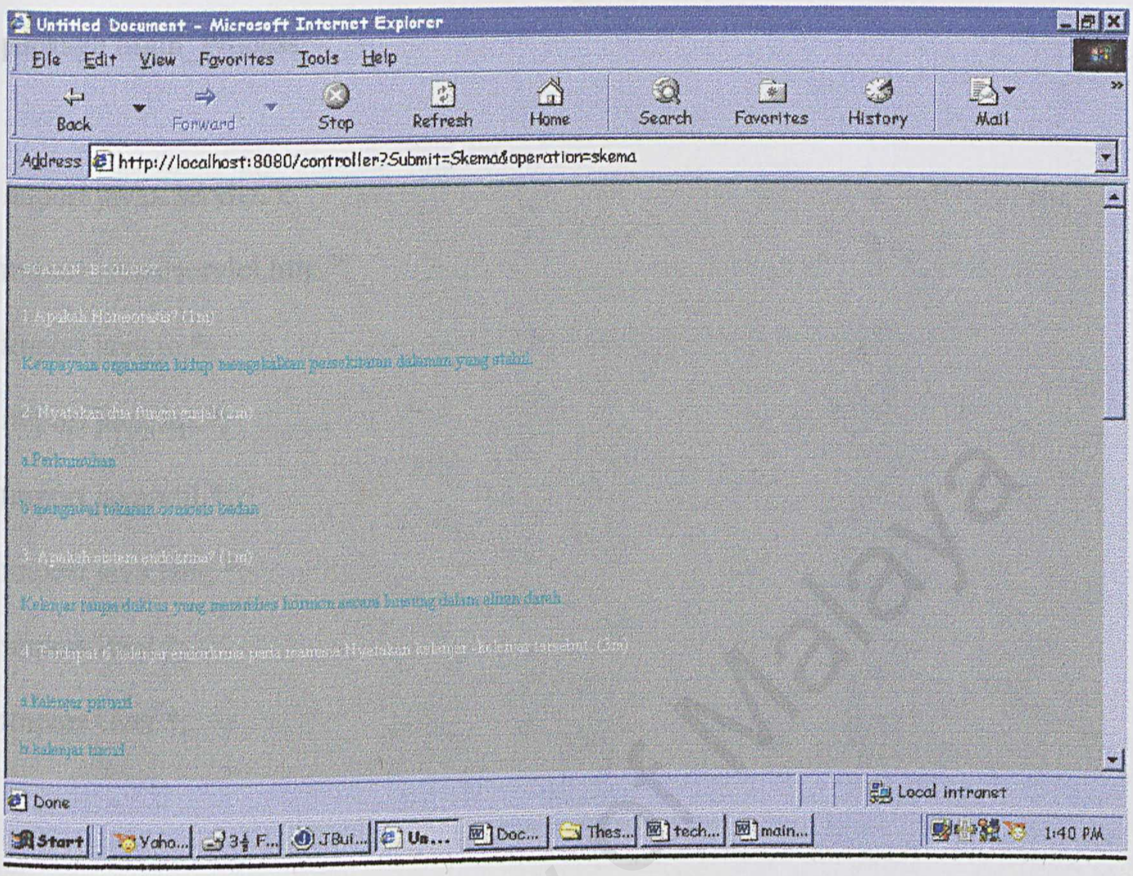
Address: p://localhost:8080/controller?s1=satu+proses+perkumuhan&s2=merembes+air+kencing&s3=&Submit=Markah&operation=markah

Selamat datang ke jawapan soalan.

Markah anda

```
1.Apakah homeostasis ? (2 markah)
 Keupayaan organisma hidup mengekal sekitar dalaman stabil 1.6 markah
2.Nyatakan dua fungsi ginjal ? (1 markah)
 - Perkumuhan dan kawal osmosis badan 1.0 markah
3.Apakah sistem endokrina ? (3 markah)
 kelenjar pituari,kelenjar tiroid,kelenjar adrenal,pankreas,ovari,testis 2.6 markah
```

The Answer Scheme for the Question


Untitled Document - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail
Address http://localhost:8080/controller?Submit=Skema&operation=skema
Done
Local intranet
Start Yaho... 3½ F... JBui... Un... Doc... Thes... tech... main...
1:40 PM

SOALAN BIOLOGY

1.Apakah Homeostasis? (1m)

Keupayaan organisma hidup mengekalkan persekitaran dalaman yang stabil.

2. Nyatakan dua fungsi ginjal (2m)

a.Perkumuhan

b.mengawal tekanan osmosis badan

3. Apakah sistem endokrina? (1m)

Kelenjar tanpa duktus yang merembes hormon secara langsung dalam aliran darah

4. Terdapat 6 kelenjar endokrina pada manusia. Nyatakan kelenjar-kelenjar tersebut. (3m)

a.kelenjar pituitari

b.kelenjar tiroid

<u>V</u>

Source Code 1

```
package Main_Process;

import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.sql.*;
import java.util.*;
import java.lang.*;
import Pool.*;
import User.*;

public class Controller extends HttpServlet {

 public JDCConnectionPool jPool ;
 Connection con ;
 UserUtilities util ;

 private static final String CONTENT_TYPE = "text/html";
 /**Initialize global variables*/
```

```
public void init(ServletConfig config) throws ServletException {

try
  {
  jPool = new
JDCConnectionPool("sun.jdbc.odbc.JdbcOdbcDriver","jdbc:odbc:Mastura","","",
"C:\\ConnectionLogPool.log", 1, 4, 30000);
  }
  catch(Exception e) {
    e.printStackTrace();
  }
  util = new UserUtilities();
  super.init(config);
}
/**Process the HTTP Get request*/
public void doGet(HttpServletRequest request,HttpServletResponse response)
  throws ServletException, IOException
  {
    response.setContentType(CONTENT_TYPE);
    String operation = request.getParameter("operation");
  if (operation == null)
  {
   gotoPage("/index.htm",request,response);
  }
  if (operation.equals("login"))
  {
   gotoPage("/Login.htm",request,response);
  }
  if(operation.equals("subscribe"))
  {
   gotoPage("/Subscribe.htm",request, response);
  }
  if(operation.equals("question"))
  {
   gotoPage("/Question1.htm",request, response);
  }
  if(operation.equals("password"))
  {
   gotoPage("/Password.htm",request, response);
  }
 if(operation.equals("markah"))
  {
   gotoPage("/Markah.htm",request, response);
  }
 if(operation.equals("skema"))
```

```
    {
      gotoPage("/Skema.htm",request, response);
    }
  }

  public void doPost(HttpServletRequest request, HttpServletResponse response)
 throws ServletException, IOException
  {
   response.setContentType(CONTENT_TYPE);
   PrintWriter out = response.getWriter();

   if(request.getParameter("user").equals(""))
   {
    out.write("Please fill your name in the login box");
   }
   else
   if(request.getParameter("pass").equals(""))
   {
    out.write("Password box not fill in. Please push the back button in your
browser!");
   }
   else
   if(CheckUserAvailable(request.getParameter("user"),
request.getParameter("pass"),true))
```

```
{

 HttpSession session = request.getSession(true);

  try
  {
   if(session.getAttribute("Connection")==null)
   {
     con = jPool.getConnection();
     if(con == null)
     {
      gotoPage("", request, response);
     }
     else
     {
      session.setAttribute("Connection",con);
      System.err.println("Connection");
     }
   }

  }
  catch(Exception ex)
  {
    System.err.println("Exception:" + ex.getMessage());
  }
```

```
      gotoPage("/Question.htm",request,response);

   }
   else
   {
     gotoPage("/Subscribe.htm",request, response);
     if(request.getParameter("nama").equals(""))
     {
      out.write("The form not properly inserted!");
     }
     else
     if(request.getParameter("katalaluan").equals(""))
     {
      out.write("The form not properly inserted!");
     }
     else
     {
      String user = request.getParameter("user");
      String pass = request.getParameter("pass");

      InsertUser(user,pass);
     }

   }
```

```
 }


  public void gotoPage(String address, HttpServletRequest request,
HttpServletResponse response)
     throws ServletException, IOException
  {
   RequestDispatcher dispatcher =
getServletContext().getRequestDispatcher(address);
   dispatcher.forward(request,response);
  }


  public boolean CheckUserAvailable (String sUsername, String sPassword, boolean
bCheckPassword)
  {
   boolean bResult = false;
   Statement stmt ;
   Connection con ;

     try
     {
      jPool = new
JDCConnectionPool("sun.jdbc.odbc.JdbcOdbcDriver","jdbc:odbc:Mastura","","",
"C:\\ConnectionLogPool.log", 1, 4, 30000);
      String query = "Select * from UserInfo where Uname like '" +sUsername+ "'";

        if(bCheckPassword == true)
          query+=" and Pword = '"+sPassword+"'";


      con = jPool.getConnection();
      System.err.println(con);
      stmt = con.createStatement();



     ResultSet rs = stmt.executeQuery(query);
      bResult = rs.next();

      stmt.close();
      jPool.returnConnection((JDCConnection)con);
      System.err.println(stmt);
      System.err.println(con);

  }
  catch (SQLException ex)
```

```
    {
     System.err.println("SQLException:" + ex.getMessage());
    }
    catch(Exception e)
    {
     e.printStackTrace();
    }

   return bResult;

  }

 public String InsertUser(String sUsername, String sPassword)
 {
  try
  {
   jPool = new
JDCConnectionPool("sun.jdbc.odbc.JdbcOdbcDriver","jdbc:odbc:Mastura","","",
"C:\\ConnectionLogPool.log", 1, 4, 30000);
   String InsertData = "insert into UserInfo (Uname, Pword)" + "values(?,?)";

   con = jPool.getConnection();

   PreparedStatement pstmt = con.prepareStatement(InsertData);
   pstmt.clearParameters();
```

```
     pstmt.setString(1, sUsername);

     pstmt.setString(2, sPassword);


     pstmt.executeUpdate();

     pstmt.close();

   }

   catch (SQLException ex)

   {

     System.err.println("SQLException:" + ex.getMessage());

   }

   catch(Exception e)

   {

     e.printStackTrace();

   }


     return "";

 }


 /**Clean up resources*/

 public void destroy() {

 }


}
```

Source Code 2

```
<HTML>
<HEAD>
<Title> JSP mit Bean </Title>
</HEAD>
<body>
<body text="#339999" bgcolor="#CCCCCC"><form
action="http://localhost:8080/controller" method=GET>
<jsp:useBean id="main" scope="session" class="User.UserUtilities" />
<%
 if(main.CheckUser(request.getParameter("user"),true))
 {
%>
<font color="#FF9900"></font><font color="#FF9900"></font><font
color="#FF9900"></font><font color="#FF9900"></font><font
color="#FF9900"></font>
<table width="100%">
  <tr>
    <td><font color="#FF9900"><b><font size="4" color="#339999">Anda telah
mendaftar
      ke dalam sistem ini. Sila tekan butang "Cari Kata Laluan" </font></b>
      <p>&nbsp;</p>
      <input type="submit" name="Submit" value="Cari Kata Laluan">
      <input class="" type="HIDDEN" name="operation" value="password">
</font></td>
<%
 }
 else
 {
%>
</tr>
<%=main.InsertUser(request.getParameter("user"),request.getParameter("pass"))%>
<%
 }
%>

</body>
</html>
```

Source Code 3

```
<HTML>

<HEAD>

<Title> JSP mit Bean </Title>

</HEAD>

<body text="#339999" bgcolor="#CCCCCC"><form

action="http://localhost:8080/controller" method=GET>

<jsp:useBean id="main" scope="session" class="User.UserUtilities" />

<%

if(main.CheckUserAvailable(request.getParameter("user"),request.getParameter("pass

"), true))

  {

%>

<font color="#FF9900"></font><font color="#FF9900"></font><font

color="#FF9900"></font><font color="#FF9900"></font><font

color="#FF9900"></font>

<table width="100%">

  <tr>

    <td><font color="#FF9900"><b><font size="4" color="#339999">Anda telah

mendaftar

    ke dalam sistem ini. Sila tekan butang "Ke Muka Soalan" </font></b>

    <p>&nbsp;</p>

    <input type="submit" name="Submit" value="Ke Muka Soalan">
  <input class="" type="HIDDEN" name="operation" value="question">
    </font></td>
  <%
 }
 else
 {
%>
 </tr>
 <tr>
  <td><font color="#FF9900"><b><font size="4" color="#339999">Pengguna yang
   tidak berdaftar. Sila tekan butang "Daftar" untuk mendaftar ke dalam
sistem</font></b>
    <p>&nbsp;</p>
    <input type="submit" name="Submit" value="Daftar">
    <input class="" type="HIDDEN" name="operation" value="subscribe">
    <%
  }
 %>
    </font></td>
  </tr>
 </table>
 <p>&nbsp;</p>
 <p>&nbsp;</p>
 </body>
 </html>
```

Souce Code 4

```
<HTML>

<HEAD>

<Title> JSP mit Bean </Title>

</HEAD>

<body>


<body text="#339999" bgcolor="#CCCCCC"><form

action="http://localhost:8080/controller" method=GET>

<jsp:useBean id="main" scope="session" class="User.UserUtilities" />


<%=main.SearchPassword(request.getParameter("user"))%>


<%out.println("Kata Laluan anda ialah:"+ main.sPassword);%>


<p>&nbsp;</p>


<input type="submit" name="Submit" value="Pendaftaran Pengguna">

<input class="" type="HIDDEN" name="operation" value="login">


</body>

</html>v
```

Source Code 5

```
<HTML>

<HEAD>

<Title> JSP mit Bean </Title>

</HEAD>

<body>

<body text="#339999" bgcolor="#CCCCCC"><form action="/controller"

method=GET>

<jsp:useBean id="main" scope="session" class="User.UserUtilities" />

<%=main.ViewAnswer(request.getParameter("s1"),request.getParameter("s2"),reques

t.getParameter("s3"))%>

</body>

</html>
```

Source Code 6

```
package Common;

/**
 * Title:        Intelligent System
 * Description:
 * Copyright:    Copyright (c) 2002
 * Company:      University Malaya
 * @author Mastura Ariffin
 * @version 1.0
 */

import java.sql.*;
import java.io.*;

public class CommonUtilities
{

 public String ReplaceHolder(String sSource, String sPlaceHolder, String sReplace)
 {
        try
        {
            int iPos=sSource.indexOf(sPlaceHolder);
              if (iPos>0)
              {
                sSource = sSource.substring(0,iPos)
                  + sReplace
                  + sSource.substring(iPos+sPlaceHolder.length());
              }//if
          }//try
          catch(RuntimeException r)
          {
          }//catch
          return sSource;
    }// public ReplaceHolder

    public String readTemplate(String sTemplateFile)
    {
        String sContent = "";

        try
        {
          FileReader file = new FileReader(sTemplateFile);
          BufferedReader buff = new BufferedReader(file);
          boolean eof = false;
            while (!eof)
        {
            String line = buff.readLine();
            if (line == null)
              eof = true;
            else
              sContent += line;
              //System.out.println(line);
        }
        buff.close();
      }//try

      catch (IOException e)
      {
        System.out.println("Error -- " + e.toString());
      }//catch

      return sContent;

  }//public readTemplate

}// public class commonUtilities
```

Source Code 7

```
package Pool;

/**
 * Title:       Intelligent System
 * Description:
 * Copyright:   Copyright (c) 2002
 * Company:     University Malaya
 * @author Mastura Ariffin
 * @version 1.0
 */

import java.sql.*;
import java.util.*;
import java.io.*;

public class ConnectionReaper extends Thread {

   private JDCConnectionPool pool;
   private long delay = 120000;          // default delay of 2 minutes

  /**
   * Create a new thread instance using the default delay
 *
 *   @param pool JDCConnectionPool that the reaper should monitor
 */
 ConnectionReaper(JDCConnectionPool pool) {
    this.pool = pool;
        this.setDaemon(true);
 }

/**
 * Create a new thread instance
 *
 *   @param pool   JDCConnectionPool that the reaper should monitor
 *   @param delay  sleep period in milliseconds
 */
ConnectionReaper(JDCConnectionPool pool, long delay) {
   this.pool = pool;
   this.delay = delay;
}

public void run() {
   while (true) {
     try {
       sleep(delay);
     } catch( InterruptedException e) { }
       pool.reapConnections();
```

Source Code 8

```
package Pool;

/**

 * Title:        Intelligent System

 * Description:

 * Copyright:    Copyright (c) 2002

 * Company:      University Malaya

 * @author Mastura Ariffin

 * @version 1.0

 */

/**

 *  The JDCConnection.java class represents a JDBC connection in the connection
pool,

 *  and is essentially a wrapper around a real JDBC connection. The JDCConnection
object

 *  maintains a state flag to indicate if the connection is in use and the time the

 *  connection was taken from the pool. This time is used by the ConnectionReaper
class

 *  to identify hanging connections.

 *

 *  @since  April 2001

 */
```

```
import java.sql.*;
import java.util.*;
import java.io.*;

public class JDCConnection implements Connection {

  private JDCConnectionPool pool;
  private Connection connection;
  private boolean inuse;
  private long timestamp;
  private int id;
  private String caller;

  public JDCConnection(Connection connection, JDCConnectionPool pool) {
    this.connection = connection;
    this.pool = pool;
    this.inuse = false;
    this.timestamp = 0;
  }

//=================================================//
//       JDCCONNECTION METHODS           //
//=================================================//
```

```
public synchronized boolean lease() {
  if(inuse)  {
    return false;
  } else {
    inuse = true;
    timestamp = System.currentTimeMillis();
    return true;
  }
}

public boolean validate() {
  try {
    connection.getMetaData();
  } catch (Exception e) {
    return false;
  }
  return true;
}

public boolean inUse() {
  return inuse;
}

public long getLastUse() {
  return timestamp;
}

public void close() throws SQLException {
  pool.returnConnection(this);
}

protected void expireLease() {
  inuse = false;
}

protected Connection getConnection() {
  return connection;
}

// =================================================================//
//     CONNECTION INTERFACE METHODS          //
// =================================================================//

public PreparedStatement prepareStatement(String sql) throws SQLException {
  return connection.prepareStatement(sql);
}

public CallableStatement prepareCall(String sql) throws SQLException {
  return connection.prepareCall(sql);
}
```

```
public Statement createStatement() throws SQLException {
  return connection.createStatement();
}

public String nativeSQL(String sql) throws SQLException {
  return connection.nativeSQL(sql);
}

public void setAutoCommit(boolean autoCommit) throws SQLException {
  connection.setAutoCommit(autoCommit);
}

public boolean getAutoCommit() throws SQLException {
  return connection.getAutoCommit();
}

public void commit() throws SQLException {
  connection.commit();
}

public void rollback() throws SQLException {
  connection.rollback();
}
```

```
public boolean isClosed() throws SQLException {
  return connection.isClosed();
}

public DatabaseMetaData getMetaData() throws SQLException {
  return connection.getMetaData();
}

public void setReadOnly(boolean readOnly) throws SQLException {
  connection.setReadOnly(readOnly);
}

public boolean isReadOnly() throws SQLException {
  return connection.isReadOnly();
}

public void setCatalog(String catalog) throws SQLException {
  connection.setCatalog(catalog);
}

public String getCatalog() throws SQLException {
  return connection.getCatalog();
}

public void setTransactionIsolation(int level) throws SQLException {
    connection.setTransactionIsolation(level);
  }

  public int getTransactionIsolation() throws SQLException {
    return connection.getTransactionIsolation();
  }

  public SQLWarning getWarnings() throws SQLException {
    return connection.getWarnings();
  }

  public void clearWarnings() throws SQLException {
    connection.clearWarnings();
  }

  public Map getTypeMap() throws SQLException {
    return connection.getTypeMap();
  }

  public void setTypeMap(Map map) throws SQLException {
    connection.setTypeMap(map);
  }

  public PreparedStatement prepareStatement(String str, int i, int j) throws
SQLException {
    return connection.prepareStatement(str, i, j);
  }


  public CallableStatement prepareCall(String str, int i, int j) throws SQLException {
    return connection.prepareCall(str, i, j);
  }

  public Statement createStatement(int i, int j) throws SQLException {
    return connection.createStatement(i, j);
  }

}
```

Source Code 9

```
package Pool;

/**
 * Title:        Intelligent System
 * Description:
 * Copyright:    Copyright (c) 2002
 * Company:      University Malaya
 * @author Mastura Ariffin
 * @version 1.0
 */

/**
 *  The JDCConnection.java class represents a JDBC connection in the connection
pool,
 *  and is essentially a wrapper around a real JDBC connection. The JDCConnection
object
 *  maintains a state flag to indicate if the connection is in use and the time the
 *  connection was taken from the pool. This time is used by the ConnectionReaper
class
 *  to identify hanging connections.
 *
 *  @since  April 2001
 */
```

```
import java.sql.*;
import java.util.*;
import java.io.*;

public class JDCConnection implements Connection {

  private JDCConnectionPool pool;
  private Connection connection;
  private boolean inuse;
  private long timestamp;
  private int id;
  private String caller;

  public JDCConnection(Connection connection, JDCConnectionPool pool) {
    this.connection = connection;
    this.pool = pool;
    this.inuse = false;
    this.timestamp = 0;
  }

// ==================================================================//
//        JDCCONNECTION METHODS          //
// ==================================================================//
```

```
public synchronized boolean lease() {
  if(inuse)  {
   return false;
  } else {
   inuse = true;
   timestamp = System.currentTimeMillis();
   return true;
  }
}

public boolean validate() {
  try {
    connection.getMetaData();
  } catch (Exception e) {
    return false;
  }
  return true;
}

public boolean inUse() {
  return inuse;
}

public long getLastUse() {
    return timestamp;
  }
  public void close() throws SQLException {
    pool.returnConnection(this);
  }
  protected void expireLease() {
    inuse = false;
  }
  protected Connection getConnection() {
    return connection;
  }
// ==================================================//
//     CONNECTION INTERFACE METHODS          //
// ==================================================//
public PreparedStatement prepareStatement(String sql) throws SQLException {
  return connection.prepareStatement(sql);
}

public CallableStatement prepareCall(String sql) throws SQLException {
  return connection.prepareCall(sql);
}
```

```
public Statement createStatement() throws SQLException {
  return connection.createStatement();
}

public String nativeSQL(String sql) throws SQLException {
  return connection.nativeSQL(sql);
}

public void setAutoCommit(boolean autoCommit) throws SQLException {
  connection.setAutoCommit(autoCommit);
}

public boolean getAutoCommit() throws SQLException {
  return connection.getAutoCommit();
}

public void commit() throws SQLException {
  connection.commit();
}

public void rollback() throws SQLException {
  connection.rollback();
}
```

```
public boolean isClosed() throws SQLException {
  return connection.isClosed();
}

public DatabaseMetaData getMetaData() throws SQLException {
  return connection.getMetaData();
}

public void setReadOnly(boolean readOnly) throws SQLException {
  connection.setReadOnly(readOnly);
}

public boolean isReadOnly() throws SQLException {
  return connection.isReadOnly();
}

public void setCatalog(String catalog) throws SQLException {
  connection.setCatalog(catalog);
}

public String getCatalog() throws SQLException {
  return connection.getCatalog();
}

public void setTransactionIsolation(int level) throws SQLException {
    connection.setTransactionIsolation(level);
  }


  public int getTransactionIsolation() throws SQLException {
    return connection.getTransactionIsolation();
  }


  public SQLWarning getWarnings() throws SQLException {
    return connection.getWarnings();
  }


  public void clearWarnings() throws SQLException {
    connection.clearWarnings();
  }


  public Map getTypeMap() throws SQLException {
    return connection.getTypeMap();
  }


  public void setTypeMap(Map map) throws SQLException {
    connection.setTypeMap(map);
  }


  public PreparedStatement prepareStatement(String str, int i, int j) throws
SQLException {
      return connection.prepareStatement(str, i, j);
    }

    public CallableStatement prepareCall(String str, int i, int j) throws SQLException {
      return connection.prepareCall(str, i, j);
    }

    public Statement createStatement(int i, int j) throws SQLException {
      return connection.createStatement(i, j);
    }

}package Pool;

/**
 * Title:        Intelligent System
 * Description:
 * Copyright:    Copyright (c) 2002
 * Company:      University Malaya
 * @author Mastura Ariffin
 * @version 1.0
 */

/**
 * JDCConnectionPool makes connections available to calling program in its
 *  getConnection method. This method searches for an available connection in
 * the connection pool. If no connection is available from the pool, a new
 * connection is created. If a connection is available from the pool, the
 * getConnection method leases the connection and returns it to the calling
 * program.
 *
 * @author jkeyes
 * @since  April 2001
 */


import java.sql.*;
import java.util.*;
import java.io.*;
import ConnectionReaper.*;
import JDCConnection.*;


public class JDCConnectionPool {

   private Vector connections;
   private ConnectionReaper reaper;
   private String driverName, url, user, password;
   private long timeout = 60000;                    // default timeout in milliseconds
   private static final int maxAttempts = 10;          // max number of times to establish a
connection
```

```
   private static final long sleepTime = 1000L;      // delay in open connection loop in
millis
   private int maxpoolsize = 10;
   private int minpoolsize = 1;
   private String logFileName = "JDCConnectionPool.log";
   private PrintStream log;

   // public static void main(String[] args) {}

  /**
   * Create a new instance of the pool using the default logFileName, minpoolsize
   * maxpoolsize, and timeout.
   *
   * @param driverName  String holding the name of the JDBC driver
   * @param url         String holding the location of the database
   * @param user        String with the database login id
   * @param password    String with the database password
   * @exception
   */
  public JDCConnectionPool(String driverName, String url, String user, String
password)
      throws Exception {
   this.init(driverName, url, user, password);
  }
```

```
 /**
  * Initialization routine run at object creation
  *
  *   @param driverName   String holding the name of the JDBC driver
  *   @param url          String holding the location of the database
  *   @param user         String with the database login id
  *   @param password     String with the database password
  *   @exception          ClassNotFoundException
  *   @exception          InstantiationException
  *   @exception          IllegalAccessException
  *   @exception          SQLException
  */
 private void init(String driverName, String url, String user, String password)
     throws ClassNotFoundException, InstantiationException,
IllegalAccessException,
     SQLException {
   this.driverName = driverName;
   this.url = url;
   this.user = user;
   this.password = password;

   if (this.minpoolsize < 1)
     this.minpoolsize = 1;
   /**
    * Create a new instance of the pool
    *
    *  @param driverName   String holding the name of the JDBC driver
    *  @param url          String holding the location of the database
    * @param user          String with the database login id
    *  @param password     String with the database password
    *  @param minpoolsize  int value for the minimum number of connections to hold
    *                  in the pool
    *  @param maxpoolsize  int value for the maximum number of connections to
hold
    *                  in the pool
    */
   public JDCConnectionPool(String driverName, String url, String user, String
password,
      String logFileName, int minpoolsize, int maxpoolsize, long timeout) throws
Exception {
     this.logFileName = logFileName;
     this.minpoolsize = minpoolsize;
     this.maxpoolsize = maxpoolsize;
     this.timeout = timeout;
     this.init(driverName, url, user, password);
   }
```

```
    try {
      this.log = new PrintStream(new FileOutputStream(logFileName));
    } catch (FileNotFoundException e) {
      System.out.println("Could not create logfile");
      log = System.out;
    }
    try {
      java.sql.DriverManager.registerDriver (
          (java.sql.Driver)Class.forName (driverName).newInstance()
      );
    } catch (ClassNotFoundException e) {
      log.println(new java.util.Date() + ":  Unable to locate " + driverName + ": "
          + e.getMessage());
      throw e;
    } catch (InstantiationException e) {
      log.println(new java.util.Date() + ":  Unable to instantiate " + driverName + ": "
          + e.getMessage());
      throw e;
    } catch (IllegalAccessException e) {
      log.println(new java.util.Date() + ":  Access violation using " + driverName + ":
"
        + e.getMessage());
      throw e;
    } catch (SQLException e) {
      log.println(new java.util.Date() + ":  Unable to register " + driverName + ": "
        + e.getMessage());
      throw e;
    }

    connections = new Vector(maxpoolsize);
    reaper = new ConnectionReaper(this, this.timeout);
    reaper.start();
  }

  /**
   * Method to remove stale connections
   */
  public synchronized void reapConnections() {
    long stale = System.currentTimeMillis() - timeout;
    Enumeration connectionlist = connections.elements();

    while((connectionlist != null) && (connectionlist.hasMoreElements())) {
      JDCConnection connection = (JDCConnection)connectionlist.nextElement();

      if((connection.inUse()) && (stale > connection.getLastUse())) {
      //&& (!connection.validate())) {
        log.println(new java.util.Date() + ":  Stale connection " +
connection.toString()
```

```
        + " has been terminated.");
            removeConnection(connection);
        }
    }
}

/**
 * Method to close all open connections
 */
public synchronized void closeConnections() {

  Enumeration connectionlist = connections.elements();

  while((connectionlist != null) && (connectionlist.hasMoreElements())) {
    JDCConnection connection = (JDCConnection)connectionlist.nextElement();
    removeConnection(connection);
  }
}


/**
 * Method to remove a connection from the pool.  Catches a SQL
 * Exception and logs the event, but does not rethrow the Exception.
 *
```

```
    *   @param connection  JDCConnection to be removed from the pool
   */
  private synchronized void removeConnection(JDCConnection connection){
    if (connection != null){
      try {
        if (!connection.isClosed())
          connection.close();
      } catch (SQLException e) {
        log.print(new java.util.Date() + ":  Error closing connection " +
connection.toString()
          + ": " + e.getMessage());
      } finally {
      connection = null;
      }
    }
    connection.removeElement(connection);
  }

  /**
   * Method returns a connection from the pool.  It ensures that the
   * minimum number of connections have been created and that the maximum
   * pool size is not exceeded.
   *
   *  @return connection  JDCConnection returned as a connection.
   *            If the maximum attempts to get the connection have
   *            been exceeded, this object will be null.
   */
  public synchronized Connection getConnection() throws SQLException {

    JDCConnection connection = null;
    int attemptCounter = 0;
    int connectionsToLaunch = 1;

    while (true) {
      attemptCounter++;
      if (getCurrentPoolSize() >= this.minpoolsize) { // don't look unless pool is
minimum size
        for(int i = 0; i < connections.size(); i++) {
          connection = (JDCConnection)connections.elementAt(i);
          if (connection.lease()) {       // claim the connection and exit if possible
            break;
          } else {
            connection = null;
          }
        }
      } else {
        connectionsToLaunch = this.minpoolsize - getCurrentPoolSize();
      }
```

```
        if (connection == null) {  // Try to create a connection if none were free
          if (getCurrentPoolSize() < this.maxpoolsize) {
            for (int i = 1; i <= connectionsToLaunch; i++) {
              try {
                connection = getNewJDCConnection();
                log.println(new java.util.Date() + ":  Opened new connection: "
                  + connection.toString());
                // return all but the last connection created to the pool
                if (i < connectionsToLaunch)
                  returnConnection(connection);
              } catch (SQLException e) {
                log.println(new java.util.Date() + ":  SQL Error opening connection: "
                  + e.getMessage());
                throw e;
              }
            }
          } else {          // max connections have been opened
            try {
              Thread.sleep(sleepTime);
            } catch (InterruptedException e) {}
            log.println(new java.util.Date() + ":  Attempt " + attemptCounter
              + " failed to establish a connection.");
          }
        }
      if (connection != null || attemptCounter == this.maxAttempts) {
        break;
      }
    } // end while

  return connection;
}

/**
 * Method to create a new JDCConnection
 *
 * @exception SQLException
 */
private JDCConnection getNewJDCConnection() throws SQLException {
  JDCConnection connection = null;

  Connection conn = DriverManager.getConnection(url, user, password);
  connection = new JDCConnection(conn, this);
  connection.lease();
  connections.addElement(connection);

  return connection;
}
```

```
  /**
   * Method to return a JDCConnection to the pool.
   *
   *  @param  JDCConnection connection to be returned for reuse.
   */
  public synchronized void returnConnection(JDCConnection connection) {
    connection.expireLease();
  }


  /**
   * Method to return the current number of active connections.
   */
  public int getCurrentPoolSize() {
    return connections.size();
  }


  /**
   * Method to return the minimum size of pool.
   */
  public int getMinPoolSize() {
    return this.minpoolsize;
  }
```

```
   /**
    * Method to return the maximum size of pool.
    */
   public int getMaxPoolSize() {
     return this.maxpoolsize;
   }

}
```

Source Code 10

```
package User;

/**
 * Title:       Intelligent System
 * Description:
 * Copyright:   Copyright (c) 2002
 * Company:     University Malaya
 * @author Mastura Ariffin
 * @version 1.0
 */

import java.io.*;
import java.sql.*;
import java.util.*;
import Common.*;
import Pool.*;

public class UserUtilities
{

  private Connection con;
```

```
   private CommonUtilities Util;
   JDCConnectionPool jPool;
   public String sPassword = "";
   //public String sTemplate = "";

 public static void main (String args[])
 {
 }

 public UserUtilities()
 {
  Util = new CommonUtilities();
  try
    {
     jPool = new
JDCConnectionPool("sun.jdbc.odbc.JdbcOdbcDriver","jdbc:odbc:Mastura","","",
"C:\\ConnectionLogPool.log", 1, 10, 30000);
    }
     catch (Exception e)
     {
     }

 }
```

```
//@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
//
//@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
 public boolean CheckUserAvailable(String sName, String sPass, boolean
bCheckPassword)
 {
   boolean bResult = false;
   Statement stmt;

  try
  {

   String query = "SELECT * FROM UserInfo WHERE Uname LIKE
'"+sName+"'";

   if (bCheckPassword == true)
    query+=" AND Pword = '"+sPass+"'";


   con = jPool.getConnection();
   stmt = con.createStatement();
   ResultSet rs = stmt.executeQuery(query);
```

```
    {
      String query = "SELECT * FROM UserInfo WHERE Uname LIKE
 '"+sUsername+"'";

     con = jPool.getConnection();
     stmt = con.createStatement();
     ResultSet rs = stmt.executeQuery(query);

     bResult = rs.next();

   }//try block
   catch (SQLException ex)
   {
     System.err.println("SQLException:" + ex.getMessage());
   }
   return bResult;

 }

//@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

```
      bResult = rs.next();



    }//try block
    catch (SQLException ex)
    {
       System.err.println("SQLException:" + ex.getMessage());
    }
    return bResult;
  }//boolean


//@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@
// check user name
//@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@
 public boolean CheckUser(String sUsername, boolean bUser)
 {
   boolean bResult = false;
   Statement stmt;


   try
  </font></td>
<%
  }
  else
  {
%>
</tr>
<%=main.InsertUser(request.getParameter("user"),request.getParameter("pass"))%>
<%
  }
%>

</body>
</html>
```

Source Code 11

```
package MarkingSystem;

/**
 * Title:        Markng System
 * Description:
 * Copyright:    Copyright (c) 2001
* Company:      Universiti Malaya
 * @author Mas
 * @version 1.0
 */

import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;
import java.text.*;

public class CubaLok {

 protected boolean stopRequested = false;

  char P[], T[];
  int  M,   N;
  int next[];

  int  Position, Current;

  int  Comparisons = 0;

  boolean Match[], Trivial[], Mismatch, Found;


  private static Object initLock = new Object();


  protected void pause(int position, boolean trivial[], int current, boolean mismatch,
boolean found, int comparisons) throws Exception {

      if (stopRequested) {

          throw new Exception("Pattern Matching Algorithm");

      }

  }


  public void stop() {

      stopRequested = true;

  }


  //void find(char t[], int n, char p[], int m) throws Exception

  void clear_table(boolean table[], int m) {

    for (int i=0; i<m; i++) table[i] = false;

  }

  int maximum(int x, int y) {

      if (x > y) return x; else return y;

  }
  int minimum(int x, int y) {
      if (x < y) return x; else return y;
  }


//@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@
 public boolean set_pattern_and_text(String pattern, String text)
 {
   boolean bResult = false;
  N = text.length();
  M = pattern.length();
  T =  new char [N];
  P =  new char [M];
  text.getChars(0, N, T, 0);
  pattern.getChars(0, M, P, 0);
  try
  {
   if(find(T,N,P,M))
   {
    bResult = true;
   }
  }
 catch(Exception e)
 {
    }
    return bResult;
  }

//@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
public boolean find(char t[], int n, char p[], int m) throws Exception
{
    boolean bResult= false;
    boolean trivial[] = new boolean[m];
    int cnt = 0;

    clear_table(trivial, m);
    for (int i = 0; i + m <= n; i++)
    {
     pause(i, trivial, -1, false, false, cnt);

     for (int j = 0; j < m; j++)
     {
      cnt++;
         if (t[i+j] == p[j])
      {
       pause(i, trivial, j, false, (j+1==m), cnt);
```

```
          bResult = true;

        }else

        {

          pause(i, trivial, j, true , false, cnt);

          break;

        }

      }

    }

    return bResult;

  }
//@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

  public static final String [] toLowerCaseWordArray(String text) {

    if (text == null || text.length() == 0) {

        return new String[0];

    }


    ArrayList wordList = new ArrayList();

    BreakIterator boundary = BreakIterator.getWordInstance();

    boundary.setText(text);

    int count = 0;

    int start = 0;
    for (int end = boundary.next(); end != BreakIterator.DONE;
          start = end, end = boundary.next())
    {
      String tmp = text.substring(start,end).trim();
      // Remove characters that are not needed.
      tmp = replace(tmp, "+", "");
      tmp = replace(tmp, "/", "");
      tmp = replace(tmp, "\\", "");
      tmp = replace(tmp, "#", "");
      tmp = replace(tmp, "*", "");
      tmp = replace(tmp, ")", "");
      tmp = replace(tmp, "(", "");
      tmp = replace(tmp, "&", "");
      if (tmp.length() > 0) {
        wordList.add(tmp);
        count++;
      }
    }
    return (String[]) wordList.toArray(new String[wordList.size()]);
}

/**
 * A list of some of the most common words. For searching and indexing, we
 * often want to filter out these words since they just confuse searches.
```

```
  * The list was not created scientifically so may be incomplete :)
  */
 private static final String [] commonWords =  new String [] {
    "a", "and", "as", "at", "be", "do", "i", "if", "in", "is", "it", "so",
    "the", "to"
 };
 private static Map commonWordsMap = null;

 /**
  * Returns a new String array with some of the most common English words
  * removed. The specific words removed are: a, and, as, at, be, do, i, if,
  * in, is, it, so, the, to
  */
 public static final String [] removeCommonWords(String [] words) {
    //See if common words map has been initialized. We don't statically
    //initialize it to save some memory. Even though this a small savings,
    //it adds up with hundreds of classes being loaded.
    if (commonWordsMap == null) {
      synchronized(initLock) {
        if (commonWordsMap == null) {
          commonWordsMap = new HashMap();
          for (int i=0; i<commonWords.length; i++) {
            commonWordsMap.put(commonWords[i], commonWords[i]);
          }
        }
```

```
          }
      }
      //Now, add all words that aren't in the common map to results
      ArrayList results = new ArrayList(words.length);
      for (int i=0; i<words.length; i++) {
         if (!commonWordsMap.containsKey(words[i])) {
            results.add(words[i]);
         }
      }
      return (String[])results.toArray(new String[results.size()]);
   }

public static final String replace( String line, String oldString, String newString )
{
   if (line == null) {
      return null;
   }
   int i=0;
   if ( ( i=line.indexOf( oldString, i ) ) >= 0 ) {
      char [] line2 = line.toCharArray();
      char [] newString2 = newString.toCharArray();
      int oLength = oldString.length();
      StringBuffer buf = new StringBuffer(line2.length);
      buf.append(line2, 0, i).append(newString2);
      i += oLength;
```

```
            int j = i;
            while( ( i=line.indexOf( oldString, i ) ) > 0 ) {
               buf.append(line2, j, i-j).append(newString2);
               i += oLength;
               j = i;
            }
            buf.append(line2, j, line2.length - j);
            return buf.toString();
        }
        return line;
    }

//@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
/*
public boolean[] Corak(String sJawapan)
{

    //String [] s = cuba.toLowerCaseWordArray(request.getParameter(sJawapan));
    int k = 0;
    String result = "";
    String patternz="";
    //String [] pattern = {"organisma","dalaman","stabil","mengekalkan"};
    k = s.length;
    if (k > 0)
    {
     result = s[0];

      for (int i= 1 ; i < k; i++)
      {
       result += s[i] ;
      }
       result = result.toString();

       for (int i = 0; i < pattern.length; i++)
       {
           patternz = pattern.toString();
            patternz=pattern[i];
             boolean [] bHasil = {cuba.set_pattern_and_text(patternz,result)};
              cuba.set_pattern_and_text(patternz,result);
             for(int j = 0;j< bHasil.length;j++)
             {
              System.err.print(bHasil[j]+ " ");
             }
       }
    }
   }

*/
```

```
/*

public String[] StringtoArray(String s, String sep)

{

  StringBuffer buf = new StringBuffer("s1");

  int arraysize = 1;

  for (int i=0;i<buf.length();i++)

  {

  if (sep.indexOf(buf.charAt(i)) != -1)

  {

   arraysize++;

  }

  String [] elements  = new String [arraysize];

  int y,z = 0;

  if ( buf.toString().indexOf(sep) != -1 ) {

  while (  buf.length() > 0 ) {

  if ( buf.toString().indexOf(sep) != -1 ) {

   y =  buf.toString().indexOf(sep);

   if ( y != buf.toString().lastIndexOf(sep) ) {

    elements[z] = buf.toString().substring(0, y ); z++;

    buf.delete(0, y + 1);

   }

   else if ( buf.toString().lastIndexOf(sep) == y ) {

    elements[z] = buf.toString().substring(0, buf.toString().indexOf(sep) );z++;

    buf.delete(0, buf.toString().indexOf(sep) + 1);
```

```
            elements[z] = buf.toString();z++;
            buf.delete(0, buf.length() );
          }
        }
      }
    }
    else {elements[0] = buf.toString(); }
    buf = null;
    return elements;
  }
}


    public String ArrayToString(String s[], String sep)
    {
    int k;
    String result = "";

    k = s.length;
    if (k > 0) {
      result = s[0];
        for (int i= 1 ; i > k; i++) {
          result += sep + s[i] ;
        }
      }
   return result;
```

Source Code 12

```
package MarkingSystem;

import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;

public class Servlet1 extends HttpServlet {

   CubaLok cuba;


 private static final String CONTENT_TYPE = "text/html";
 /**Initialize global variables*/
 public void init() throws ServletException
 {
  cuba = new CubaLok();
 }
 /**Process the HTTP Get request*/
 public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
  response.setContentType(CONTENT_TYPE);
```

```
    PrintWriter out = response.getWriter();

    String operation = request.getParameter("operation");
     if (operation == null)
     {
      gotoPage("/index.htm",request,response);
     }
    else
    {


    }
  }

 public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
 {
  response.setContentType(CONTENT_TYPE);
  PrintWriter out = response.getWriter();

  if(request.getParameter("s1").equals(""))
  {
   out.write("Please fill your name in the login box");
  }
  else
  {
```

```
String [] s = cuba.toLowerCaseWordArray(request.getParameter("s1"));
int k = 0;
String result = "";
String patternz="";
String [] pattern = {"organisma","dalaman","stabil","mengekalkan"};
k = s.length;
if (k > 0)
{
 result = s[0];

  for (int i= 1 ; i < k; i++)
  {
    result += s[i] ;
  }
   result = result.toString();

   for (int i = 0; i < pattern.length; i++)
   {
       patternz = pattern.toString();
        patternz=pattern[i];
         boolean [] bHasil = {cuba.set_pattern_and_text(patternz,result)};
          cuba.set_pattern_and_text(patternz,result);
         for(int j = 0;j< bHasil.length;j++)
         {
          System.err.print(bHasil[j]+ " ");
```

```
                    }
                }
            }
        }
    }

//@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
  public void gotoPage(String address, HttpServletRequest request,
HttpServletResponse response)
    throws ServletException, IOException
  {
    RequestDispatcher dispatcher =
getServletContext().getRequestDispatcher(address);
    dispatcher.forward(request,response);
  }


 /**Clean up resources*/
 public void destroy() {
 }
}
```

REFERENCES

REFERENCES

1. Kendall,Kenneth E.,&Kendll,Julie E.(1999).Systes Analysis and Design:International Edition.4thed.New Jersey : Prentice Hall International,Inc.
2. Pressman,Roger S.(2001) .Software engineering;A Practitioner's Approach.5th ed.New York : McGraw-Hill Companies,Inc.
3. Lucas,Peter.,Linda Van Der Gaag.(1991).Principles of Expert System.:1st ed.britain:Addison – Wesley Publishing Company,Inc.
4. Charniak,E and McDermott,D.(1986).Introduction to Artificial Intelligence.Reading,MA:Addison-Wesley Publishing Company,Inc.
5. www.dcs.gla.ac.uk/Keith
6. ciir.cs.umass.edu
7. www.google.com
8. knowledge-technologies.com
9. http:// ipdwsb.np.edu.sg/lt/feb00/innov.htm
10. AHO, A.V., 1990, Algorithms for finding patterns in strings. in *Handbook of Theoretical Computer Science, Volume A, Algorithms and complexity*, J. van Leeuwen ed., Chapter 5, pp 255-300, Elsevier, Amsterdam
11. CROCHEMORE, M., HANCART, C., 1999, Pattern Matching in Strings, in *Algorithms and Theory of Computation Handbook*, M.J. Atallah ed., Chapter 11, pp 11-1--11-28, CRC Press Inc., Boca Raton, FL.
12. GUSFIELD, D., 1997, *Algorithms on strings, trees, and sequences: Computer Science and Computational Biology*, Cambridge University Press.

13. WATSON, B.W., 1995, *Taxonomies and Toolkits of Regular Language Algorithms*, Ph. D. Thesis, Eindhoven University of Technology, The Netherlands