AUTOMATIC QUESTION DETERMINATION LEVEL BASED ON OBE USING BRUTE FORCE STRING MATCHING

LOO ZHANG BIN

UNIVERSITI MALAYSIA PAHANG

AUTOMATIC QUESTION DETERMINATION LEVEL BASED ON OBE USING BRUTE FORCE STRING MATCHING

LOO ZHANG BIN

THESIS SUBMITTED IN FULFILMENT OF THE DEGREE OF COMPUTER SCIENCE (SOFTWARE ENGINEERING)

FACULTY OF COMPUTER SYSTEM AND SOFTWARE ENGINEERING

UNIVERSITI MALAYSIA PAHANG

2013

UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS

JUDUL: …….

SESI PENGAJIAN: ……

SAYA ……(HURUF BESAR)

Mengaku membenarkan tesis/laporan PSM ini disimpan di Perpustakaan Universiti Malaysia Pahang dengan syarat-syarat kegunaan seperti berikut:

1. Tesis/Laporan adalah hakmilik Universiti Malaysia Pahang.
2. Perpustakaan Universiti Malaysia Pahang dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institut pengajian tinggi.
4. **Sila tandakan (√)

☐	SULIT	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972) *
☐	TERHAD	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan) *
☐	TIDAK TERHAD	

Disahkan Oleh

……. ……

Alamat tetap: Penyelia:

Tarikh:…… Tarikh:……..

*Sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali sebab dan tempoh tesis/laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

SUPERVISOR DECLARATION

I hereby declare that I have read this thesis and in my opinion this thesis/report is sufficient in terms of scope and quality for the award of the degree of Bachelor of Computer Science ( )

Signature :……

Supervisor Name:

Date :……

DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries which have been duly acknowledged.

Date

Loo Zhang Bin

CB10036

ACKNOWLEDGMENTS

First and foremost, I would like to thank the one and only God for giving me patience and good health throughout the duration of this Degree research. Other than that, I would like to express my gratitude to my supervisor and academic advisor, Dr. Rohani Binti Abu Bakar for her advices, guidance, and support for me to finish up my thesis in this semester.

Moreover, I would like to thank University Malaysia Pahang for giving me the opportunity to further my studies as a degree student and learn more knowledge, which is helpful for my future. I deeply thanks to all the Faculty of Computer System and Software Engineering's staff for help me during my study time.

Lastly, I would also like to thank to my family and fellow friends who gave me support in everything. I will never forget those who help me out in my studies. Thank you all for spend so much time to help me. Thanks.

ABSTRACT

Applying outcome-based education (OBE) in teaching and learning at the tertiary education level has become an important initiative in Malaysia as taken by the government. However, since the concept of OBE is relatively new, some lecturers face difficulty in evaluating the level of question that they produce. Hence this research aim is to develop an automatic question determination level based on OBE using brute force string matching prototype. The focus on this system is to generate a final exam question template with specific fonts and spacing and to evaluate the level of final exam question based upon the teaching and learning taxonomy published by UPM using string matching technique. The major findings that emerged are as follows: a) the suitable string matching technique to be use is brute force string matching due to the nature to the system and development constraints, b) all the outcomes are categorized into two main domains, including the Cognitive and Psychomotor domains. Furthermore, the development process is done using waterfall model software development life cycle. In conclusion, this system is to help user reduce the process and the workload of generating a final exam template.

ABSTRAK

Menggunakan hasil berasaskan pendidikan (OBE) dalam pengajaran dan pembelajaran di peringkat pengajian tinggi telah menjadi satu inisiatif penting di Malaysia seperti yang diambil oleh government. Walau bagaimanapun sejak konsep OBE agak baru, beberapa orang pensyarah menghadapi kesukaran dalam menilai tahap soalan yang mereka produce. Oleh itu tujuan kajian ini ialah untuk membangunkan tahap penentuan soalan automatik berdasarkan OBE menggunakan kekerasan perkataan kasar sepadan prototaip. Fokus sistem ini adalah untuk menjana peperiksaan akhir template soalan dengan fon tertentu dan spacingand untuk menilai tahap soalan peperiksaan akhir berdasarkan pengajaran dan pembelajaran taksonomi diterbitkan oleh UPM menggunakan teknik padanan tali. Penemuan utama yang muncul adalah seperti berikut: a) tali sesuai teknik yang sepadan untuk digunakan adalah kuasa perkataan kasar yang hampir sama kerana sifat sistem dan kekangan pembangunan, b) semua hasil dikategorikan kepada dua bidang utama termasuk kognitif dan domain psikomotor. Tambahan pula proses pembangunan dilakukan dengan menggunakan model air terjun perisian kitaran hayat pembangunan. Kesimpulannya sistem ini adalah untuk membantu pengguna mengurangkan proses dan beban kerja menjana template peperiksaan akhir.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

UMP - Universiti Malaysia Pahang

UPM - Universiti Putra Malaysia

OBE - Outcome-Based Education

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Research Background

The move towards applying outcome-based education (OBE) in teaching and learning at the tertiary education level has become an important initiative in Malaysia as taken by the government. Through Board of Engineers, Malaysia (2009) has admitted to Washington Accord; which is a multinational agreement which recognizes the substantial equivalency of engineering degree programs. One of the key requirements for full membership of the Accord is to fully employ elements of outcome-based education.

The focus to attain certain specified results in terms of individual student learning is the educational approach of OBE. The achievement program needed to measure so that continues improvement can be done to upgrade the quality of graduates. All the outcomes are categorized into three main domains, including the Cognitive, Psychomotor and Affective domains.

The cognitive domains describe knowledge and the development of student's intellectual skills. These domains consist of six major classes starting from the simplest conduct (understanding) to the most intricate (evaluation). Furthermore, the psychomotor domain expresses the physical or technical skill that should be achieved by each student. This skill includes of coordination, physical movement and use of the motor skill areas and so forth. Progress of these skills involves practice that is measured in terms of techniques,

distance, precision, speed, or procedure. The skill evaluates the range from perception (lowest skill) until to origination (highest skill).

Affective domains define the object emotionally, for example, appreciation, motivation, enthusiasms, feelings, attitudes, and values. In the recent practices, affective domains will evaluate together with other soft skills elements such as leadership, team working, communication skills, critical thinking and etc.. However, in this study we do not consider other assessments, which are normally happened in the offered courses except for final exam. In this study, we do not consider these following soft skills elements as elements of final exam questions. If these questions are available, we will group it as an affective domain question.

Advantage of OBE to be implemented at in the university will always alert and concerned about the quality of the graduate produced through the development of more systematic, innovative and flexible teaching methods. Increase in student exposure to professional practice through industrial training, site visits and industry linked projects or assignments will be encouraged.

UMP as a one of IPTA's at Malaysia has enforced by MOHE to adopt OBE in the teaching and learning activities, specifically in preparing teaching and assessment materials. Beside UMP, there are others IPTA's that has adopted successfully this OBE in their curriculum such as UPM, USM, UNIMAS, UM and etc. As a new university, UMP has chosen UPM as a model to develop own OBE approach in the campus. As a result, OBE has started implemented at UMP since 2006. Until now, UMP has adopted teaching and learning taxonomy guideline published by CADe, UPM in developing teaching and learning material. Thus, this project will employ the same material as a guideline in order to evaluate the level of questions.

1.2 Problem Statement

Since the concept of OBE is relatively new, some lecturers face difficulty in evaluating the level of question that they produce. As instances, FSKKP decided that final exam question must consist of at least 60% cognitive according to different program with its required level; diploma programs are at C3 level while degree programs are at C4 level. As consequences of difficult to get the same idea and understanding in evaluating the question level, a proper guide is urgently needed to ensure the standard and quality of final exam paper is acceptable.

In producing the high standard quality of final exam paper, some lecturer too focuses on the questions itself and may neglect the formatting issues. It is also because of the process is quite tedious. The final exam template generator will help lecturers to focus more on producing quality questions and leave the formatting issues to the generator.

1.3 Objectives

The goals to be achieved on the project are the following:

i. To study string matching technique in order to evaluate the level of questions.
ii. To develop a prototype that will generate a final exam question template with standard fonts and spacing.
iii. To evaluate the level of the final exam questions according to the teaching and learning taxonomy published by UPM using string matching.

1.4 Scope

The boundary of the project and users are as follow:

i. The template consists of two types of exam question, which are structured and essay questions.
ii. Implementation of teaching and learning taxonomy is published by Center for Academic Development (CADe), UPM in the system.
iii. Two main domains, including the Cognitive and Psychomotor domains.
iv. Generate PDF templates that consist of final examination paper, schema for final examination paper and OBE report for the final examination paper.

1.5 Thesis Organization

This thesis consists of six (6) chapters. Chapter 1 Introduction, is to give awareness to the readers about the propose topic. Chapter 2 Literature Review, purpose of this chapter is to explain about the selected project. Next Chapter 3 Methodology will discuss the overall approach and framework of research. Chapter 4 Design and Implementation are to develop the framework and model through flow work. Further on, record all the processes involved in research development and describe how it was structured. Chapter 5 Result and Discussion, will explain about the findings or the results from the data analysis. Lastly, Chapter 6 Conclusion, to make a summary for research that has been done.

1.6 Conclusion

The problems faced by lecturers in generating final exam template with quality standards, has motivated the development of automatic question determination level based on OBE using brute force string matching hence should be implemented and applied in the related sector. Furthermore, if the system is successfully implemented it will promote Malaysia's quality education to a higher-level competing internationally equally among other renown universities.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The practice of judging the worth as an individual is called an assessment, while educational assessment involves evaluating and collecting data evolving from planned learning programs or activities. The assessment in University Malaysia Pahang composes of 60% carry marks and 40% final exam. Carry marks consist of cumulative marks from lab sheet, assignment, test and etc.

An assessment practice that is gaining reputation is standards based assessment, which involves measuring student's ability on well defined course objectives (Tomlinson and McTighe, 2006). Standardized assessment is advantageous for several reasons. First of all, it can yield quantifiable information that can be used to screen out students who in need of further assessing. Information following the student's areas of strength and weakness are provided. Lastly, the most important of results is to be interpreted and ideas about an individual's skills are generalized. In Malaysia, standardized assessment is applied with outcome-based education (OBE).

The employment of elements of Outcome-Based Education has earned Malaysia full rights of participation in the Washington Accord. The responsible bodies in each of the current signatory countries will distinguish the substantial equivalency of engineering degree programs with this multinational agreement. Graduates of accredited programs are equally recognized as has to achieve the academic requirements for admission to carry out.

engineering practices in any constituent country as advises by the significant essentials of the agreement. Organizations of each member country have recognized graduates from accredited programs are ready to perform engineering at the entry level. This culture transformation without doubt contributed very significantly towards the growth of quality assurance for graduate outcomes (Basri and et.all, 2008).

2.2 Outcome Based Education

Assessment with OBE is done by respective institution to reflect the process of Continuous Quality Improvement (CQI). Outcomes of OBE are in the grounding of graduates for professional practice. Furthermore, rather than focusing on the process in achieving the results it requires documented proofs on how the programmed conveys and builds up although this may be equally significant. The demonstration of that the graduate has achieved the require outcome is a vital part on the approach of the OBE programs. The outcomes are categorized into three main domains, including the Cognitive, Psychomotor and Affective domains.

2.2.1 Cognitive

The cognitive domains describe understanding and the growth of student's mental skills. These domains consist of six levels beginning from the easiest behavior (understanding) to the most intricate (evaluation). The six (6) levels of cognitive domain are as follows (Madya and Jantan, 2009):

i. **Knowledge**: identify or recall data.

   Examples of applied terms: define, describe, recall, recognize, remember, knows, what, names, when, list, selects.

ii. **Comprehension**: express that the student has adequate knowledge to organize and sort out the matter mentally.

Examples of applied terms: converts, compare, contrast, rephrase, distinguishes, infers, interprets, paraphrases, summarizes.

iii. **Application**: Obtaining an answer from a question that can relate to knowledge gain beforehand.

Examples of applied terms: apply, use, employ, separates, classify, analyzes, outlines, choose, solve how many.

iv. **Analysis**: Advance order of questions that involves students to have in dept critical thinking. The students are asked in cognitive practices for analytic questions, for example: identify the cause of a specific events, analyze and consider existing data to reach a conclusion or generalization,

Examples of applied terms: determine, support, identify causes, evidence, illustrate conclusions, analyze, why.

v. **Synthesis**: Advance order of questions that involves students to have new and creative thoughts. In questions, students are asked to make predictions, generate new communications, and solve problems. This type of question accepts the different variety of creative answers.

Examples of applied terms: predict, combines, write, designs, develop, synthesize, and construct, tells, reconstructs, revises, modifies.

vi. **Evaluation**: Advance order of questions that does not have only an accurate answer. It involves students to a critic the value in an idea or the key to a

problem. The student may in addition be asked to present an outlook on the topic. In synthesis, questions may be used to initiate a class dialogue, as this type of question allows students to realize there are a variety of opinions on the subject.

Examples of applied terms: compares, interprets, justifies, summarizes, concludes, and criticizes.

2.2.2 Psychomotor

The psychomotor domain incorporates the use of motor-skill area, coordination, and physical movement. Improvement of these skills involves practice and is judge in terms of technique's implementation, range, process, speed or accuracy (Simpson, 1972). The skills evaluate the range from perception (lowest skill) until to origination (highest skill). The seven (7) levels of psychomotor domain are as follows (Madya and Jantan, 2009):

i. **Perception**: The skills to detect signals from stimulus to guide motor activity. This ranges from conversion, cue selection, and sensory stimulation.

Examples of applied terms: chooses, describes, detects

ii. **Set**: The emotional, physical, and mental sets to act with eagerness. These three sets are characters that predetermine a person's reaction to divers' circumstances.

Examples of applied terms: begin, displays, explains, moves, proceeds, react.

iii. **Guided Response**: The early phases in becoming skilled at an intricate skill that involves imitation and test and fault couple with carry out led to achieve better presentation.

Examples of applied terms: copies, traces, follows, react, responses.

iv. **Mechanism**: The immediate phase in educating physical ability with the outcomes that is habitual with a medium intensity of assurance and expertise.

Examples of applied terms: assembles, constructs, dismantle, displays, grinds, and organizes.

v. **Complex Over Response**: The adept performance of motor acts that engage intricate movements that are achievable with a least amount of wasted attempt and a high level of assurance to be successful.

Examples of applied terms: heats, manipulates, measures, sketches.

Note: the terms maybe similar as mechanisms but will have adjectives or adverbs that point out that the performance is faster, enhanced, precise, etc.

vi. **Adaptation**: movements that can be adapted in support of special environments.

Examples of applied terms: adapts, alters, changes, rearranges.

vii. **Origination**: fresh movements can be formed for particular situations. Learning outcomes focuses on vastly developed skills with creativity.

Examples of applied terms: arranges, build, combines, originates.

2.2.3 Affective

The affective domain involves the behavior in which we deal with things emotionally, for instance, attitudes, appreciation, motivations, values, enthusiasms, and feelings. (Krathwohl, Bloom, Masia, 1973). The five (5) levels of affective domains are as follows (Madya and Jantan, 2009):

i. **Receiving**: The concern is that the student's concentration is focused. The student passively attends to certain phenomena or stimulus projected outcomes include the student's responsiveness, listens attentively, listens considerately, shows understanding to social problems.

   Examples of applied terms: sits erect, points to, chooses, identifies, asks, locates, etc.

ii. **Responding**: The student actively participates, not only attends to the stimulus but responds in some manner. Learning outcomes focuses on take pleasure in aiding others, follows rules, participates in class conversation, finishes homework, show interest in subject, etc.

   Examples of applied terms: writes, discusses, performs, complies, answers, presents, assists helps, practices, reads, etc.

iii. **Valuing**: The significance of the student attaches to a certain entity, occurrence, or behavior. It varies from acceptance to commitment for example such as assuming accountability for the functioning of a crowd; attitudes and gratitude. Learning outcomes focuses on demonstration confidence in democratic development, values the part of science in everyday life, shows apprehension for other's wellbeing, etc.

Examples of applied terms: shares, explains, proposes, initiates, differentiates, justifies, and etc.

iv. **Organization**: developing a philosophy of life, conveying together different significances, and starting to put up an internally reliable value system comparing, resolving disagreements amid them, relating and synthesizing values. Learning outcomes focuses such as recognize the necessity for equilibrium involving liberty as well as responsibility in a democracy, and admits responsibility for personal actions, etc.

Examples of applied terms: compares, arranges, integrates, combines, generalizes, modifies, organizes synthesis, etc.

v. **Characterization by a Value or Value Complex**: At this stage, the person has detained a value coordination that has organized his manners for an extended time that a characteristic has been produced. Actions are predictable, consistent and pervasive. Learning outcomes focuses are concerned with social, personal, and emotional amendment: maintains fine health practice, cooperates in group activities, displays self confidence in working independently, etc.

Examples of applied terms: acts, displays, listens, perform.

2.3 Technique for String Matching

This section within the document describes three different string matching techniques in detail, which are suitable to be adapted in this project. The techniques are latent semantic analysis for text based, Boyer Moore string search algorithm and brute force string searching.

2.3.1 Latent Semantic Analysis for Text Based

LSA is a technique in natural language processing; a statistical model of word handling that allows assessments of semantic resemblance between pieces of textual information. It was developed to progress towards the effectiveness of information retrieval by performing retrieval based on the derived "semantic" contents of words in a query hence avoiding some problems of synonymy.

To examine the text, LSA first generates a matrix of occurrences of each word in each document (sentences or paragraphs), then uses singular value decomposition (SVD). SVD a technique strongly linked to eigenvector decomposition and factor analysis. It is a mathematical technique; a matrix containing word counts per paragraph is constructed from a large portion of text. The function is to decrease the number of columns while preserving the similarity structure among rows. Words are then evaluated by taking the cosine of the angle between the two vectors formed by any two rows. Value close to 1 corresponds to very alike terms while values close to 0 represent very different terms. (Dumais, 2005).

LSA does need a great quantity of text in order to carry out the SVD analysis. Typically, 200 contexts would be the minimum needed. LSA required a vast amount of processing power, plus the majority’s analysis is currently performed on UNIX workstations. Furthermore, LSA cannot capture polysemy (words of multiple meanings). However, this method is automatic and swift, allowing fast measurements of the semantic similarity between pieces of textual information (Foltz, 1996).

2.3.2 Boyer Moore String Search Algorithm

Robert S. Boyer and J Strother Moore developed this BM algorithm in 1977 and have set the standard for practical string search literature as a competent string searching algorithm. It is based on the unconventional idea to start comparing characters at the end of the pattern rather than the beginning. (Wirth, 1985)

The algorithm without the string searched in the text it preprocesses the string to search for the pattern. It is very suitable used for applications in which the text does not continue through numerous searchers. The BM algorithm triggered two heuristics, which are bad character and good suffix on a mismatch to trim down the amount of comparisons. The heuristics are independent, and they are used simultaneously where the utmost shift computed by the two heuristics is reflected on after each effort during the searching phase (Boyer and Moore, 1977).

Lastly, the algorithm runs quicker as the pattern length increases. It exploits information collected through the preprocess step to pass over parts to the text, hence giving in a lesser constant issue than several other string algorithms.

2.3.3 Brute Force String Searching

Brute force string searching is known as the most basic string matching, where as it means just to verify every single character from the text to match against the pattern. The applications of brute force string matching are rather straightforward. We must search for a match between the first characters of the pattern with the first character within the text. If they don't match we proceed to the second character and onwards. The process is complete when the exact match of the text and pattern is found.

The advantages of using brute force string matching is that it is a straightforward approach, and the most suitable approach for sequential search, if compared to others and are able to be applied at a variety of problems. The brute force approach yields reasonable algorithms of some value with no limitations on sizes (Knuth, 1973).

2.4 Existing System

This section of the project shows the studies and reviews of the current existing system related research. It also shows how to apply the specific techniques into the existing system.

2.4.1 Characterizing the Quality of Essays

This trial utilized information on how semantically alike when a user wrote in an essay and to what the user had understood. Using latent semantic analysis from the given essays the system returns a level controlled list of the matching sentences in the original texts. (Foltz and Martin, 2004) It used this cosine measure as a characterization of the quality of the essay. Its scoring system is based on the similarity of the sentences compared to the original text. This comes close to provide for an assessment of retention of information. It reflects the scale to which; In their essays, user can remember and use the information from the texts.

2.4.2 Question Generation

Automatically generating questions is an important task in many different contexts, including dialogue system, intelligent tutoring systems, automated assessment and search interfaces. Question is used to express informational needs. As computer systems become more advance and are expected to be more adaptive and autonomous, their informational needs grow, and being equipped with the ability to ask question has clear advantages. In recent years, a significant body of work has begun accumulating on question generation. Hence a competitive question generation and answering project used in undergraduate natural language processing courses is presented. This semester-long project challenges teams of three of four students to use available components to construct systems that ask and answer questions about an arbitrary Wikipedia article.

Most of the student developed systems had the following characteristics in common, multiple levels of preprocessing of the input articles using existing, freely available tools for various tasks such as sentence's boundary detection, tokenization, parsing, named entity recognition, and co reference resolution; use of existing knowledge sources, in particular, the WordNet lexical database to replace words with synonyms, antonyms and other related words; hand written transformation rules for generating questions from source text. These rules operated on either list of words, dependency parse trees, or phrase-structure parse trees; coverage of only a few of the question types and linguistic phenomena related to question. Most groups chose to focus on particular cases in which questions were easy to generate.

2.5 Development Environment

This sub section defines the set of software-development life cycle phases, software and hardware tools used to create the system.

2.5.1 Development Methodology

The waterfall model is the model of a software-development life cycle. It starts with instituting system requirements and software requirements and continues with architectural design, detailed design, coding, testing, and maintenance. This model gives emphasis to preparation in early stages and ensures design errors before they develop. Furthermore, the thorough document and planning make it work well for projects in which quality control is a main concern. The waterfall model serves as a standard guide to other software life cycle models.

1. **System requirements**: Set up the prerequisites for creating the system, together with the hardware components, software resources, etc.

2. **Software requirements**: It's the most crucial phase for the whole project. A detail list of software functionality and limitation is presented. It finds out which system requirements the software effects. A top-level analysis and design are documented as this phase.

3. **Architectural design**: Design the framework of the systems which includes the interaction between major components.
4. **Detailed design**: Following the software components given produces a specification for how each module is developed.

5. **Coding**: Develop the comprehensive design specification

6. **Testing**: The system is tested to check if it meets the functional and performance requirements from the customer. A test report is generated, which contains the test results.

7. **Maintenance**: After the software releases with the customer feedbacks, to identify the problems and perform enhancement requests if needed.

2.5.2 Development Technique

JavaScript is an object scripting language is used in web pages to code functions, which are embedded in HTML pages. The code can run locally in a client browser and respond to their actions quickly hence making an application extra receptive. Furthermore, JavaScript improves user experience by creating intermediary effects such as sliding and fading animation. The JavaScript web server displays host objects representing HTTP request and response objects. The program interrogates and operates to dynamically generate web pages. It is dependent upon the computer in which the web pages are being displayed.

PHP: Hypertext Preprocessor is a server scripting language intended for web development to create dynamic web pages. It is to be embedded into an HTML source document other than calling an outside file. The code is interpreted by a web server with a PHP processor component which produces the out coming web page. It has been advanced to comprise a command line interface capability. It can also be used in standalone graphical applications. PHP can be deployed as a standalone shell and also on the majority web servers.

2.5.3 Development Hardware

This section covers the hardware requirements needed to develop and run the system. The hardware requirements are categories into two environments, which are development and end user requirements. Both environments require low level hardware requirements hence in order to develop or run the system can be done from using a general laptop which has installed it's necessary application.

i. Development

Table 2.1: Development Requirements

	Windows requirements	**Mac requirements**	**Linux requirements**
Operating system	Windows XP SP 2+ Windows Vista Windows 7 Windows 8	Mac OS X 10.6 or later	Ubuntu 10.04+ Debian 6+ OpenSuSE 11.3+ Fedora Linux 14
Processor	Intel Pentium 4 or later	Intel	Intel Pentium 3 / Athlon 64 or later
Free disk space	1 GB		
RAM	1 GB		

ii. End-user

Table 2.2: End User Requirements

	Windows requirements	**Mac requirements**	**Linux requirements**
Operating system	Windows XP SP 2+ Windows Vista Windows 7 Windows 8	Mac OS X 10.6 or later	Ubuntu 10.04+ Debian 6+ OpenSuSE 11.3+ Fedora Linux 14
Processor	Intel Pentium 4 or later	Intel	Intel Pentium 3 / Athlon 64 or later
Free disk space	100 MB		
RAM	128 MB		

2.5.4 Development Software

The usage software development tools will be discussed and displayed in table in this section. There require three different types of software to develop this system. The first is Notepad ++. It is distributed as free software as a robust editor for a variety of programming scripting languages, support syntax highlighting and allows working with multiple open files. XAMPP the secondary software an open-source cross platform web server solution package and consist of MySQL that enables a user to create database in MySQL among the local host.

Table 2.3: Software Requirements

Software	Description
Notepad ++	As a medium to write and compiler or editor
XAMPP Control	To allow developer runs the system in the local host during execution and link local host's database. Apache is used as web server.
MySQL	It is database which stored information from the developed system can be updated and inserted.
Filezilla	It is a way to upload and download the system.

2.6 Conclusion

Currently is it found out that there does not exist any system that helps lecturers evaluate or generate the final exam template, hence the study and research gathered from this chapter require to produce necessary information to further develop the propose system. The understanding of each various level of domain is to help seek a suitable technique to help the system obtain the best output. Furthermore, the study of related existing system will provide a better user experience in terms of functional and non functional requirements. Lastly, the development is specified here in order to prepare the required environment to conduct the next phase of the project.

CHAPTER 3

METHODOLOGY

3.1 Introduction

In Chapter 3 methodology, it presents all the necessary information. A reader will need to reproduce the study and to understand all the steps were taken to ensure the reliability and validity for the study. The chapter outlines the details about the research undertaken to address the questions posed in Chapter 1 Introduction. It explores the research question in more depth, and discusses what methods are the most appropriate, given the aims and nature of the research.

Furthermore, this chapter briefly discusses about the structure, planning and control for process of developing the final exam template generator based on teaching and learning taxonomy using string matching. Chapter 3 includes the phases of the software-development life cycle used in this project, the view from various high level diagrams, and detail explanation of the technique chosen to be used for this project.

3.2 Development Methodology

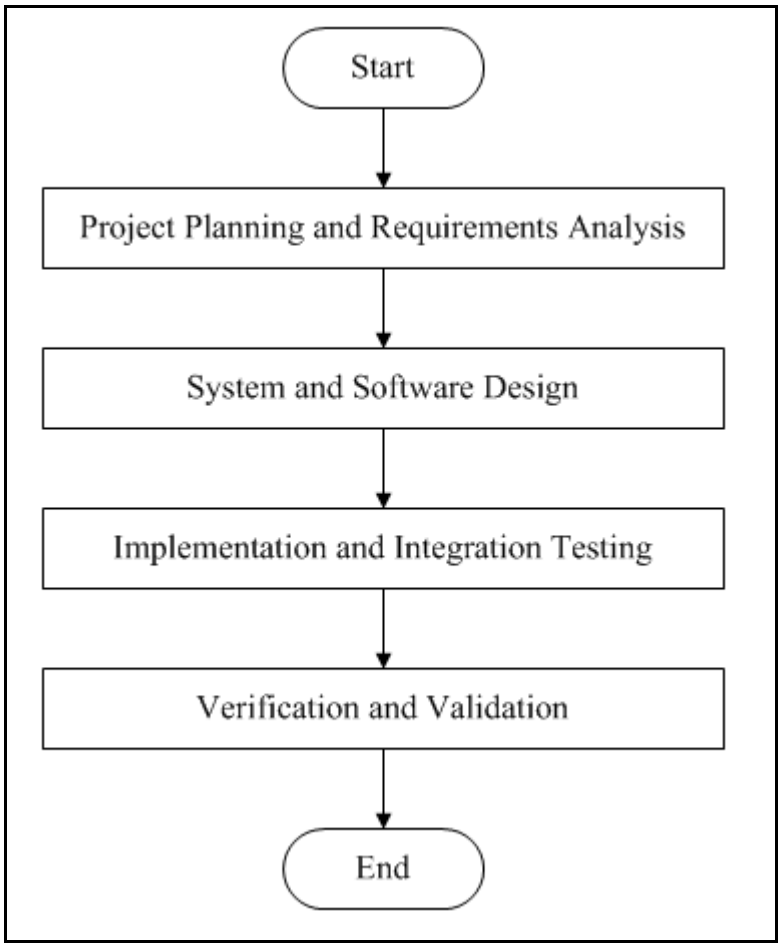


Figure 3.1: Waterfall Model

The software-development life cycle methodology that is chosen for this project is the waterfall model. It has been judged as the classic approach to software methodology, this model portrays a development method that is linear and sequential. Once a phase of development is completed, it proceeds to the next phase and there is no turning back like a waterfall.

Advantages of the waterfall model are that is that documentation takes places at the end of each phase giving high visibility and quality to the project. Goals can be set for the completion of each phase, and evaluation can be done from time to time, to check if the

project. It is going as per mile stones. There is clear partition of work and control over the model, making it easier to set schedule for the tasks to be completed within a specified time frame. furthermore, processes of this model are carried out in linear approach; the cost of resources required for it is minimal. The model does not have any overlapping phases hence does not have to go through different iterative steps.

Other than the numerous advantages stated above, the reason for choosing the waterfall model methodology is that the fact of this project is handled by a single person. The emphasis on requirements and design is crucial and suitable due to the limitation of time from the project schedule.

The phases of the waterfall model are shown in the following:

i. **Project Planning and Requirement Analysis:** This phase will depict all the possible requirements on the system to be developed. In this phase project scope, objectives and problem statement is defined. The requirements are gathered from the study of available technique and existing system from available resources. An interview is conducted for an insight of current university environment and can be included within the system. Finally, the results are finalized and documented in Chapter 1 introduction and Chapter 2 literature review. Refer to Appendix A for Gantt chart and Appendix B for Question of Interview.

ii. **System and Software Design:** A system design is set up base on the requirement's specification given from the previous phase. The system design facilitates by providing hardware and system requirements. Various high level diagrams are presented to help provide a better view and understanding about the system, for example, context, data flow diagram, system architecture and etc.

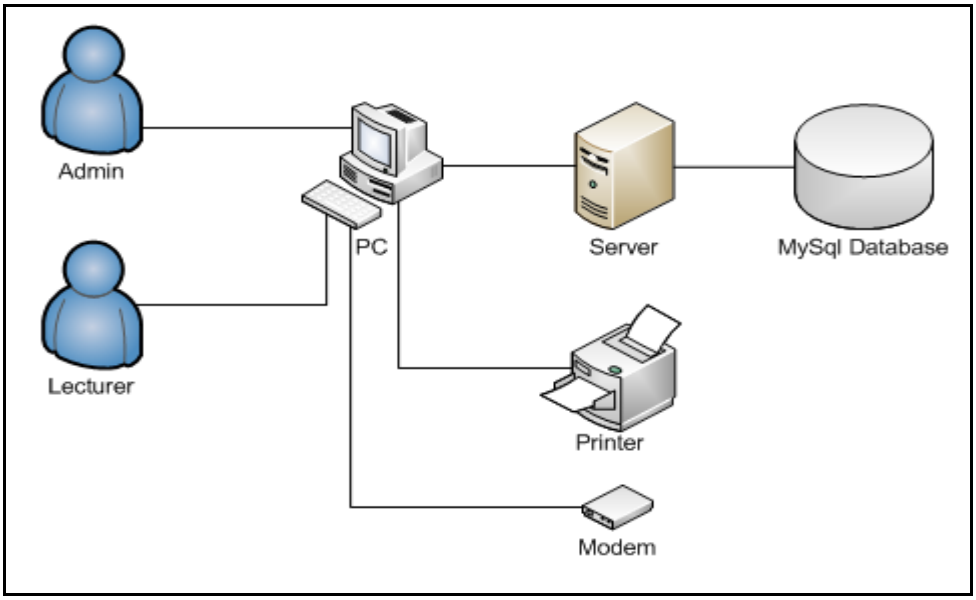


Figure 3.2: System Architecture

The figure 3.2 it shows that this system general requirement to operate the system. This system is a web application which requires usage of server with a database to run and save all the important data and documents. From the previous chapter 2, the end user will need a computer or laptop with the minimum hardware requirements stated with suitable web browsers installed to run smoothly. The printer is provided as an accessory used to print out the output from the system while the modem is to provide access to Internet usage.

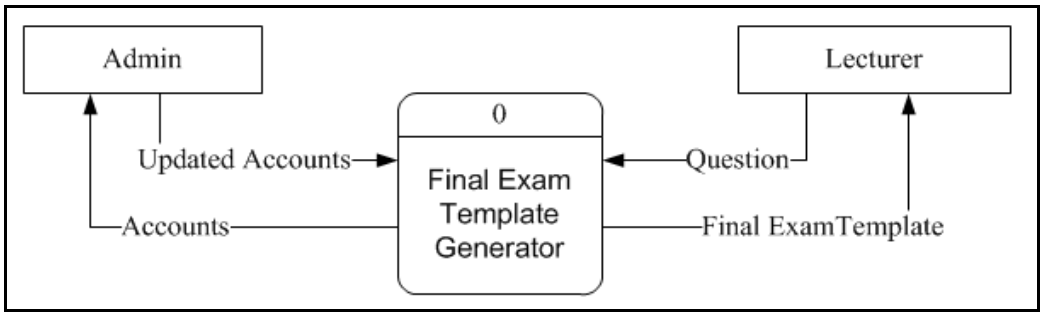


Figure 3.3: Context Diagram

Following on the figure 3.3 shows the boundaries within the system with clearly define external entity that will communicate with the system. There are 2 external entities that act as end users for this system those are admin and lecturer. The roles of admin manage all the accounts of the system, while the lecturers are the main users who whom they key in their data, for example, a question to be evaluated and finally able to generate a final exam template by the system.

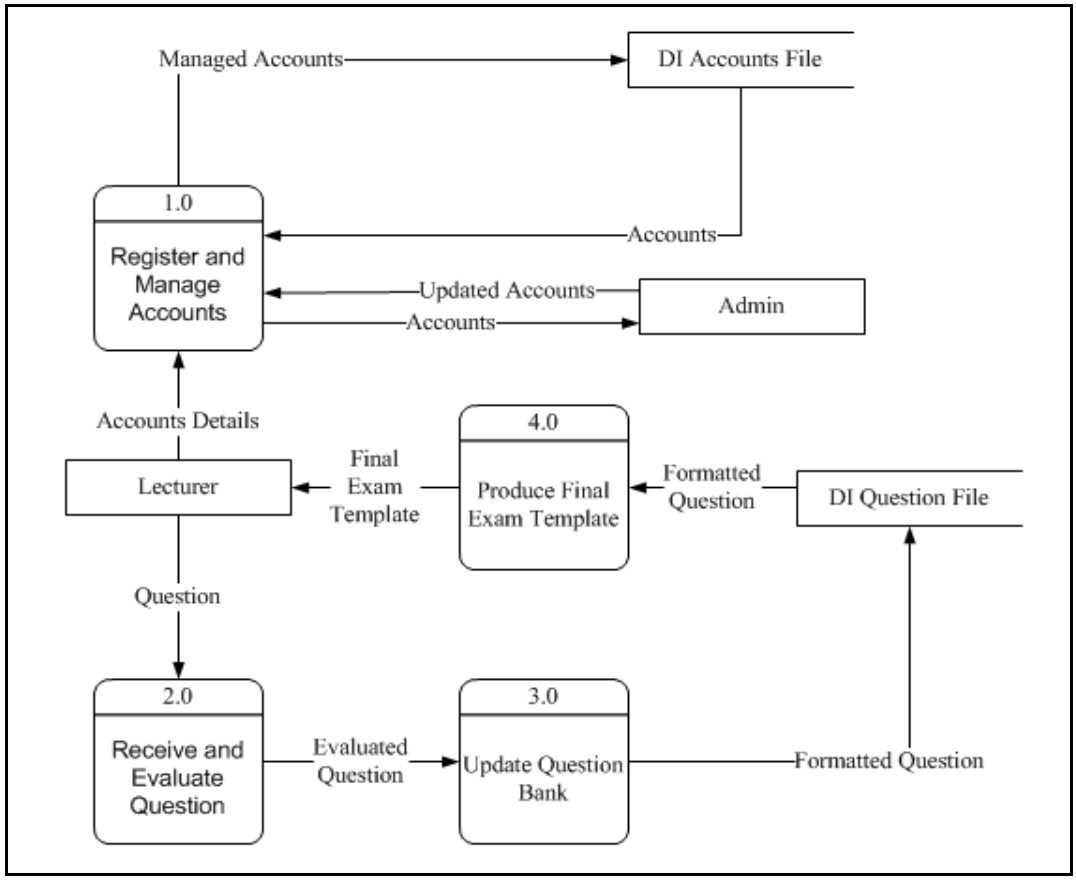


Figure 3.4: DFD level 0

Figure 3.4 is a level 0 data flow diagram which shows the primary individual processes within the system. The main function within the system is depicted from the diagram that is to receive the input of questions from the lecturers to be evaluated as output. Evaluated question can also be saved, edited and deleted in the database as shown on the process 3.0. The additional function within this

system is to gather those evaluated questions in order to generate the final exam template. Other processes within the system are the registration and management of the lecturers of this system.

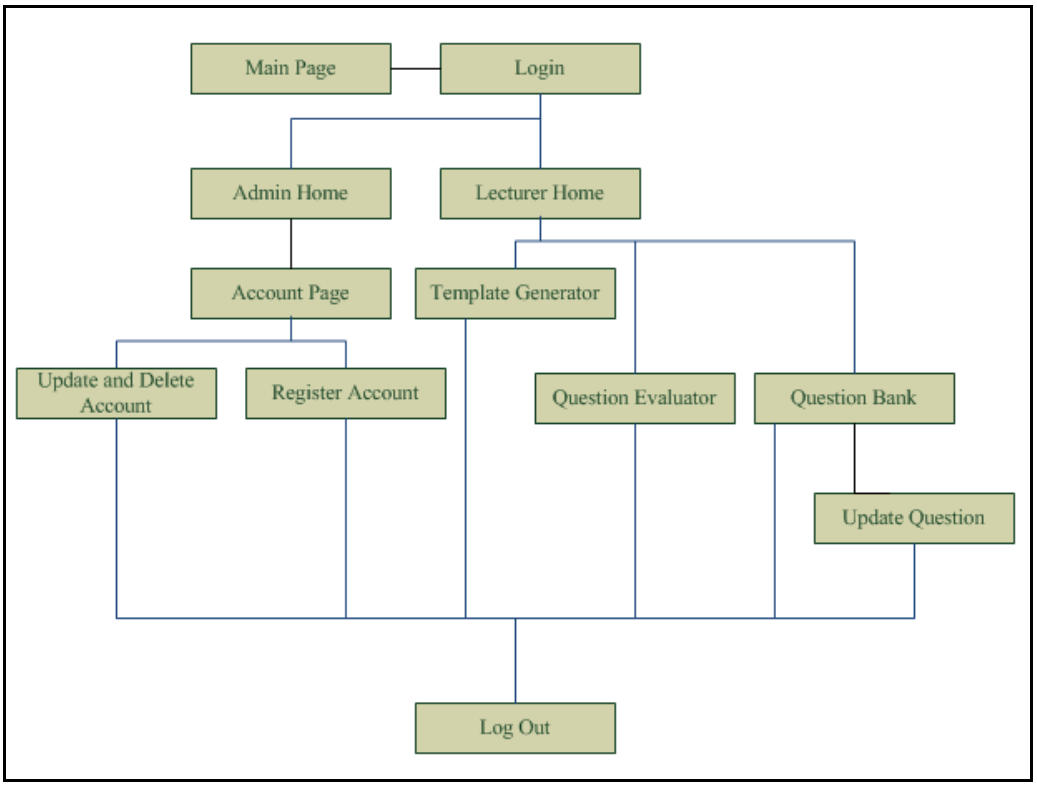


Figure 3.5: Dialog Diagram

Figure 3.5 shows the flow of interaction or interface from the system. This system is divided into two ways that are the admin side and lecturer side. Both end users are able to view the main page and from the login, page is where the system will determine the roles of the users and redirect them to their specific home page after login. The logout process will send the users back to the main page.

The management within the system is managed by the admin where they are able to view the accounts in the account page. Furthermore, the accounts' page allows them to register new accounts or update and delete the current registered accounts.

The lecturers are able to evaluate their question in the question evaluator page. Following on the question can be uploaded to the database, and to be updated later on. Finally generating final exam template is made available to the template generator page.

iii. **Implementation and Integration Testing:** from the conversion of design into a machine-readable form. The software modules are now further divided into units. Testing these units can be done separately and is done by the programmer itself, to ensure that there are no bugs. Following on the units of the software are then integrated together, and a system is built. The system is to be check for function and performance requirements from the customer. Testing is done, step by step as classified in the test plan, to ensure input produces genuine results, which consent with the required results. The system is divided to four modules as shown in the following tables.

Table 3.1: Module 1

Module 1	Register and Manage Accounts
Function	To insert, update, or delete accounts
Description	A set of programs that allow user to store, modify and extract information from database.
Primary User	Admin

Table 3.2: Module 2

Module 2	Receive and Evaluate Question
Function	To receive and evaluate the question
Description	A set of programs to evaluate whether the input with brute force string matching based on teaching and learning taxonomy
Primary User	Lecturers

Table 3.3: Module 3

Module 3	Update Question Bank
Function	To insert and update question
Description	A set of programs that allow user to store, modify and extract information from database.
Primary User	Lecturers

Table 3.4: Module 4

Module 4	Produce Final Exam Template
Function	To generate an final exam template
Description	A set of programs that uses evaluated question to generate an final exam template for lecturers
Primary User	Lecturers

iv. **Verification and Validation:** This is the most vital phase of the project. It is a series of procedures to check whether the system assembled correctly to fulfill the requirements and specification of its intended purpose. The samples for testing involving lecturers are conducted to check its functionality.

Operation and Maintenance: the completed tested software is delivered to the customer. Customer feed backs are taken and any changes, if required are made during this phase. This phase ends as the software is retired. However, this phase is not implemented due to the limitation of project schedule.

3.3 Brute Force String Matching

The reason for choosing this algorithm is due to effectiveness against short text with the short patterns. The advantage of this algorithm is that it does not require any preprocessing of the text hence is it effective at none repeatable matching. Extra feature of this algorithm is that without preprocessing of the text, it will reduce the space usage of the system.

The following figure 3.6 shows an string matching technique algorithm used for the system. Brute force string matching in general means to test out every particular character from the text to match against the pattern.

```
Algorithm BruteForceStringMatch(T[0...n-1], P[0...m-1])
    for i ← 0 to n-m do
        j ← 0
        while j < m and P[j] = T[i+j] do
            j++
        if j = m then return i
    return -1
```

Figure 3.6: Brute Force String Matching Algorithm

Based on the algorithm as shown in figure 3.6, the description of the parameter is as follows:

P = pattern

T = text

m = length of pattern

n = length of text

i = the order of character in the text

j = the order of character in the pattern

The principles behind brute force string matching are straight forward, in order to give a demonstration with the explanation of the algorithm an example is given in the text "HELLO WORLD!" and pattern "O WO".

The followings steps are taken as shown below:

Step 1: From the figure 3.7 will demonstrate the first step as we must check for a match between the first character of the pattern and the first character within the text.

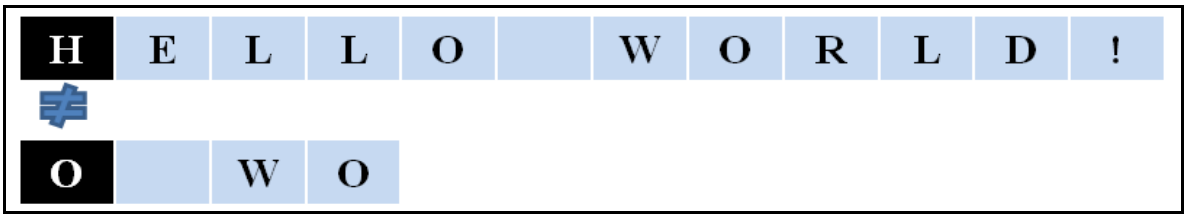


Figure 3.7: Demo 1

Step 2: If the characters don't match it will move forward to the second character within the text, and it will compare the first character of the pattern with the next character within the text as shown in the figure 3.8

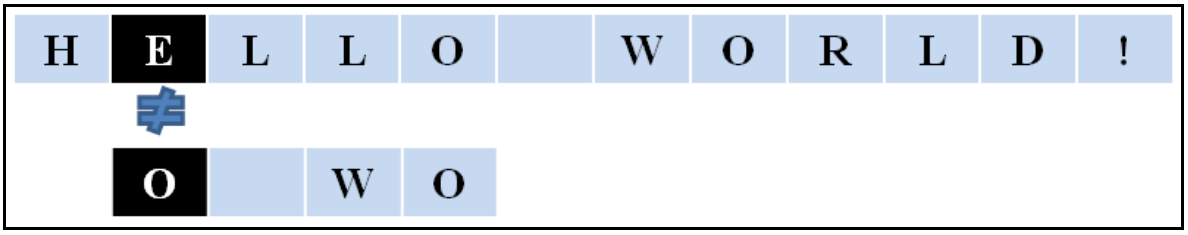


Figure 3.8: Demo 2

Step 3: Following on if they don't match again the last phase will repeat Step 2 till a match is found, or till it reaches the end to the test. However, from the figure 3.9 shows if a sample of a character from the text matches against the first character of the pattern we move forward to the next character of the pattern and the following characters.

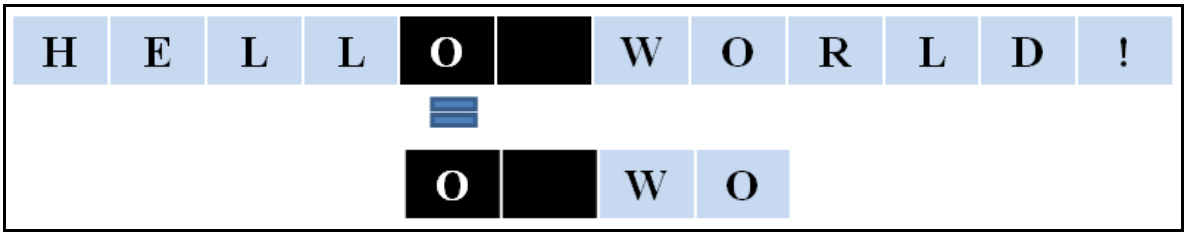


Figure 3.9: Demo 3

Step 4: Finally the last step just because it detects a match between the first characters from the pattern with some character of the text, it does not indicate the pattern shows in the text. It will shift forward to see whether the full pattern is contained into the text as shown in the figure 3.10.

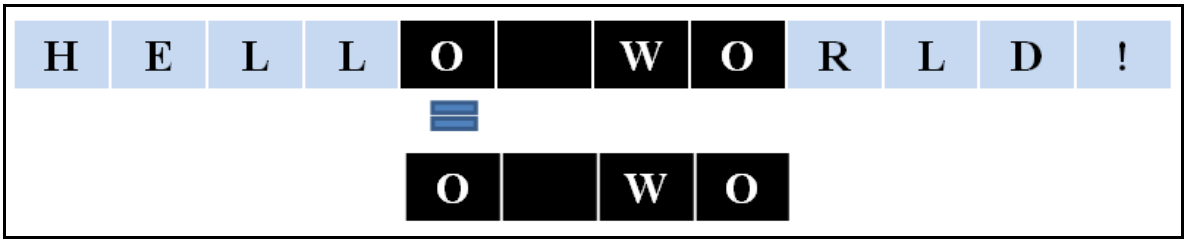


Figure 3.10: Demo 4

3.4 Conclusion

The adaptation and understanding of waterfall methodology of this project have given understanding of each phase before moving on to the next phase. Furthermore, getting the requirements and design right of the start first has improved quality standards within the project. This methodology will aid knowledge transfer to the next phase which is vital to ensure minimal wastage of effort and time and reduce the risk of schedule slippage. The data flow diagram from this chapter has provided great understanding in terms of the framework from the highest level. Finally, with the detail demonstration of the brute force string matching technique that provides a clear understanding of the system main operating function.

CHAPTER 4

DESIGN AND IMPLEMENTATION

4.1 Introduction

This chapter will organize as follows. We will start with discussing a propose system or prototype flow. Then we will illustrate how we employed brute force algorithm for keyword matching to suggest the level of questions with presenting the algorithm of brute force itself. Then we will continue our discussing on explaining how we design and implementing a question bank. Question bank will act as a database to keep questions with their information before selecting by a lecturer to prepare final exam questions. In the third part, we will illustrate how we design and implement the PDF generator to generate the questions and schema based on a verified template in PDF format. In general, development stage is divided into three main phases, which are; (1) Development of string matching algorithm, (2) Development of the question bank as a database of the prototype and (3) Development of final exam and schema template as an output of the propose prototype. However, the main concern for the study is to develop string matching algorithm based on the brute force algorithms so that we can automatically suggest the level of each question based on OBE level.

4.2 The Proposes System Flow

This section describes the proposes system flow to illustrate how the algorithm is embedded into the prototype in order to allow it to determine the level of final exam question based on OBE guideline. Figure 4.1 show the general system flowchart indicates main activities in the proposes prototype. Based on the system flowchart, the prototype is divided into 2 main operations, which are preparation of set of question and generating template of document.

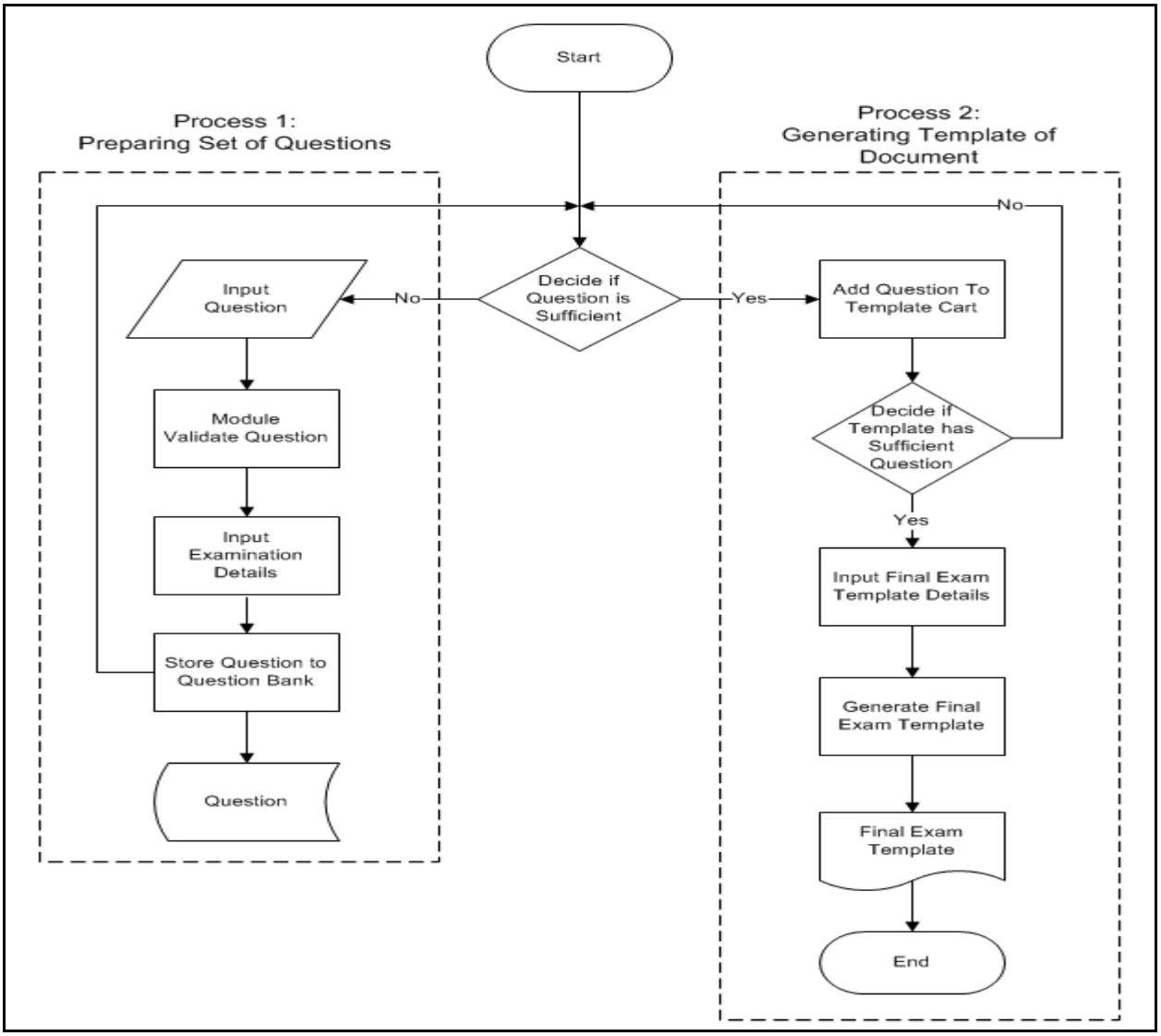


Figure 4.1: System Flow Chart

As shows in figure 4.1, process 1 is a main process for this project with the objective to automatically suggest the appropriate level of question based on OBE level. In the process 1, user or lecture will input the desire questions and the algorithm will automatically help users to suggest the OBE level by executing brute force string matching technique. How it works? User or lecture should input the questions then, will process this input with searching for appropriate words that represent components and levels from the outcome-based learning. Details of implementation will be explained in the next section.

On the other hands, process 2 concentrate on the process to generate the template of documents in PDF format. Three different documents that require in preparing the final exams are prepared. There are:

i. Final exam paper - The main document consist of number of questions.
ii. Schema for final exam paper - The document consist of marking guideline together with sample of answers.
iii. OBE report form - Form the represent summary report of questions and level of each questions.

Sample of output is available in Chapter 5 meanwhile details of implementation will be further discuss in the next section.

4.3 Brute Force String Matching Algorithms Implementation

The principle of brute force string matching is to check between the first character after the pattern with the first character within the text. Details of illustration how brute force works at chapter, section 3.3. Figure 4.2 illustrate how we employed brute force algorithm into the propose prototype. Furthermore, figure 4.3 illustrate brute force string matching algorithm's function. In this case, we have employed PHP scripting language to develop the function.

```
Algorithm BruteForceStringMatch(T[0...n-1], P[0...m-1])
    for i ← 0 to n-m do
        j ← 0
        while j < m and P[j] = T[i+j] do
            j++
        if j = m then return i
    return -1
```

Figure 4.2: Brute Force String Matching Algorithm

```
function matching_string($pattern, $subject)
{
    $n = strlen($subject);
    $m = strlen($pattern);
    for ($i = 0; $i <= $n-$m; $i++)
    {
        $j = 0;
        while ($j < $m && $subject[$i+$j] == $pattern[$j])
        {
            $j++;
        }
    if ($j == $m) return true;
    }
    return false;
}
```

Figure 4.3: Brute Force String Matching Function

Based on short syntax to the algorithm as shown in figure 4.3, the description of the parameters as follows:

$pattern = pattern

$subject = text

$m = length of pattern

$n = length of text

$i = the order of character in the text

$j = the order of character in the pattern

Table 4.1: Cognitive Inner

C01	C02	C03	C04	C05	C06
define	change	apply	analyze	arrange	appraise
describe	compare	build	attributes	collect	choose
find	convert	change	categorize	combine	compare
identify	distinguish	choose	classify	compose	conclude
label	examples	classify	compare	construct	consider
list	explain	construct	contrast	create	criticize
locate	express	demonstrate	differentiate	design	critique
memorize	extend	discover	distinguish	develop	decide
name	generalize	dramatize	experiment	formulate	deduce
recite	give	illustrate	infer	hypothesis	defend
recognize	illustrate	make	inquire	invent	evaluate
record	infer	modify	investigate	organize	judge
relate	interpret	paint	list	originate	rate
select	match	prepare	point out	plan	recommend
tell	paraphrase	produce	probe	predict	relate
	predict	show	select	produce	select
	relate	sketch	separate	revise	summarize
	restate	solve	subdivide	role play	support
	rewrite	use	survey		weight
	summarize				
	transform				

Table 4.2: Cognitive Outer

C01	C02	C03	C04	C05	C06
a play	analogy	a drama	a conclusion	alternative courses of action	a standard compared
articles	cartoon	a list	a syllogism	article	a standard established
events	cartoon	a map	a word define	book	conclusion
films	casual	a meeting	an argument	experiment	court trial
filmstrip	comparison	a painting	broken down	formulation of	evaluation
magazine	conclusion of implication based	a paper which	checked	game	group discussion
newspapers	diagram	a solution	diagram	hypothesis or question	recommendation
people	drama	another	identified	invention	self evaluation
radio	graph	diagram	part of a	play	survey
readings	model	follow an outline	propaganda	principles or standard	valuing
recordings	of like or unlike	illustration	questionnaire	set of rules	
shows	on data	one gear into	report	song	
television	outline	photograph	statement	speculate on or plan	
test	own	sculpture	survey		
	recording	shifting			
	relationship	smoothly from			
	skill				
	speech				
	statement				
	story				
	summary				
	tape				

Table 4.3: Psychomotor Level

P01	P02	P03	P04	P05	P06	P07
choose	begin	copy	assemble	assemble	adapt	arrange
describe	display	follow	calibrate	build	alter	build
detect	explain	react	construct	calibrate	change	combine
differentiate	move	reproduce	dismantle	construct	rearrange	compose
distinguish	proceed	respond	display	dismantle	reorganize	construct
identify	react	trace	fasten	display	revise	create
isolate	show		fix	fasten	vary	design
relate	state		grind	fix		initiate
select	volunteering		heat	manipulate		originate
			manipulate	measure		
			measure	mend		
			mend	mix		
			mix	organize		
			organize	sketch		
			sketch			

The patterns are considered is come from the OBE guideline, produce by UPM. For this thesis, we only consider two (2) domains, which are: (1) cognitive and (2) psychomotor. In this study, we did not consider the affective domains as it should consider a diversity of inner domains and more complex. Figure 4.4 shows the sample of OBE guideline produce by UPM.

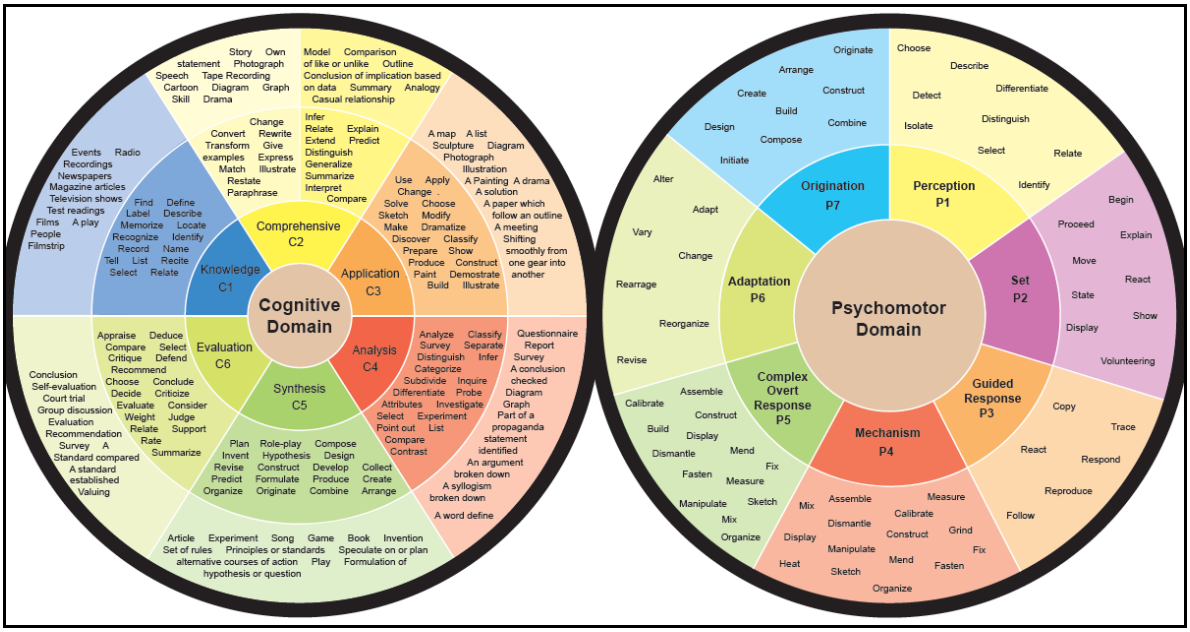


Figure 4.4: The circle that represent words in cognitive and psychomotor domain

From figure 4.4, we can see that the words that act as patterns in this case situated in two different circle; inner and outer circle. Table 4.1 - 4.2 shows the words from outer and inner circle corresponding level of cognitive and Table 4.3 for psychomotor domain.

In the first stage, the algorithm will be looking over the list of words from the question to compare with each pattern or words from the inner circle of cognitive domain. If there are same words falling into two different level, the list of patterns from the outer circle will be checked to suggest the most appropriate level.

To illustrate it, let's consider the following question and how the brute force string matching works. Example from final examination for BCS 1133, System Analysis and Design. "Sketch a dialog diagram to illustrate the sequences flow of Open Course Registration System. You may add some assumptions to complete the flow, if so, state clearly your assumptions."

Step 1: String Matching Inner Ring

The submitted question is match against the list of patterns from the inner ring of cognitive domains. Words from the question which have correspondingly matched correctly against the pattern list will be recorded with an array with its level.

Output: C2 illustrate, C3 illustrate, C3 sketch and C6 rate

Step 2: Calculate the number of different levels detected.

The array generated from step 1 will be processed in step 2, in which it will calculate the total number of different levels found from the array.

Output: Total of three different levels matched, which are C2, C3 and C6.

Step 3: String Matching Outer Ring

This step 3 will only be initiated once the processed array from step 2 has detected a minimum of 2 or more different levels of inner cognitive domain. Following on, the question will now be match against the patterns list from the outer ring of cognitive domain with corresponding levels. The corresponding level to be match is determine from the output from step 2. In which this example only C2, C3 and C6 list of patterns will be match with the question provided.

Output: C2 diagram and C3 diagram

Step 4: Display Suggestion

An interpretation as the result is given as a suggestion to the user. The result shows the list of words matched from the inner ring and the outer ring.

Output: display suggestion

However, the steps are different for psychomotor domain as it only contains a single ring hence step 2 and step 3 are not available psychomotor string matching. Thus, the algorithms at the end will suggest to the user that the question should fall into level 2 and 3 in cognitive domain and level 2, 4 and 5 for psychomotor domain. Story board of page to display is as shown in figure 4.5.

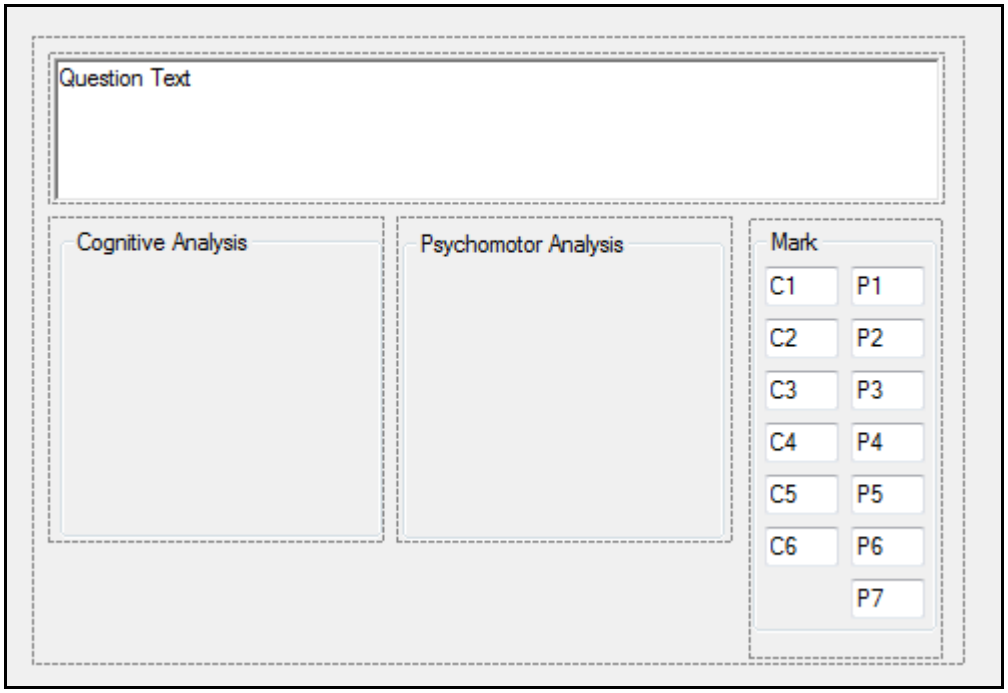


Figure 4.5: Story board of result from Brute Force String Matching

4.4 The Implementation of Question Bank

The question bank is a database on the system in which involve around all the main processes of the prototype as shown in figure 4.1. As a database, in general question bank involves numbers of different tables to store the data. Figure 4.6 shows the ER diagram to represent the tables that involved on this project.

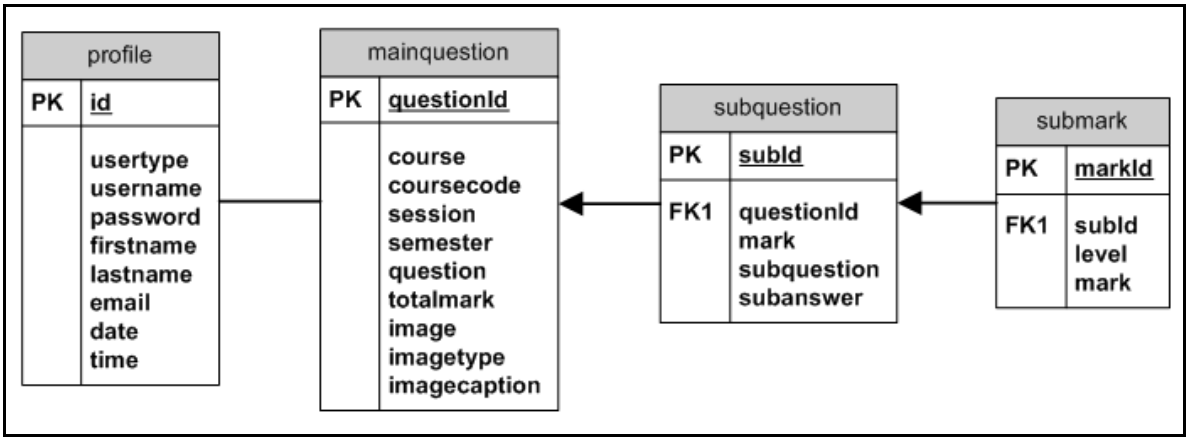


Figure 4.6: ER diagram for Final Exam Template Generator

There is a total of 4 tables involve in the system; (1) profile, (2) mainquestion, (3) subquestion and (4) submark.

The profile table stores the detail information about each user on the system. It consisted of the user type, user name, password, first name, last name, including the date and time which are recorded during registration. The data are used to validate the user during login and redirected them to their respective home page. User type data are used to determine the type of user such as admin or lecturer. After validation, the users are sent to their respective page.

Following on, the mainquestion table is a table which stores all the question generated with its course detail. The course details course name, course code, session and semester. The question and its total mark are stored also stored in this table. If the question

requires an image its details will be stored as image location, image type to determine whether it is a figure or table and the image caption.

Furthermore, subquestion table is a table which stores all the sub question of the main question. Each question is able to have up to four sub question. The foreign key questionId links the two table together. This table stores the sub question along with its answer and mark.

Lastly, submark table is the table which stores cognitive and psychomotor domain details of each sub question with its corresponding marks. The foreign key subId links the two table together.

```php
<?php

define("DATABASE_HOST","localhost");
define("DATABASE_USER","cb10036");
define("DATABASE_PASSWORD","10036");

//To establish a connection to database and save in $conn
$conn = mysql_connect(DATABASE_HOST,DATABASE_USER,DATABASE_PASSWORD);

//If connection failed then display mysql error
if(!$conn)
{
    die("Could not connect to database" );
}

//To select one particular database to be used
mysql_select_db("cb10036",$conn) or die("Could not open products database");
```

Figure 4.7: SQL command to connect database

Figure 4.7 shows the database connection codes written in PHP language. The name of the database selected to be used is called "cb10036".

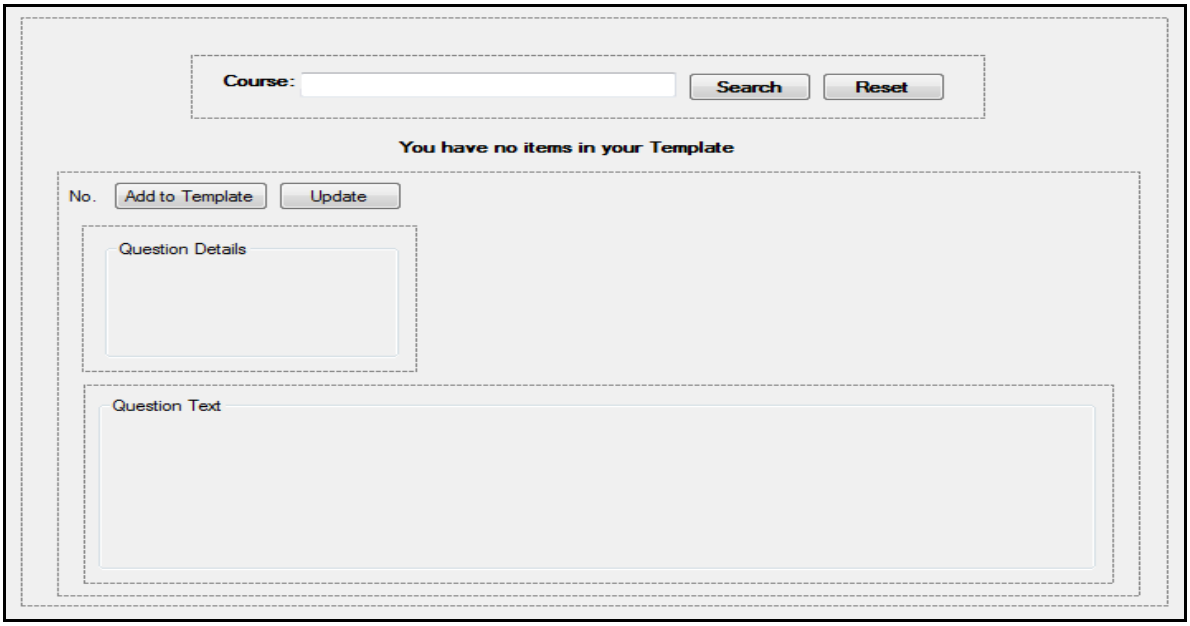


Figure 4.8: Story board Question Bank display

Figure 4.8 shows the storyboard of the interface for question bank display. The question bank is a vital part in the system in which involve around all the main processes within the system. That it is the place where are question stored are being displayed on the system.

On top of every question, are placed with two functions, which are available in the form with the button that is the "Add to Template" Button and "Update" Button to the lecturer. The "Add to Template" button is a function which is based on the cart system to allow lecturers to add the question into the cart. The cart can view later on to be review and finalize the selected question again. Whether is it needed to be included the final exam template. The "Update" Button is used by the lecturer when the lecturer would like to renew a question by updating it to a newer version which fulfils the lecturer demands. This function will retrieve the details of the question from database and direct it to a form to be updated, reanalyze and stored in again as a new question without changing the original question.

The users are allowed to do a search in order to sort the question with the search bar. It is a very useful function that provides a lecturer with the option of listing down all the question to a specific course in the question bank display. This option is to help the lecturer to select the question to be view on the system which are relevant.

The "Search" button function which uses the inputted course text into the system to set as a session "$_SESSION['searchCourse']" as shown on the figure 4.9 while the "Reset" button function is being used to unset the session to return the question bank page to default view.

```php
<?php
session_start();
include("dbase.php");
unset($_SESSION['searchCourse']);
echo "<script type='text/javascript'> window.location='questionBank.php'</script>";
?>
```

Figure 4.9: Reset Question Bank

```php
<?php
session_start();
include("dbase.php");
extract($_POST);
$_SESSION['searchCourse'] = $searchCourse;
echo "<script type='text/javascript'> window.location='questionBank.php'</script>";
?>
```

Figure 4.10: Search Question Bank

Figure 4.10 shows that when loading question bank page it will check if the session "$_SESSION['searchCourse']" has been set to display the specific course else it will display an default view. This is to help the question bank page whether to display the default view or display it with a list specified course question. Lastly, figure 4.11 shows the code that decide whether the question bank list is displayed in default view or with a search course view.

```
if(!empty($_SESSION['searchCourse'])) {
$test = $_SESSION['searchCourse'];
$query="SELECT * FROM mainquestion WHERE course='$test'";

}else{

    $query="SELECT * FROM mainquestion";
}
$result=mysql_query($query,$conn) or die("Could not execute search bar questionBank.php");
$num_rows= mysql_num_rows($result);
```

Figure 4.11: Question Bank List

4.5 Implementation of Template Generator

Obviously generating template is a main output from the purpose prototype. In this study, we considering to generate three different template in PDF format, which are:

(1) Final examination question paper.
(2) Schema for the final examination papers.
(3) OBE reports for the final examination paper.

Each template requires a different formatting and input as stated by Faculty and CAIC, UMP. The conversation of the details and question to an PDF template is due to the usage of an PHP library called as FPDF. FPDF is a free PDF generator who uses objected oriented scripting language OOPHP; however, the customization of the function within the FPDF class is required in order to generate the specific format, for example, the front page, header, footer and etc. Below we will discuss details the requirement and implementation for each required template as stated above.

(a) Final Examination question paper.

The template is a sample final exam paper. It serves as a written examination paper which consists of a cover page and a set of question that evaluates the skills or knowledge to the student who taken the test. All the questions generated in this template is taken from

the database which can be viewed from the question bank page. Figure 4.12 shows the story board for exam requirements details required for the final exam front page.

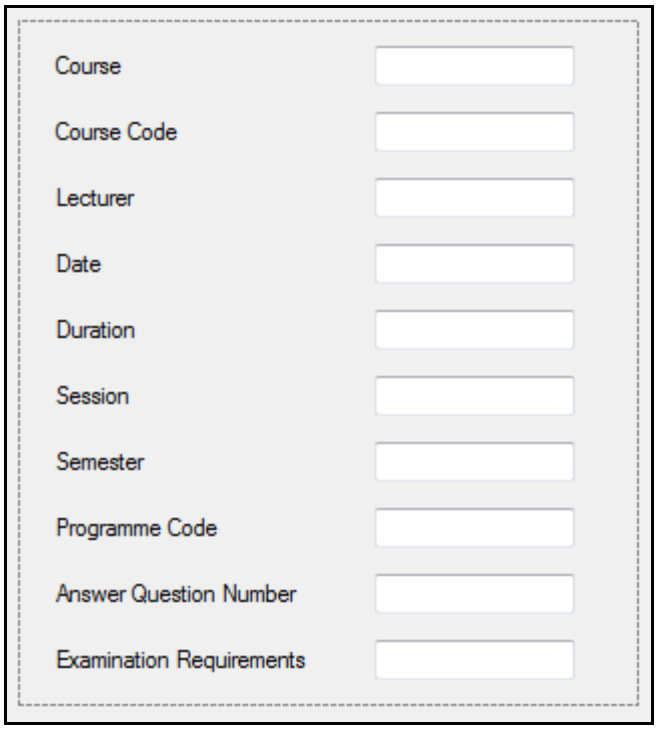


Figure 4.12: Story board of Exam Requirements

(b) Schema for final examination paper.

This template is served as a marking guide for the lecturer. The different between this template and the final examination question paper is the addition of answer's schema to the question. This answer is an act as an ideal answer which the student is expected to write. The lecturer will mark the students paper according to this template. Figure 4.13 shows the story board for the user to submit the question and schema which later will be validated and displayed in the question bank display.

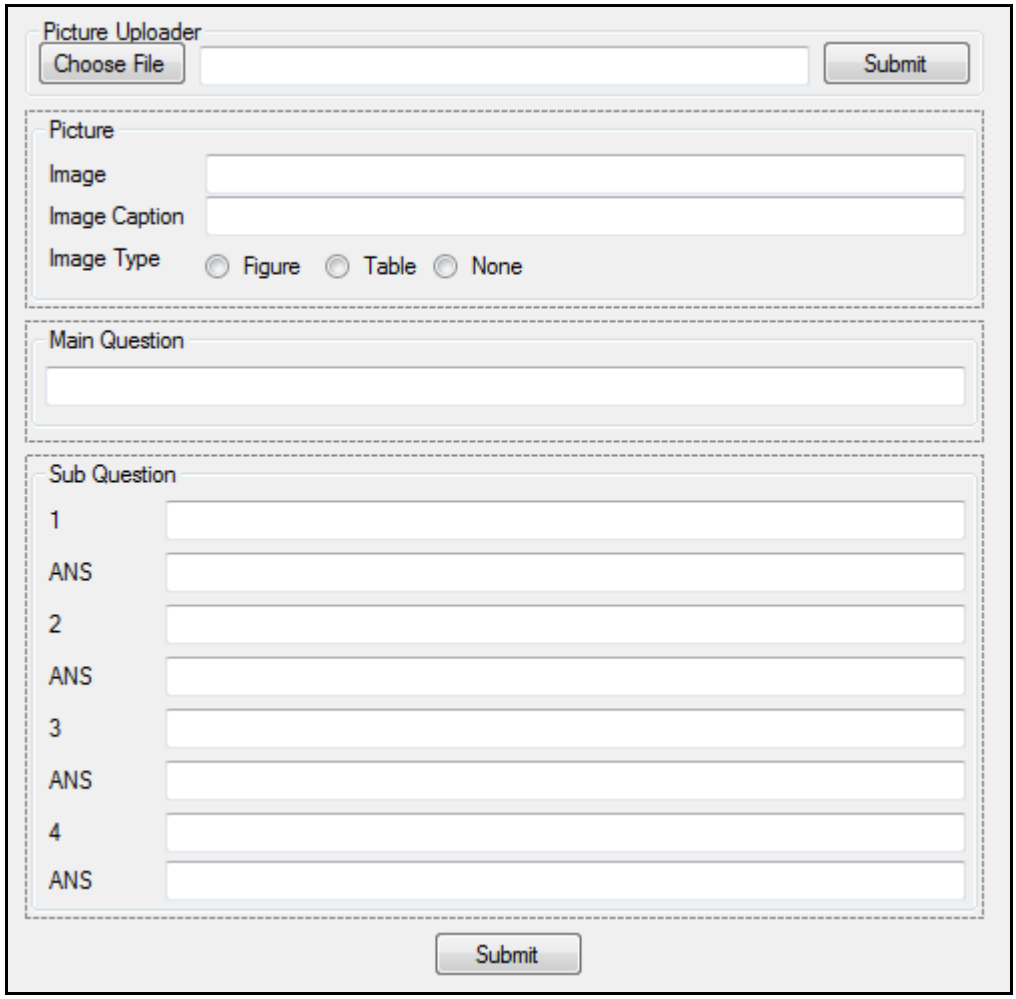


Figure 4.13: Story board for Question Submission

(c) OBE report for the final examination paper.

Each final exam question must consist at least 60% cognitive has been decided by FSKKP. In order to help user to monitor the total mark for each level this template is provided. This template consists of domain mark distribution and its report. The sample of interface is shown in figure 4.14 and figure 4.15.

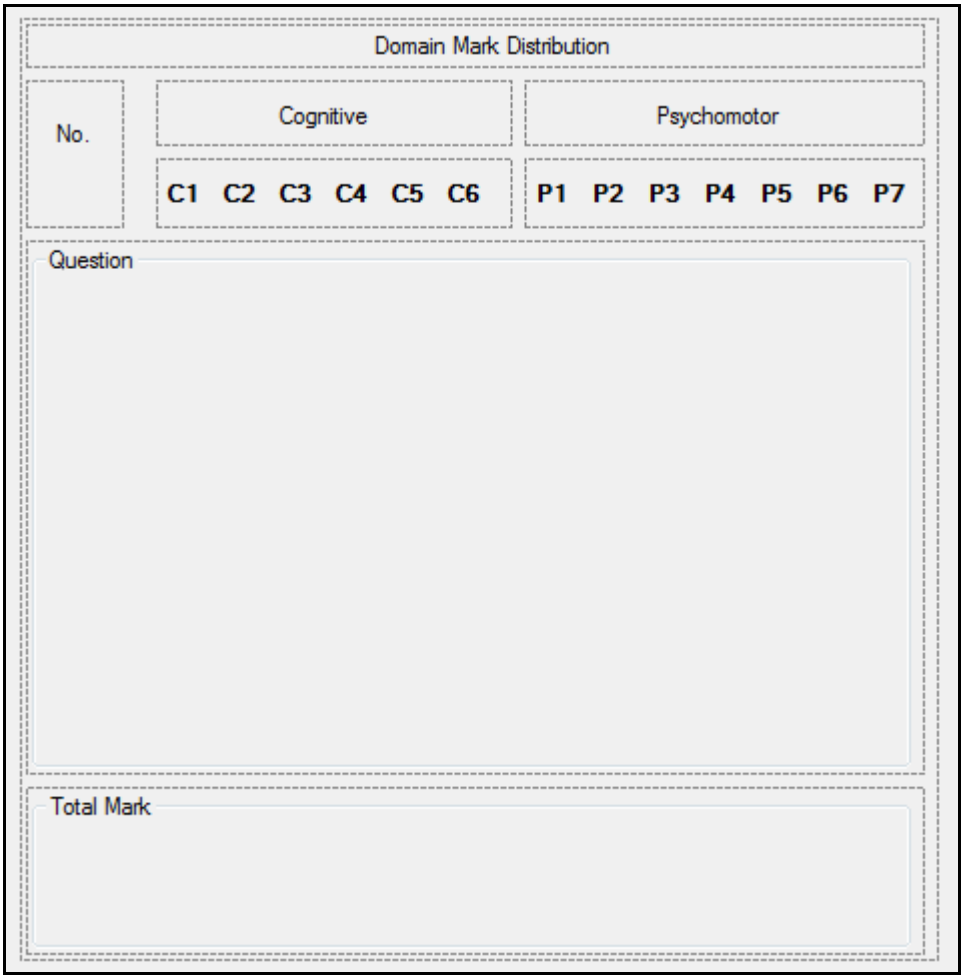


Figure 4.14: Story board for Domain Mark Distribution

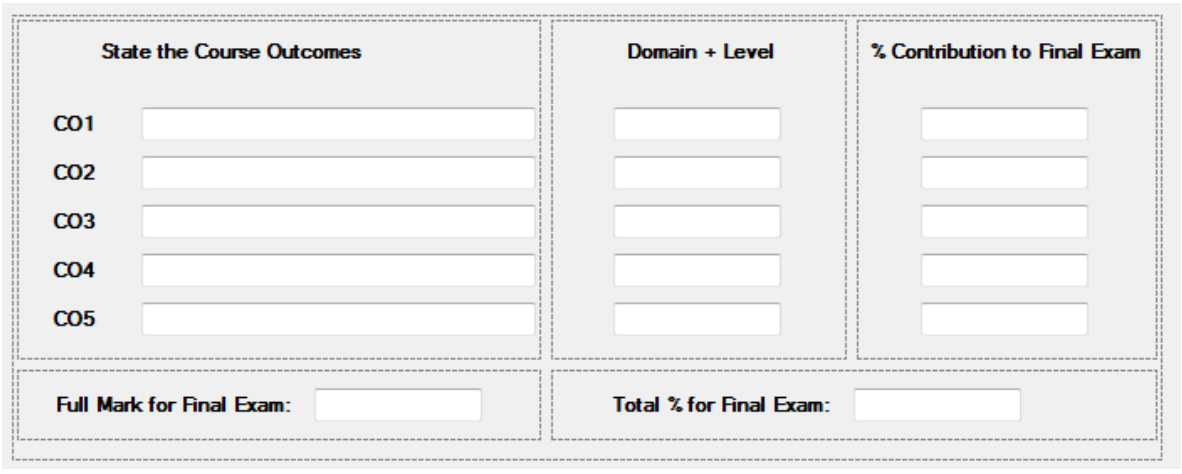


Figure 4.15: Story board for OBE Report

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Introduction

The purpose of this chapter is to explain about the finding of the results from data analysis based on the objectives.

5.2 Sample of main page from the system

The design for user interface should consist of software design principles such as consistency minimal surprise, and user diversity. The reason to have these principles is because the interface is the part where a user interacts with the system. It is important that a user interface is not confusing to the user. Below we will discuss details of the system interface that was design based on the story board drawn from chapter 4.

Whenever lecturer's login to the system, after validation the system will redirect the user from login page to the question bank page. Figure 5.1 in an sample question bank interface taken from the system which displays the list of questions from the database. The search bar is placed above the list of questions for easy access. As we can see from the figure also shows that on top of the question consists of two buttons "Add to Template" and "Update" as design.

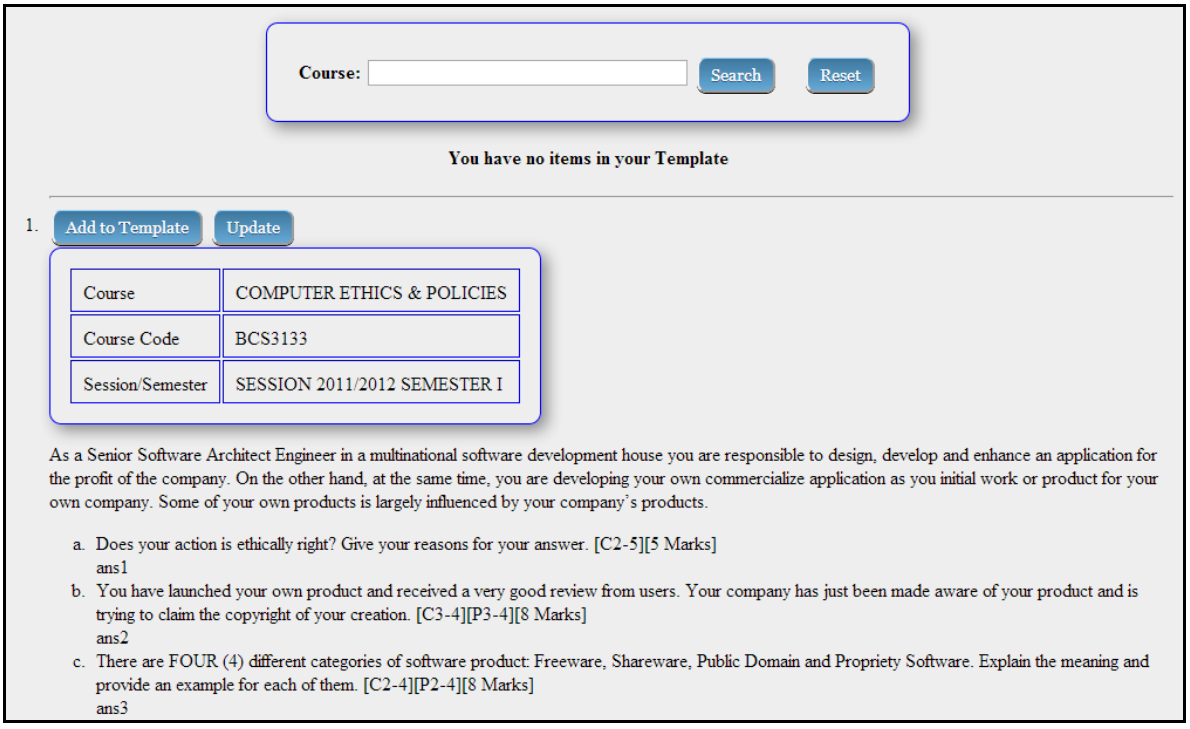


Figure 5.1: Question Bank Interface

The figure 5.2 shows the interface of question validation. This is the interface in which the lecturer will submit the details about the question to be asses by the system brute force string matching. From the figure, the table main focus is to be simplistic. The alternating color for the rows from the sub question table is to prevent the user from confusion while inserting the question.

The following figure 5.3 shows the interface of the result of brute force string matching. As we can see from the figure, it is based upon a sample sub question which has been submitted through an interface as shown in figure 5.2. This figure shows the result from the brute force string matching of the sub question. The table below it has three columns that display the cognitive analysis, psychomotor analysis and the mark to be the key in based upon the lecturer preferences.

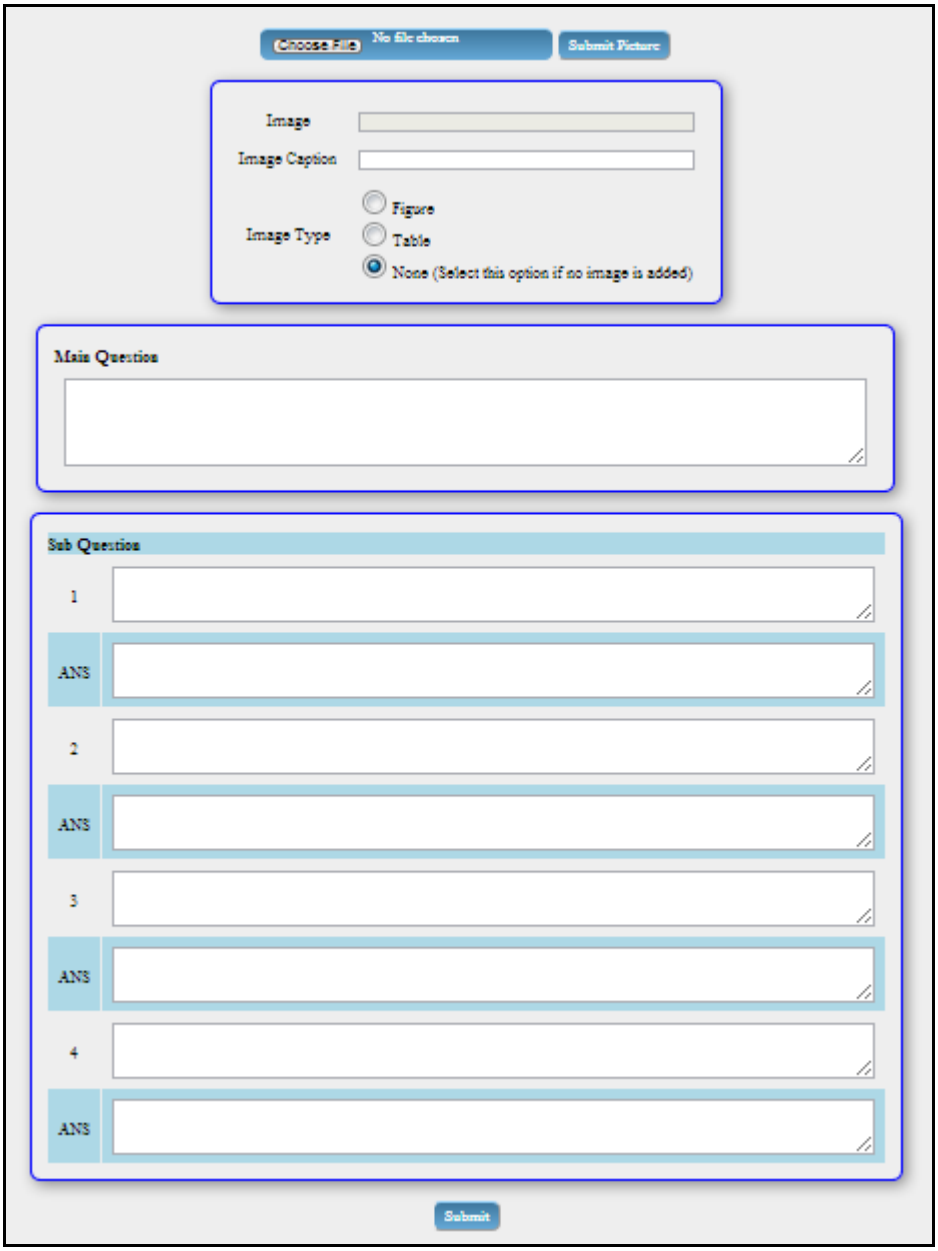


Figure 5.2: Question Validation Interface

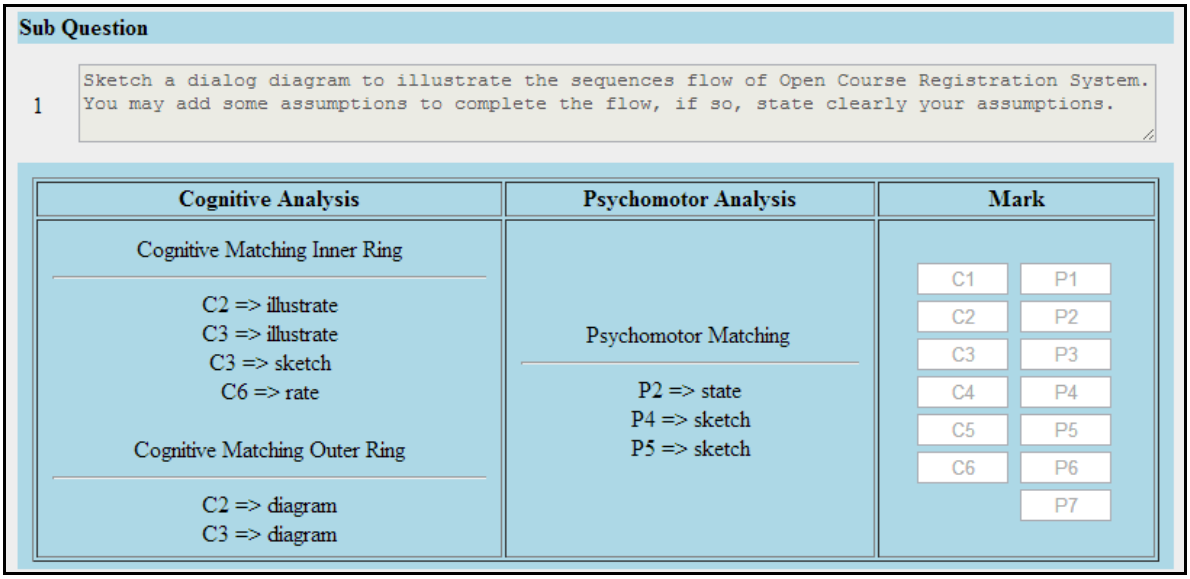


Figure 5.3: Result of Brute Force String Matching Interface

Besides validating and viewing the question, the second main process on the system is to generate the final exam template. There are total of 3 different template, which are final examination question paper, schema for final examination paper and OBE report for the final examination paper.

The front-page requirement's detail is a vital part of the final examination paper which will provide the student whom is taking the paper necessary information regarding the paper. All the detail which is needed to be inputted by the lecturer are as shown on the figure 5.4 such as the course name, course code, examination requirements and etc.

Next the OBE report for the final examination paper is a template which provides the detail's domain mark distribution table for the user to monitor whether the generated paper has passed course outcome requirement. The sample interface of domain mark distribution table can be viewed in the figure 5.5. Lastly, the course outcome details interface is shown in figure 5.6 is part on the OBE report which is required to show what is the requirement of the course outcome.

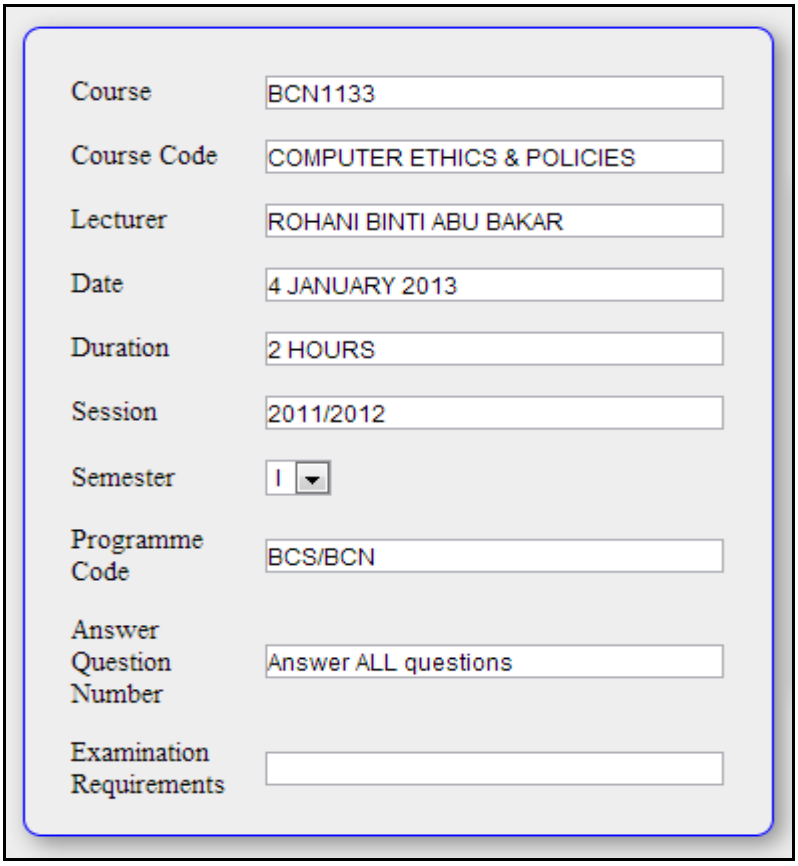
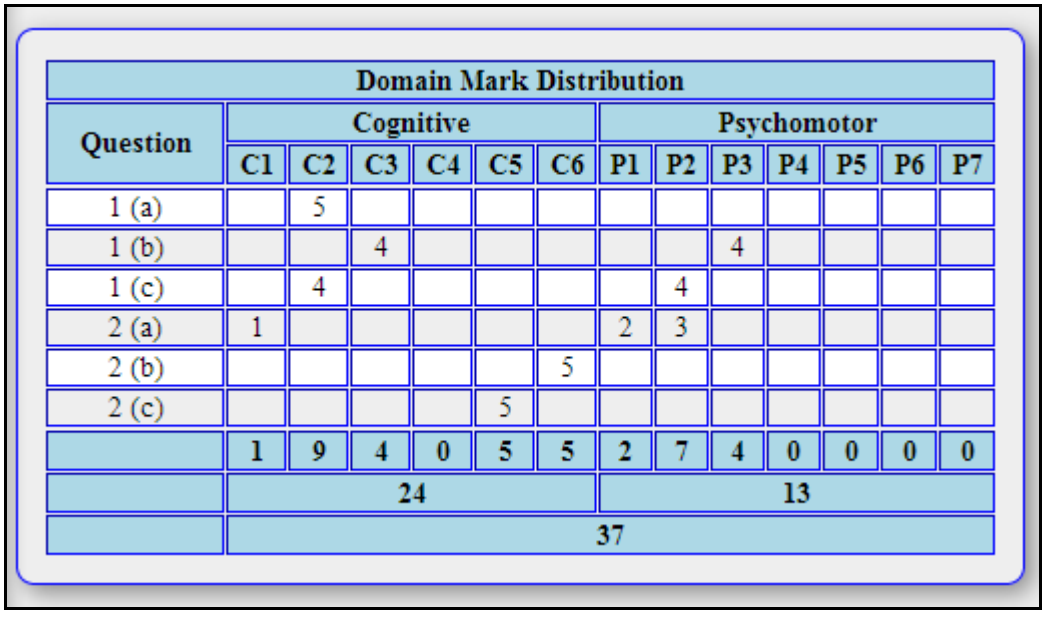


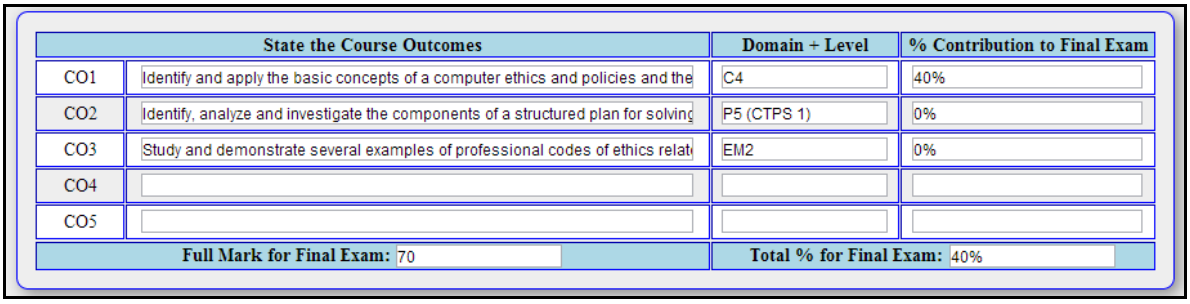
Figure 5.4: Final Exam Front Cover Details Interface



Domain Mark Distribution

Question	Cognitive						Psychomotor						
	C1	C2	C3	C4	C5	C6	P1	P2	P3	P4	P5	P6	P7
1 (a)		5											
1 (b)			4						4				
1 (c)		4						4					
2 (a)	1						2	3					
2 (b)						5							
2 (c)					5								
	1	9	4	0	5	5	2	7	4	0	0	0	0
	24						13						
	37												

Figure 5.5: Domain Mark Distribution Details Interface



State the Course Outcomes		Domain + Level	% Contribution to Final Exam
CO1	Identify and apply the basic concepts of a computer ethics and policies and the	C4	40%
CO2	Identify, analyze and investigate the components of a structured plan for solving	P5 (CTPS 1)	0%
CO3	Study and demonstrate several examples of professional codes of ethics relat	EM2	0%
CO4			
CO5			
Full Mark for Final Exam: 70		Total % for Final Exam: 40%	

Figure 5.6: Course Outcomes Details Interface

5.3 Result Analysis

The developed prototype, Automatic Question Determination Level Based On OBE Using Brute Force String Matching has met all the objectives of this project, which are:

i. To study string matching technique in order to evaluate the level of questions.
ii. To develop a prototype that will generate a final exam question template with standard fonts and spacing.
iii. To evaluate the level of the final exam questions according to the teaching and learning taxonomy published by UPM using string matching.

The first objective has been successfully achieved with the studies of different string matching technique Section 2.3 and finally choosing brute force string matching as the technique to be used. Furthermore, the demonstration of brute force string matching technique algorithm is covered in the Section 3.3 in the figure 3.6.

The complexity of the brute force string matching is stated as O(n.m). Which "n" is the length from the text while "m" is the length of the pattern. This complexity is displayed graphically with figure 5.7 and 5.8 (Stoimen, 2012). Both the graph shows rapidly growing function even with fix length of text or fix length of pattern.

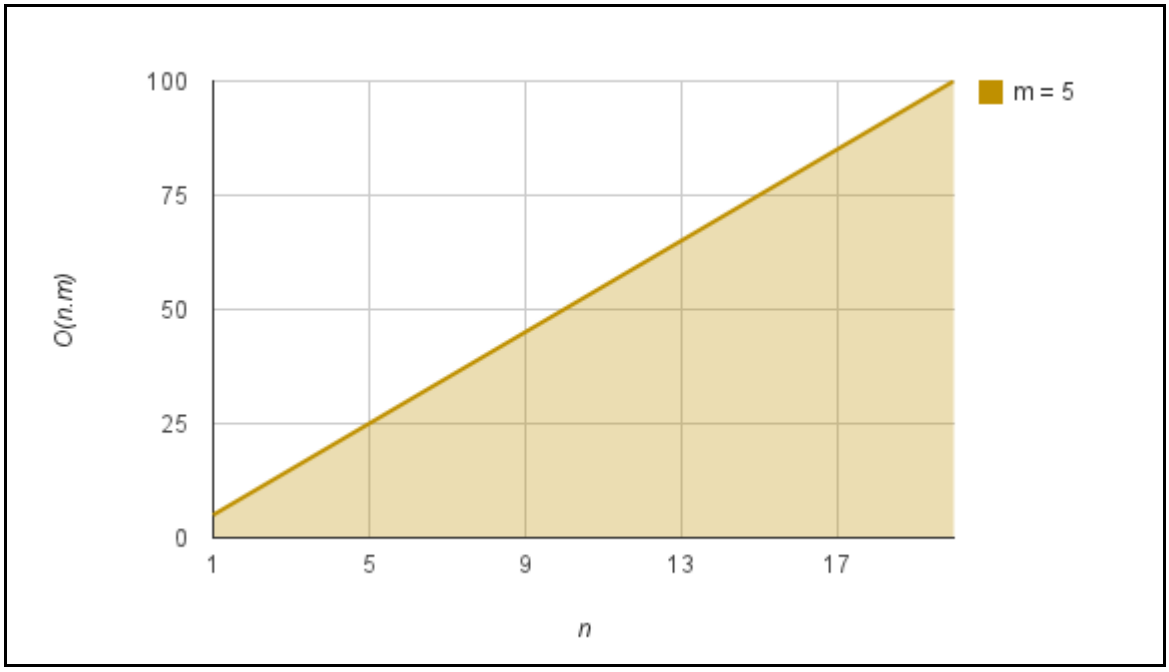


Figure 5.7: Complexity Graph with Fix Pattern

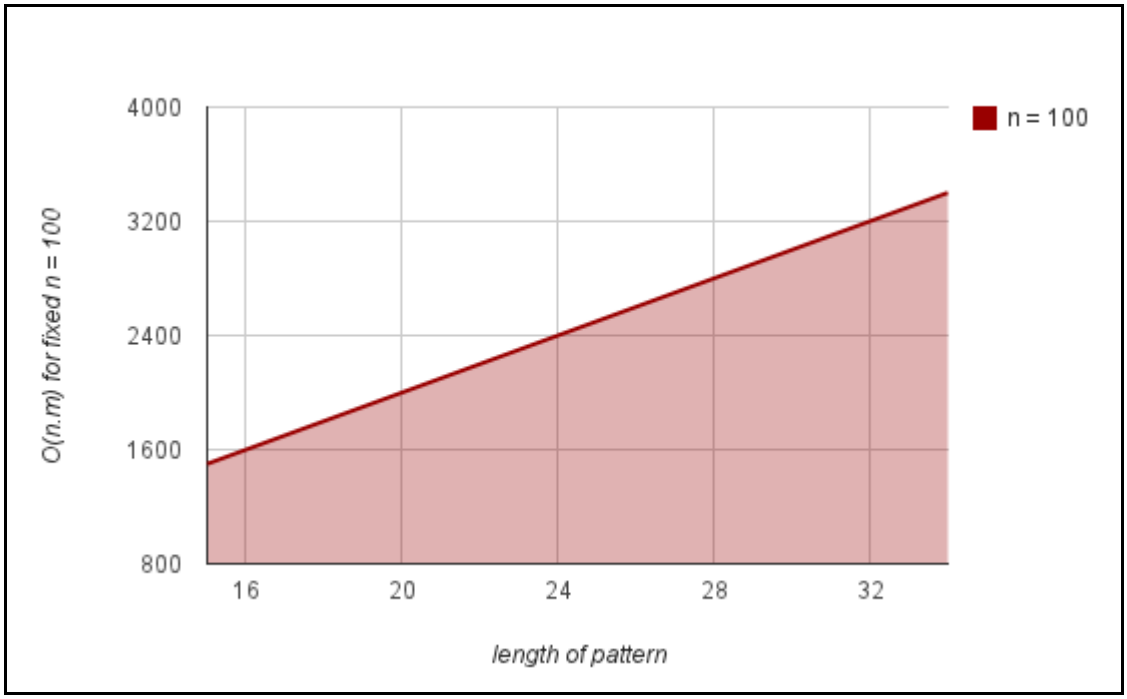


Figure 5.8: Complexity Graph with Fix Text

The second objective has been achieved and documented in the Section 4.5 Final Exam Template Generator. The section covers the requirements to generate the final exam template and the usage of a customized OOPHP library FPDF.

Finally, the third objectives have also been met by the implementation of the brute force string matching algorithm of the system which can be seen with explanation using a sample question on the pattern list from the teaching and learning taxonomy published by UPM shown in Table 4.1, Table 4.2 and Table 4.3. In Section 4.3 brute force string matching is displayed with the actual codes as shown on the figure 4.2.

5.4 Research Constrains

The Automatic Question Determination Level Based On OBE Using Brute Force String Matching has introduced and give awareness to the importance to this system to help lecturers to generate final exam templates, but this system still has its own constrains.

The first constrain found within this prototype is related to the OBE levels. It is mainly due to the limitation of the list from the teaching and learning taxonomy published by UPM. The limitation of this list that acts as the pattern for the brute force string matching also limits the variety of question that can be analyzed to determine the cognitive and psychomotor levels. In this study, we do not consider the affective domain as it should consider a diversity of inner domains, which is more complex.

The second constrain has been noted as the system only give the suggestion of domain levels to the user, unable to provide decision of the levels. It is not a type of decision support system which are able analyze previous data and present to the user.

Disadvantages of using brute force string matching also proves to be a constrains as it is a straight-forward technique which only matches the text and the pattern by each character. It does not concern with the pattern synonyms, words that are different but has

the same meaning or homonym, same words with different meanings. Without the ability to understand, the meaning to the pattern proves to be a problem such as the example given in the figure 4.4. As the result shows the detection of pattern "rate", which are match from the words "illustrate" which brings a different meaning.

CHAPTER 6

CONCLUSION

6.1 Introduction

The purpose of this chapter is to make a conclusion for the research that has been done and to provide future suggestion to produce a better system. The prototype system that has been developed has to achieve all the objectives stated in this thesis as shown in chapter 5 result and discussion.

The algorithm which has been studied brute force string matching has been implemented successfully into the system as it can be seen as the process of determining the question levels. While the software-development life cycle that is chosen in this project, which is the waterfall model has also worked well in this project. As each phase of the waterfall model provides high visibility and quality to the project. Miles's stones are set and evaluated from time to time to make sure progress is going as planned. Each task is understood and done properly before moving on to the next.

Lastly, the difficulty in this project mainly lies within the development within the system. By using the brute force string matching as the main function within the system, it has to produce a result which will link the result to its specific OBE domain levels. Generating final exam templates as PDF file is one of the main time-consuming process as it requires studies of an online OOPHP library called FPDF. To fulfill the requirements of the final exam formats it has to be customized to display a desired output.

6.2 Future Suggestion

In this thesis, it is found that the prototype system can be implemented with promising results. The system so far has proven itself by able to help the lecturer automatically format the final exam template to specific page specification and help reduce the difficulty and determining the cognitive and psychomotor levels. However, throughout the development within the system a few constrain has come across.

The system only gives the suggestion which in the end the user has to make the decision for the domain levels. It for the future suggestion, the system may analyze data or previous entry of question to present to the user just like an decision support system.

Factors such as the limitations on the teaching and learning taxonomy provided by UPM, which is used in the system are not enough to determine the cognitive and psychomotor levels. The disadvantages in using brute force string matching technique as it only considers words, which are unable to differentiate the synonyms and homonyms within them. This system can be improved by including the semantic research from natural language processing, which can distinguish the different meanings of words.

REFERENCE

1. Carol A. Tomlinson, Jay McTighe. 2006. *Integrating Differentiated Instruction and Understanding by Design: Connecting Content and Kids*. Association for Supervision & Curriculum Deve.
2. H. Basri, A. B. Che Man, W. H. Wan Badaruzzaman and M. J. M. Nor. 2004. Malaysia and The Washington Accord: What It Takes For Full Membership. International Journal of Engineering and Technology, **1**(1): 64 - 73.
3. Prof. Madya, Dr. Jaafar Jantan. 2009. A Brief Guide to Learning Domains.
4. Simpson, E. (1972). *The classification of educational objectives in the psychomotor domain: The psychomotor domain.* Vol. 3. Washington, DC: Gryphon House.
5. Krathwohl, D. R., Bloom, B. S., and Masia, B. B. (1973). *Taxonomy of Educational Objectives, the Classification of Educational Goals. Handbook II: Affective Domain*. New York: David McKay Co., Inc.
6. Susan T. Dumais (2005). Latent Semantic Analysis. *Annual Review of Information Science and Technology* **38**: 188
7. Foltz, P.W. 1996. Latent Semantic Analysis for text-based research. Behavior Research Methods, Instruments and Computers **28**(2): 197-202.
8. N. Wirth. 1985. *Algorithms and Data Structures.* Prentice Hall.
9. Boyer, Robert S.; Moore, J Strother. October 1977. A Fast String Searching Algorithm.". *Comm. ACM* (New York, NY, USA: Association for Computing Machinery) **20** (10): 762–772.
10. Donald E. Knuth. 1973. *Art of Computer Programming, Volume **1**: Fundamental Algorithms* . Addison-Wesley; Later Printing edition.
11. Foltz, P.W. and Martin, M.A. 2004, *Automated Team Discourse Annotation and Performance Prediction Using LSA*, Proceedings of HLT/NAACL
12. Stoimen. (March 27, 2012). Computer Algorithms: Brute Force String Matching. In stoimen's web log. Retrieved undefined, from http://www.stoimen.com/blog/2012/03/27/computer-algorithms-brute-force-string-matching/.

APPENDIXES

APPENDIX A

	Task Name	Duration	Start	Finish	Predecessors
1	**1.0 Project Plannning and Requirement Analysis**	**37 days**	**Wed 05-09-12**	**Thu 25-10-12**	
2	1.1 Discuss with supervisor	3 days	Wed 05-09-12	Fri 07-09-12	
3	1.2 Define problem statement, objective and scope	10 days	Mon 10-09-12	Fri 21-09-12	
4	1.3 Write Chapter 1 Introduction	10 days	Mon 10-09-12	Fri 21-09-12	
5	1.4 Submit Chapter 1 Introduction	0 days	Fri 21-09-12	Fri 21-09-12	4
6	1.5 Study project requirements and technique	24 days	Mon 24-09-12	Thu 25-10-12	
7	1.6 Write Chapter 2 Literature Review	24 days	Mon 24-09-12	Thu 25-10-12	5
8	1.7 Submit Chapter 2 Literature Review	0 days	Thu 25-10-12	Thu 25-10-12	7
9	**2.0 System and Software Design**	**62 days**	**Fri 26-10-12**	**Mon 21-01-13**	
10	2.1 Discuss overall approach and technique used	32 days	Fri 26-10-12	Mon 10-12-12	
11	2.2 Write Chapter 3 Methodology	32 days	Fri 26-10-12	Mon 10-12-12	8
12	2.3 Submit Chapter 3 Methodology	0 days	Mon 10-12-12	Mon 10-12-12	
13	2.4 Develop framework and model	30 days	Tue 11-12-12	Mon 21-01-13	
14	2.5 Write Chapter 4 Design and Implementation	30 days	Tue 11-12-12	Mon 21-01-13	12
15	2.6 Review Chapter 4 Design	0 days	Mon 21-01-13	Mon 21-01-13	14
16	**3.0 Implementaion and Integration Testing**	**50 days**	**Tue 22-01-13**	**Mon 01-04-13**	
17	3.1 Generate codes and testing	50 days	Tue 22-01-13	Mon 01-04-13	
18	3.2 Write Chapter 4 Implementation	50 days	Tue 22-01-13	Mon 01-04-13	15
19	3.3 Submit Chapter 4 Design and Implementation	0 days	Mon 01-04-13	Mon 01-04-13	18
20	**4.0 Verfication and Validation**	**26 days**	**Tue 26-03-13**	**Tue 30-04-13**	
21	4.1 Verify and validate system	14 days	Tue 26-03-13	Fri 12-04-13	
22	4.2 Write Chapter 5 Results and Discussion	14 days	Tue 02-04-13	Fri 19-04-13	19
23	4.3 Submit Chapter 5 Results and Discussion	0 days	Fri 19-04-13	Fri 19-04-13	22
24	4.4 Summarize overall research	7 days	Mon 15-04-13	Tue 23-04-13	
25	4.5 Write Chapter 6 Conclusion	7 days	Mon 22-04-13	Tue 30-04-13	23
26	4.6 Submit Chapter 6 Conclusion	0 days	Tue 30-04-13	Tue 30-04-13	25