i-BANK QUESTION: SUBJECTIVE QUESTION (iBQ-SQ)

MUHAMMAD ARIF BIN OMAR

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BORANG PENGESAHAN STATUS TESIS

JUDUL   : i-Bank Question: Subjective Question (IBQ-SQ)

SESI PENGAJIAN   : 2015/2016

Saya MUHAMMAD ARIF BIN OMAR

mengaku membenarkan tesis (PSM/Sarjana/Doktor Falsafah) ini disimpan di Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dengan syarat-syarat kegunaaan seperti berikut:

1. Tesis dan projek adalah hakmilik Universiti Teknikal Malaysia Melaka

2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan untuk tujuan pengajian sahaja.

3. Perpustakaan Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi 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

_____   _____

(TANDATANGAN PENULIS)   (TANDATANGAN PENYELIA)

Alamat tetap: NO, 161 Kampung Jeniang   (En Yahya Bin Ibrahim)

08700 Gurun

Kedah Darul Aman

Tarikh: 26/9/2016   Tarikh: 26/8/2016

CATATAN: * Tesis dimaksudkan sebagai Laporan Projek Sarjana Muda (PSM)

           **Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa.

i-BANK QUESTION: SUBJECTIVE QUESTION (iBQ-SQ)

MUHAMMAD ARIF BIN OMAR

This report is submitted in partial fulfillment of the requirements for the Bachelor of Computer Science (Database Management)

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA
2016

DECLARATION

I hereby declare that this project report entitled

i-BANK QUESTION: SUBJECTIVE QUESTION (iBQ-SQ)

is written by me and is my own effort and that no part has been plagiarized

without citations.

STUDENT : _____ Date : 2 6 / 8 / 2016

(MUHAMMAD ARIF BIN OMAR)

I hereby that I have read this project report and found

this project report is sufficient in term of the scope and quality for the award of

Bachelor of Computer Science (Database Management) with Honors.

SUPERVISOR : _____ Date : 26/8/2016.

(EN YAHYA BIN IBRAHIM)

DEDICATION

To my mother Kelesem binti Omar

To my supervisor, Mr Yahya Bin Ibrahim.

To my friend and family.

ACKNOWLEDGEMENT

Firstly, I would like to express my grateful to the Almighty Allah S.W.T for giving me the opportunity to complete my journey in making the thesis. Without the will of Him it would be impossible to finish the thesis. All of the strength and idea is form Him, He is the true knowledge Owner. Salutations and peace to His beloved messenger; Prophet Muhammad S.A.W.

Beside I would like to express my sincere gratitude to my supervisor Mr Yahya bin Ibrahim who taught me the basic of coding and problem solving, without him programming will be a set of problem but with his full dedication to teach his student, its transform the idea of problem into puzzle in which, there will be solution at every move.

I also want thank to my mother who always supporting me spiritually throughout my final year project, whiteout her prayer I would not be able to stood at where I am now. I also want to thank to my brother in law for giving me the guidance and idea for making the project. His help will not be forgotten.

Last but not least, I would like to thank to all my friend who has helping me directly or indirectly on the i-Bank Question: Subjective Question (IBQ-SQ)

THANK YOU

ABSTRACT

i-Banking Questions (iBQ-SQ) is a platform for teachers to upload their questions to the whole world. iBQ-SQ system includes primary and secondary schools. All teachers and non-teachers can submit their specialty in creating a good question and the whole world will recognize them about their questions. To entice them to make a good question, the system will automatically reward uploaders based on the number of questions for each uploader. In addition, this system also provides a search utility to find another question based on primary or secondary schools or by years of study, topic, sub-topic or level of difficulty. iBQ-SQ is developing a system development life cycle (SDLC). The waterfall model was chosen as the project methodology. The overall system is using Java Server Page (JSP) and Oracle 9i. Entity relationship diagram (ERD) and data flow diagram (DFD) is a design that is designed to make the system easier to understand.
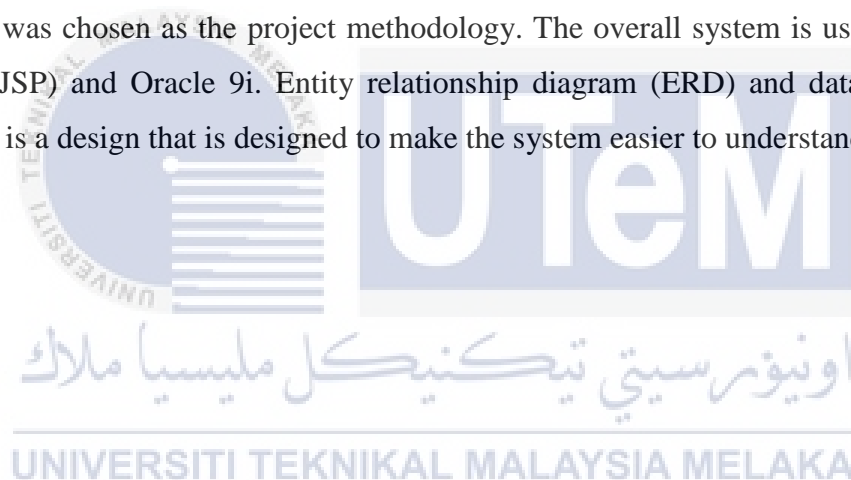
ABSTRAK

i-Banking Soalan (iBQ-SQ) adalah salah satu platform untuk guru untuk memuat naik soalan mereka ke seluruh dunia. iBQ-SQ meliputi sekolah rendah dan sekolah menengah. Semua guru dan bukan guru dapat mengemukakan soalan dan mereka yang membuat soalan yang baik dapat peluang untuk seluruh dunia akan mengakui soalan mereka. Untuk menarik mereka membuat satu soalan yang baik, sistem secara automatik akan memberi balasan kepada pemuat naik berdasarkan bilangan soalan bagi setiap pemuat naik. Selain itu, sistem ini juga menyediakan utiliti carian untuk mencari soalan lain berdasarkan sekolah rendah atau menengah atau berdasarkan tahun pengajian , topik , subtopik atau tahap kesukaran. iBQ-SQ adalah membangunkan menggunakan kitaran hayat pembangunan sistem (SDLC). Model Waterfall di pilih sebagai metodologi projek. Sistem keseluruhan menggunakan Java Server Page (JSP) dan Oracle 9i. Entiti rajah hubungan (ERD) dan rajah aliran data (DFD) adalah direka untuk membuat aliran sistem yang lebih mudah difahami.

TABLE OF CONTENTS

CHAPTER V I   TESTING

CHAPTER VII   CONCLUSION

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

iBQ-SQ	-	i-Bank Question Subjective Question
CPU	-	Central Processing Unit
DBMS	-	Database Management System
DFD	-	Data Flow Diagram
ERD	-	Entity Relationship Diagram
GUI	-	Graphic User Interface
RAM	-	Random Access Memory
SDLC	-	System Development Life Cycle
UTeM	-	Universiti Teknikal Malaysia Melaka

CHAPTER I

INTRODUCTION

1.1   Project Background

i-Bank Question: Subjective Question (iBQ-SQ) Paper is one of the platform for teacher to upload their question to the worldwide. This iBQ-SQ system coverage primary and secondary school. All teachers and non-teachers are able to forward their speciality in making a good questions and the worldwide will acknowledge them regarding to their question. To attract them making a good question, the system will automatically reward the uploader based on number of upload question for each uploader. Besides, this system also provide search.

utility to search other question based on primary or secondary school or based on year of study or topic or subtopic or level of difficulty.

1.2   Problem statement

i. There are no platform for teacher and non-teacher to upload their question paper to the worldwide.

   - Teacher difficult to find the platform to show their idea's of a good question to other teacher.

ii. It is hard and takes time to know which uploader are the most upload to be reward.

   - Teacher that have a good question did not get a reward and acknowledge for thier idea's.

iii. Takes time to know what kind of question are the most upload

   - Teacher hard to know who are the most brilliant idea's of question and has many good question.

1.3   Objective

i. To develop a platform for teacher and non-teacher uploading subjective question paper together with schema

   - User can upload their question based on subject, chapter and topic.

ii. To investigate which uploader are the most upload and automatically awarding them.

   - User can know which teacher are the most upload question based on month.

iii. To investigate which of the question are have the most question download.

   - User can know which question are the most download

1.4   Scope

There are two main scopes involved in the develop i-Bank Question: Subjective Question (iBQ-SQ) Paper.That are the user and module scope. User scope focuses on the system's user only and their roles while module scope determines the details of function that is categorized by each module.

1.4.1   User Scope

There are one users only in i-Bank Question: Subjective Question (iBQ-SQ) Paper which are the teacher. The teacher is in charge of managing their question. Besides that, the teacher also can see which teacher are frequently upload and the highest download rating.

1.4.2   Modules Scope

i.   Question Management Module

Question management module will keep a record of all question currently based on subject, chapter and topic. It can be update and delete. So, teacher can get a full control of their question.

ii.   Other Question Module

User can view other question and see detail about the question and search question based on subject, chapter and topic

iii.   Rewarding Module

User can see the most top uploader of question monthly.

1.5 Project significance

i. Store a new question examination that allow other to access the question.

- This system help teacher that have new idea for the examination question to bring the idea to worldwide.

ii. Data Storage be more structured

- Data storage techniques are used to improve data storage.

iii. Reduce Time

- This system also will use less of time for the accessing data or information process data will be faster.

iv. Reduce Use of Paper

- By developing this system, the operation cost can be reduced for a long time period.

v. To Make the System Management More Efficient and Effective

- The database systems are designed with the aim of providing more efficient access to information and improve the management of information storage to be more systematic.

1.6 Expected Output

i. Teacher can upload a question that are suitable and store in follow the subject, chapter and subtopic

- Teacher can share the question to other by digitalized their question in this system.

    ii.   Teacher can review the top uploader question and download question.

- The analysis help teacher to know who are the teacher that have new quality and famous question.

    iii.   Teacher can easily find idea and new question in this system.

- Teacher can search question int this system based on subject,chapter and subtopic.

1.7   Conclusion

i-Bank Question: Subjective Question (iBQ-SQ) Paper is an efficient system to for the teacher to exposed their question to other teacher. It also can give credits to the teacher that is always have a new question and share it to worldwide. It brings a benefit to other teacher and help them to always get a new updated question that are still in the scope of the subject, chapter and subtopic.

CHAPTER II

PROJECT METHODOLOGY AND PLANNING

2.1 Introduction

Methods of development for this i-Bank Question: Subjective Question (iBQ-SQ) Paper is waterfall method. For this model if any problems occur in any stages, it can be detecting and refer to stages before and make an error correction for it. In advance, waterfall is simple approach and argue, easily understandable and explainable phases. There are stages in waterfall model, which are Analysis, Design, Implementation, Testing, and Maintenance. Every stage will only start if the stage before have been finished or nearly finish. Thus, Waterfall model are chosen based on System Development Life Cycle (SDLC) as methodology to develop our system.

2.2   Project Methodology

The current system that are used now does not efficient and effective during the operation. So that this i-Bank Question: Subjective Question (iBQ-SQ) Paper will be used to further improvement. The Waterfall Model in DBLC starts from 4 stages that are Analysis, Design, Implementation, Testing and Maintenance. The System Development Life Cycle (SDLC) contains six phases: database initial study, database design, and implementation and loading, testing and evaluation, operation and maintenance and evaluation.

2.2.1   Database Initial Study

The purpose of the database initial study is to analyze the situation faced. Next is define problems and constraints where information can be divided into two categories which are formal and informal. Most of the information are difficult to search. This is because, the current system records the information of data manually that need to be search from one file to another file. Not just that, the information is not secure because anyone can read the file. Other than that is define the objectives where the database system that wants to be developed must be designed in order to solve at least the major problems that identified during the problem discovery process. The initial study phase where contribute to the problem solution.

After that is define scope and boundaries. The system's scope will define the extent of the design related to the operational requirement. By knowing the scope, it will help to define the required data structures, the type and numbers of entities, the physical size of the database and so on. The boundaries are known as external to the system. Boundaries also required by existing hardware and software to accomplish system goals. Finally, the to-be database system is analyzed using the Entity Relationship Diagram. Then, the

project work plan and Gantt chart will build to develop this system.

2.2.2   Database Design

The second phase focuses on the design of the database model that will support the objectives. The conceptual design of the i-Bank Question: Subjective Question (SQ) Paper is made using Entity Relationship Diagram (ERD). This data modeling will be used to create the abstract database structure to be easier to understand. Moreover, it represents a clear view of the business and its' functional parts. The selection of the DBMS software is important to the information's system for a smooth operation. The end users also must be always aware of both DBMS and the database. After that, the logical design is developing by using Data Dictionary and Data Normalization. The physical design is then developing when Data Schema is produced. It can be defining as a process of select the data storage and data access characteristics of the chosen database.

2.2.3   Implementation and Loading

During this phase, the database for i-Bank Question: Subjective Question (iBQ-SQ) Paper is actually built by using the Data Definition Language (DDL), Data Manipulation Language (DML) and Data Control Language (DCL). In modern relational DBMS, a new database implementation requires the creation of special storage-related constructs to address the end-user tables. After the database has been created, the data must be stored in to the database tables. During the implementation and loading phase, other performances, security, backup and recovery must be address in the system.

2.2.4 Testing and Evaluation

Once the data have been loaded into the database, testing and evaluation phase occurs that will test and fine tunes the database for performance, integrity, and concurrent access and security constraints. The testing and the evaluation phase using the database tools. If the database implementation fails to meet the system's evaluation criteria or requirement, several options will be considered to enhance the system:

- For performances related issues, the designer must consider fine tuning specific system and DBMS configuration parameters. The best sources of information are the hardware and software technical reference manuals.

- Modify the physical design.

- Upgrade or change the DBMS software or the hardware platform

2.2.5 Operation

Once the database has passed the evolution stage, it is considered to be operational. At this point the database, management and users will compose a complete information system. The beginning of the operational phase consistently starts the process of the system evolution. When all the targeted end-users entered the operation phase, the problems that could not predict during the testing phase can be detected.

2.2.6 Maintenance and Evolution

The database administrator must be prepared to perform routine maintenance activities within the database. Some of the required periodic maintenance activities included such as follows: -

- Preventive maintenance (backup).

- Corrective maintenance (recovery).

- Adaptive maintenance (enhancing performance, adding entities and attributes and so on).

- Assignment of access permission and their maintenance for new and old users.

- Improve the efficiency and usefulness of system audits and to monitor system performance.

- System security using access level.

∴

2.3 Project Schedule and Milestone
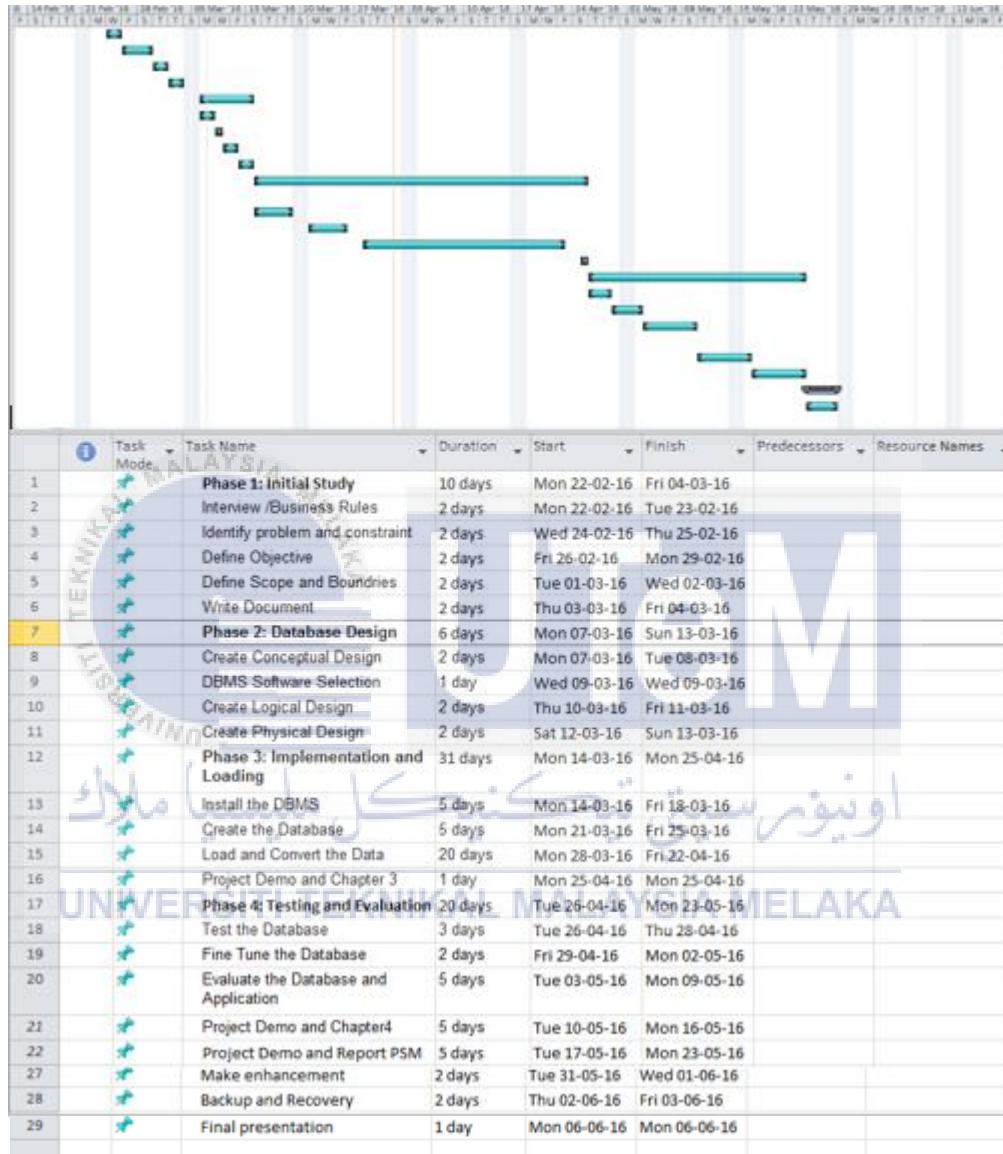
Figure 2.1: Project schedule and milestone

2.5   Conclusion

As a conclusion, with the uses of System Development Life Cycle (SDLC) the project is on schedule. The DBLC is composed of six major phases; Database Initial Study, Database Design, Implementation and Loading, Testing and evaluation, Operation, Maintenance and Evaluation. This chapter is covered on introduction of the chapter, domain for the system, project methodology and project schedule and milestone.

CHAPTER III

ANALYSIS

3.1   Introduction

System analysis is an important phase in methodology to identify the problems and the technical feasibility. Analysis are depending to the requirements of the system. It focused on the problem domain that must have a solving. Analysis phase is to further understand the user's requirements and the problem domain. Besides that, analysis is also to show and describe the actions of the system and to develop the best system in order to meet the needs of user. The problems that occur must be documented and be solved during development phase. The process of this analysis is requirements modeling such as non-functional requirement, functional requirements and other such as software and hardware requirements will be discussing further in this chapter.

3.2   Problem analysis

To implement and develop a system successfully, it must follow the procedures. Objective of the project is to ensure that teacher can upload question using this system. The current system that use by teacher is cannot upload specific question. It just comes with a set of question with all the chapter or topic. Teacher need to arrange the question manually for the specific topic or chapter. If teacher need a specific topic, they need to find manually from a bundle of question. The current system also only has question from provided only. Teacher cannot add their own question or share their question to others. Student cannot use the system because it provided of teacher only, so student hard to find the question for their exercise.

3.3   The proposed improvements/solutions

The current system that teacher use to find the question is only past year and them provide question only. Teacher cannot make their own question and share their question to others.

Figure 3.1: Sasbadi question bank system

Figure 3.2: Page for choose topic

The current system only can choose the topic without specific question for example chapter, topic, category and level. Their system also only for the teacher, so student cannot use the system.

Figure 3.3: Login page

System to be develops to replace the current. This system wills easier the teacher and also student to generate the question for the base on their

subject, chapter, category, level and topic. So that teacher can share their question for all students.

Figure 3.4 System to be develop

3.4   Requirement analysis of the to-be system

The system is for teacher to upload question and answers schema and data was recorded in the system. System needs the teacher to manage the question and give the answer schema to their student. Requirement analysis is a important requirement in hardware, software and other aspect.

3.4.1   Functional Requirement (Process Model)

This i-Bank Question (Subjective Question) provides the information about the question and answers for the teacher This system about question bank for teacher stores their question and shares it with other teachers. Teacher can view all their

download history or delete it. This system is develop using Java EE-Eclipse, and sublime text.

Figure below shows the Flow Chart and Data Flow Diagram (DFD) how the system would interact with teacher and student. Teacher need to login into the system to download the question with answers schema and student no need to login to download the question but without answer schema.
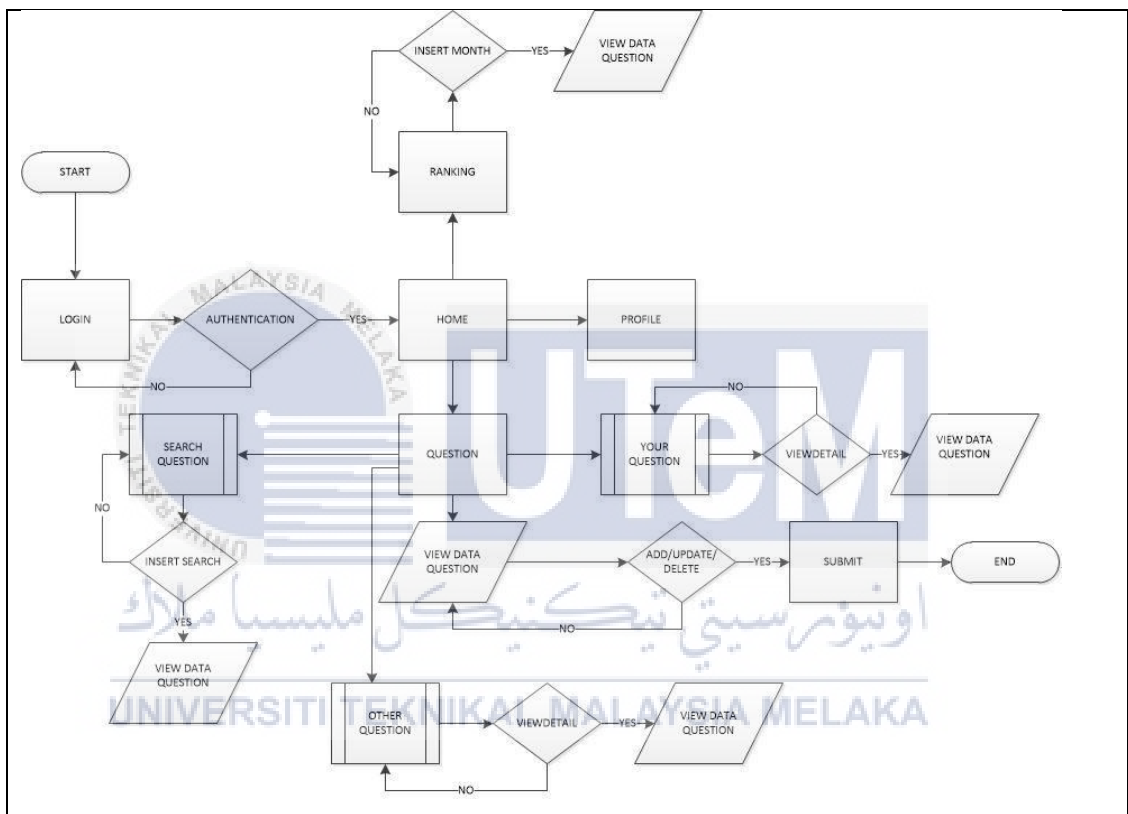
Figure 3.5: Flow chart for teacher

Figure 3.7 shows the context diagram that shows how the system boundaries, external entities that interact with the system, and the major information flows between the entities and the system.
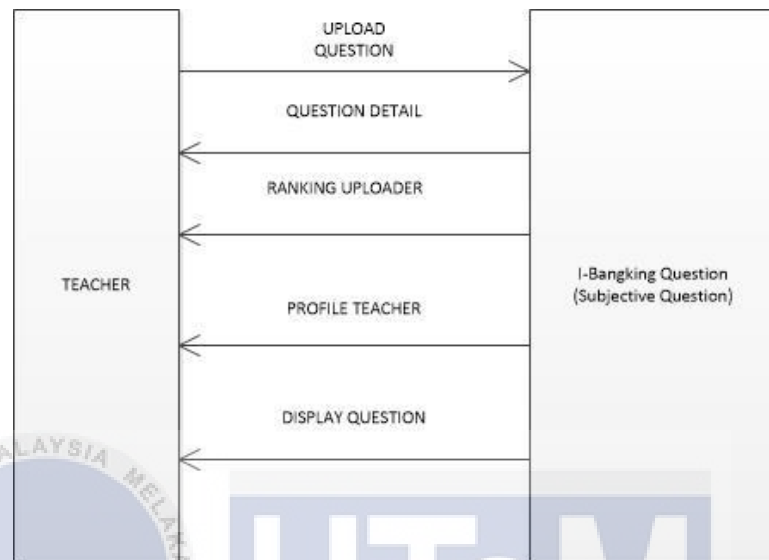
Figure 3.6: Context Diagram of system to be develop

A data flow diagram (DFD) is a graphical representation the flow of data through an information system. DFD also be used for the visualization of data processing. The figure below shows the flow of the system and the operation of that system.
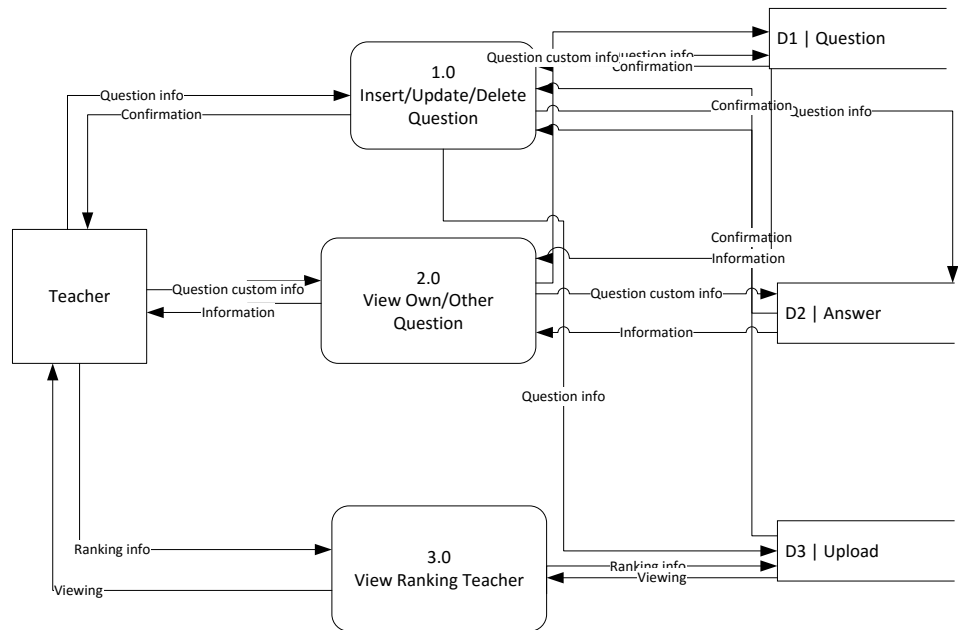
Figure 3.7: DFD Level 0 of System to be developed
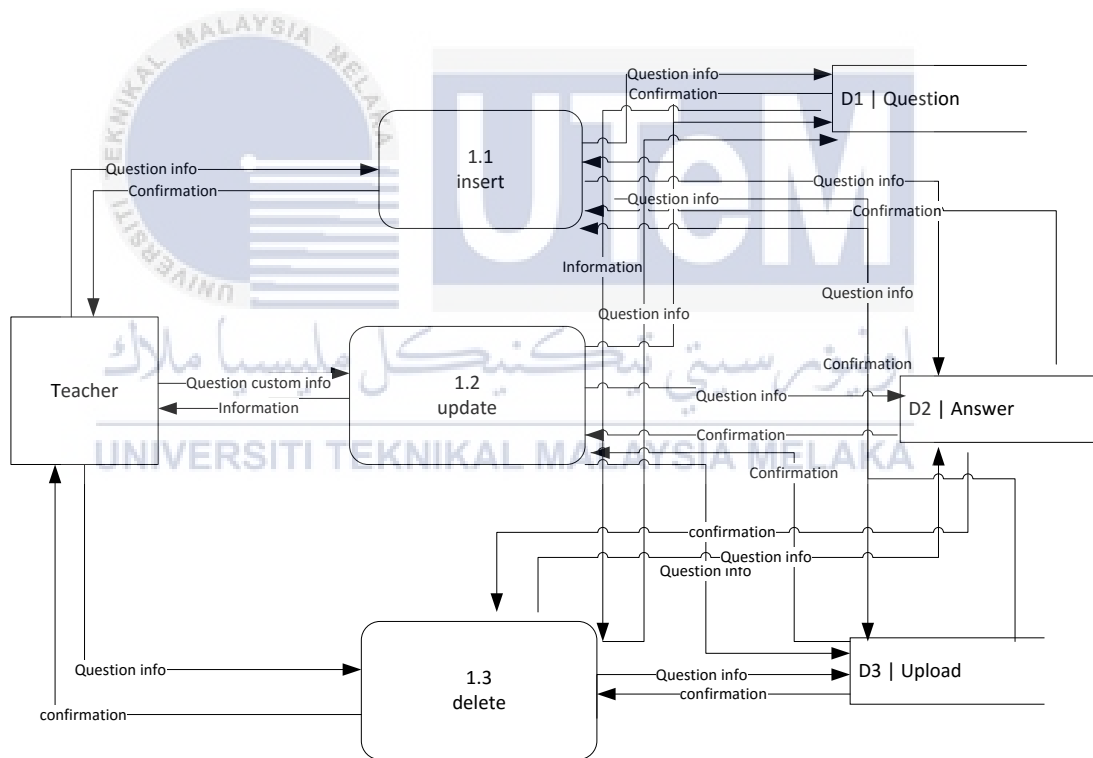
Figure 3.8: DFD Level 1 for Manage Question
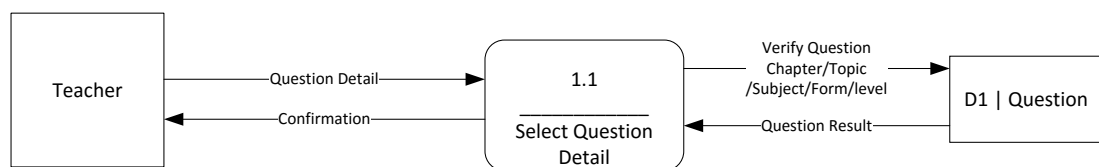
Figure 3.9: DFD Level 1 Teacher See Question Detail
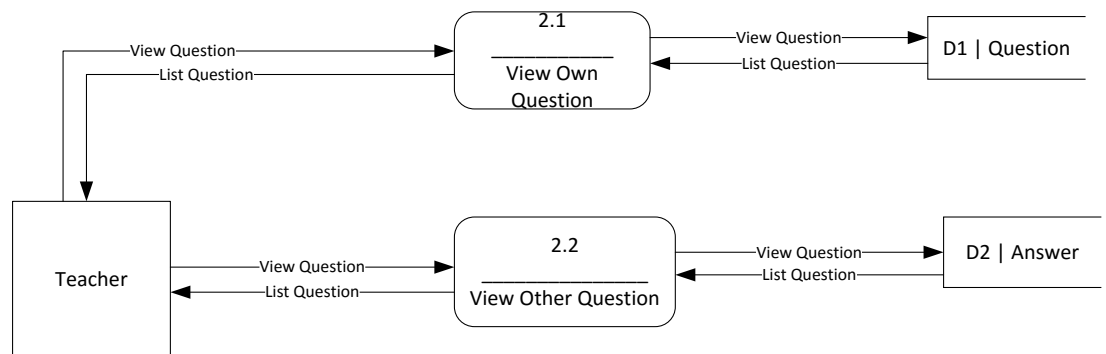
Figure 3.10: DFD Level 2 for Teacher Own and other

Question
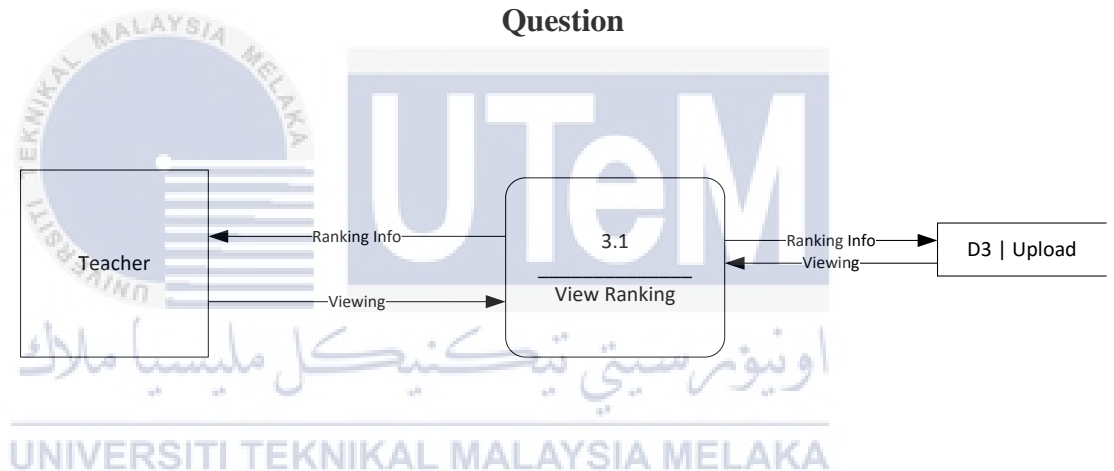
Figure 3.11: DFD Level 3 for View Ranking

3.4.2 Non-functional Requirement

The non-functional requirement of this system have to consider for make sure the system in well performance. The maintainability is important because it take large amount to spend on the changing. Secondly is reliability, where the system has capability to maintain its performance.

The third one is security. The integrity requirements determine the properties of the security system, or restrict access to its data to specific users and protect the privacy of the data entered into the software. For example, the teacher must login to make any changes to the question and answer.

3.4.3   Other Requirements

Below are defined the specific project requirement in terms of software and hardware to be develop this i-Bank Question (Subjective question)

3.4.3.1   Software Requirement

Table 3.1: Software Requirement for Development

Software	Description
Java EE-Eclipse	Tools for Java developers creating Java EE and Web applications
Sublime Text Editor	Used to read and write the coding for the system.
Microsoft Visio 2013	Used to create or draw diagrams such as ERD for database design, Context Diagram and DFD to show the functional requirement of the system and Flow Chart of the system.
Microsoft Office Word 2013	Used for the documentation.
Microsoft Project	Used to manage the project purpose such as Gantt Chart to show the timeline or milestones for the project development

Table 3.2: Software Requirement for Server

Software	Description
Apache Tomcat v8.0 Server	Web server that implements Java EE specifications - JavaServer Pages (JSP)
Microsoft Windows 8.1 Pro	Operating System as a platform for DBMS and system development installed on it. Windows 8 is new released from Microsoft and has better performance than Windows 7.
Oracle	Database to store the data for this system.

Table 3.3: Software Requirement for Client

Software	Description
Mozilla Firefox / Google Chrome / Internet Explorer / Edge	Web browser to preview this website. It is recommended for user to using latest version of it.

3.4.3.2   Hardware Requirement

These hardware requirements are must meets server and client need such as personal computer. Nowadays, perhaps the hardware for client and server is much better than these minimum requirements. The hardware requirements are shown in Table 3.4.

Table 3.4: Hardware Requirement

Hardware	Description	Server	Client
Hard Disk	Hard disk is the main storage in a computer where all the software installed on it.	Minimum 10GB free disk space	Minimum 300 MB free Disk space
Memory (RAM)	Memory is defined as Random Access Memory (RAM) provides space for the computer to read and write data to the accessed by the Central Processing Unit (CPU) or processor	Minimum requirement of memory is 2 GB, though 3 GB is recommended.	Minimum 512 MB of memory, through 2 GB is recommended.
Processor	Processor is the electronic component which is act as 'brain' for a computer. The higher the processing speed, the better its performance.	Minimum 2.27 GHz speed of CPU processor.	Intel Pentium Minimum 1.3 GHz speed of CPU processor.

3.5   Conclusion
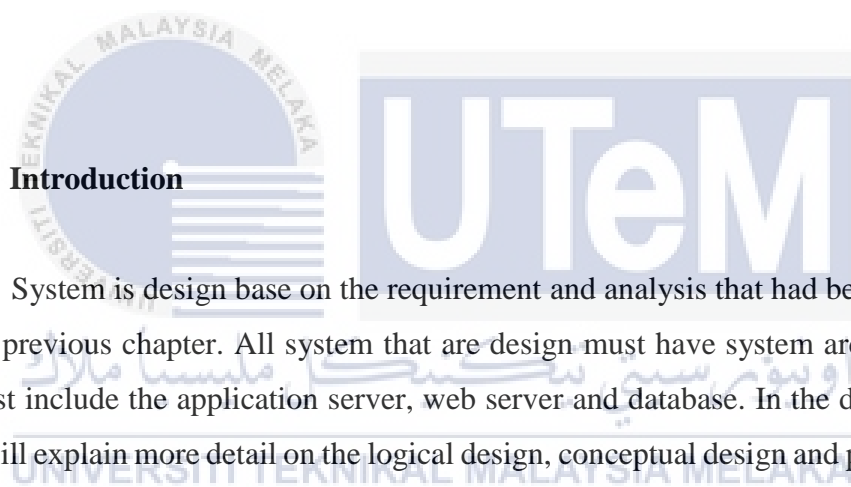
In this chapter, it was discussing about requirement analysis and current system. Collection of information about the system that are meets the system requirements are be used to improve the system requirements that are related to the i-Bank Question (Subjective Question).

CHAPTER IV

DESIGN

4.1   Introduction

System is design base on the requirement and analysis that had been collected on the previous chapter. All system that are design must have system architecture. This must include the application server, web server and database. In the database design, it will explain more detail on the logical design, conceptual design and physical design. All the design that are create are represent on how the system works. This can be view on the Graphical User Interface (GUI) of the system. All the information in current system will be the information for the developer to design this project.

4.2   System Architecture Design

Application architecture is characterized by the functional decomposition of applications, service components, and their distributed deployment. By breaking this system down in such a manner, it could provide an improved scalability,

availability, manageability, and good resource utilization. Scalability refers to a system's ability to handle increasingly heavier loads from users (activity). In other words, this system will be able to easily handle the increase without slowing down, or worse, breaking down completely. A "tier" itself is nothing more than a functionally separated hardware and software component that performs a *specific function*. A i-Bank Question: Subjective Question (iBQ-SQ) Paper has been broken up into various levels of functionality, each capable of some degree of horizontal scaling.
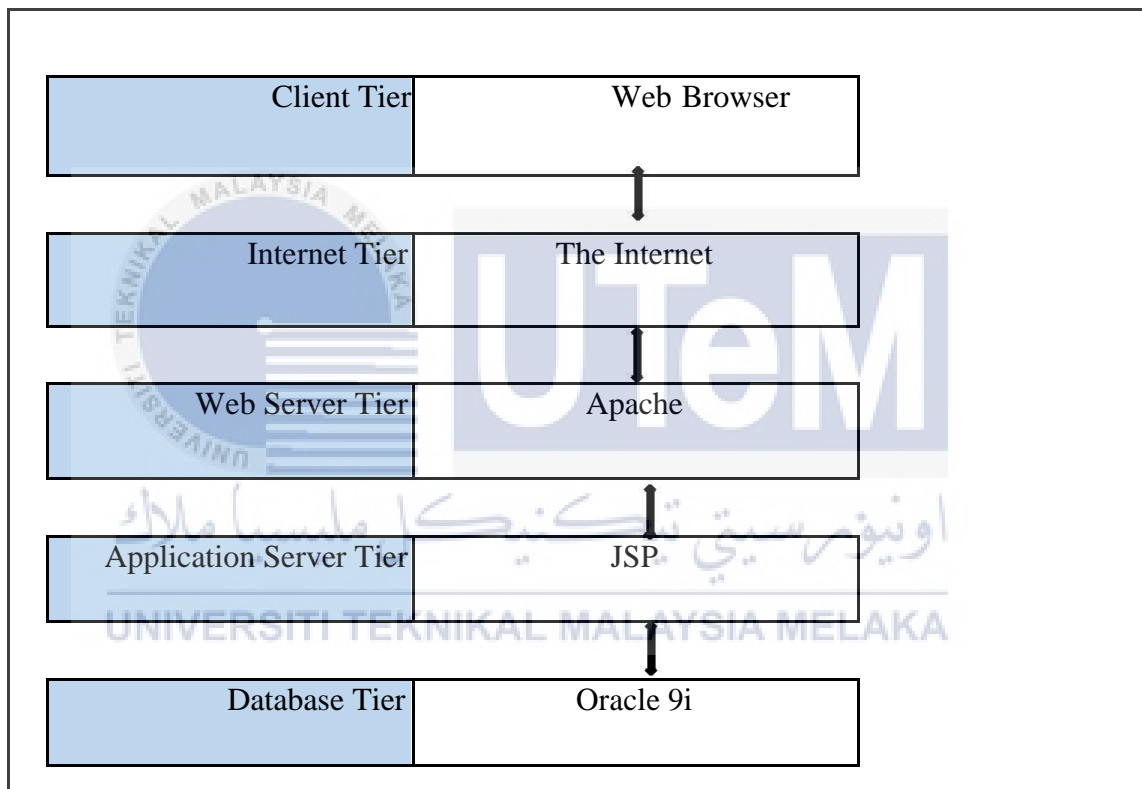
Figure 4.1: System Architecture i-Bank Question (iBQ-SQ)

4.3 Database Design

4.3.1 Conceptual Design

Conceptual design is the process of construct a data model which is entity relationship (ER) model. It helps to check the redundancy and validate the model. ER contains of entities, relations and attribute.

4.3.1.1 Entity Relationship Diagram (ERD)
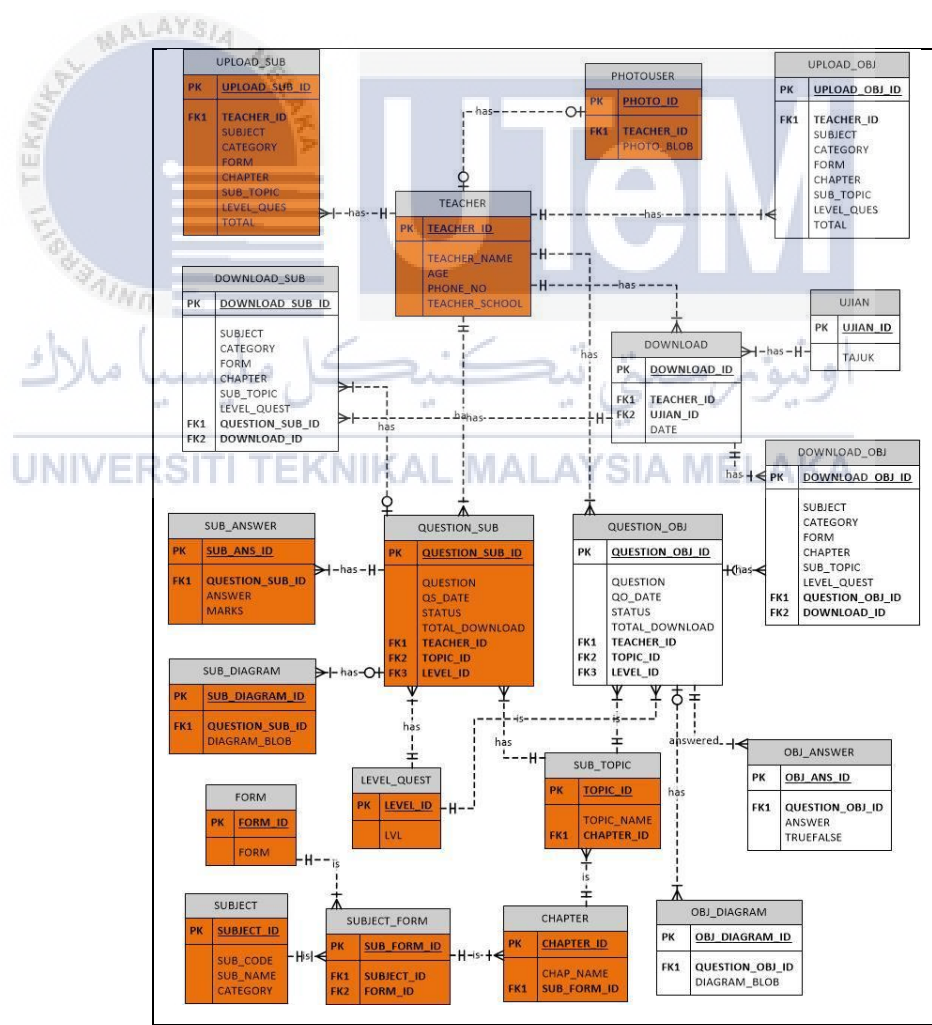
Figure 4.2: ERD i-Bank Question (iBQ-SQ)

4.3.1.2 Business Rules

A. Relationship between [TEACHER] and [UPLOAD_SUB]

☐ One teacher can upload many question

☐ One question is handled by one teacher.

B. Relationship between [TEACHER] and [PHOTOUSER]

☐ One teacher has one photo

☐ One photo for one teacher only.

C. Relationship between [TEACHER] and [QUESTION_SUB]

☐ Each question has to many answer

☐ Each answer is referred to one question.

D. Relationship between [QUESTION_SUB] and [SUB_ANSWER]

□ Each question has to many answer

□ Each answer is referred to one question.

E. Relationship between [QUESTION_SUB] and [SUB_DIAGRAM]
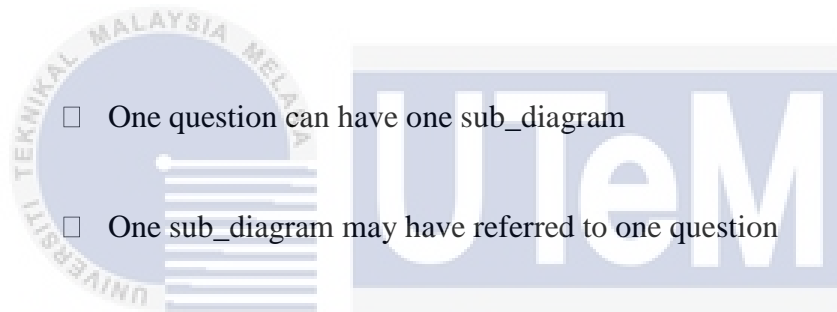
□ One question can have one sub_diagram

□ One sub_diagram may have referred to one question

F. Relationship between [LEVEL_QUEST] and [QUESTION_SUB]

□ One level question can have many question

□ One question referred to one level question

G. Relationship between [SUBTPOPIC] and [QUESTION_SUB]

☐ One subtopic can have many question

☐ One question referred to one subtopic

H. Relationship between [FORM] and [SUBJECT_FORM]
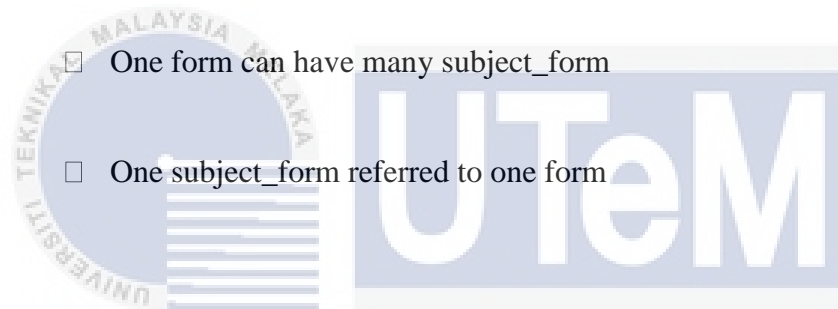
☐ One form can have many subject_form

☐ One subject_form referred to one form

I. Relationship between [CHAPTER] and [SUB_TOPIC]

☐ One chapter can have many sub_topic

☐ One sub_topic referred to one chapter

J. Relationship between [SUBJECT_FORM] and [CHAPTER]

☐ One subject_form can have many chapter

☐ One chapter referred to subject_form

4.3.2   Logical Design

4.3.2.1 Data Dictionary

Data dictionary are the transformation of ERD into the design. It is the basic things to organize the database. It consists of table name, table column, data type, primary key, foreign key and other requirement that are needed. Please refer to **APPENDIX A**.

4.3.2.2 Query Design

In this logical design, iBQ-SQ will query the data question by using join queries and sub queries in the form of SQL statements. This type of queries is used to show an only selected data to be show to user. The queries are implemented in the procedure statement. In order to select more than one column to be execute, cursor is used. So that, data from different table can be view by user. Below is some example of query statement that can get through the database.

Table 4.2: Example of procedure

```
create or replace PROCEDURE viewquestion2(
        view_teacher_id IN QUESTION_SUB.teacher_id%TYPE,
                ref_cursor_view OUT SYS_REFCURSOR
                )
as
BEGIN
```

```
OPEN ref_cursor_view FOR


Select    q.QUESTION_SUB_ID,q.question,    l.lvl,    st.topic_name,    c.chap_name,
s.sub_name, s.category,
 f.form, q.status,q.total_download,to_char(q.qs_date,'DD/Mon/YYYY') as tdate
from question_sub q,level_quest l, sub_topic st, chapter c, subject_form sf, subject s,
form f
where l.level_id =q.level_id
and st.topic_id = q.topic_id
and st.chapter_id = c.chapter_id
and c.sub_form_id = sf.sub_form_id
and sf.subject_id = s.subject_id
and sf.form_id = f.form_id
and  q.teacher_id=view_teacher_id
ORDER BY to_char(q.qs_date,'Mon') desc,to_char(q.qs_date,'DD') DESC;
  COMMIT;


END;
```


Figure 4.3: Question record

4.3.3 Physical Design

4.3.3.1 Selection of DBMS

To develop this system, Oracle 9i are chosen as the DBMS. Oracle can query the data faster than other database even for complex query. However other database cannot do the complex query. This is because, when it reaches the limit of complex query, it may not support or maybe some function is not provided.

4.3.3.2 The Usage of Store Procedure/Trigger

For this system, there are some trigger are used in order to manage the data more easily and logically. For example, when there are some data change, it will trigger other table data will to change. Trigger also are used to smooth the database management, some store procedure is used in terms of function and procedure.

Table 4.3: Example of procedure and trigger

Store procedure	Trigger
loginArep -use to verify user_id and password from database	**Trig_before_Answer** **Trig_before_diagram** **Trig_before_userphoto** **-**Use for generate primary key

InsertAnswer **InsertQuestion** **DeleteQuestion** **UpdateQuestion** -use to insert delete or update data in database	**Trig_before_question_sub** **-**use before insert,update and delete question If insert it will generate primary key If update Date question will update If delete Data in child table will delete first
Viewquestion2 **Viewquestion3** **viewquestionbyid** -use to retrieve data from database to view	**Trig_after_question_sub** -use after insert, update and delete question If insert Data also will insert into table upload If update Data in table upload will change If delete Data in table upload will change
Searcharif2 -use to search data from database	
Refcursorank -use to determine rank from database	

4.3.3.3   Security Mechanism

Table 4.4: Security Mechanism

Login	Need to refill both column for user id and password. If not, teacher cannot enter the system.
	If the user id or username not match, pop-up massage will be prompt out.

4.4   Graphical User Interface (GUI) Design

This system is developing with user friendly user interface design to easier user to understand the system flow. The purpose of the user interface design is to make sure that the user easy to use the system without facing the coding environment. The interface design is arranging properly in order the user use the system easily. The input design of the iBQ-SQ can be refer in the Figure below.

Figure 4.4: User Interface – Application Home page

Figure 4.5: User Interface – Application Login

Figure 4.6: User Interface – Main Page

Figure 4.7: User Interface – Question

Figure 4.8: User Interface –add Question

Figure 4.9: User Interface –Other question
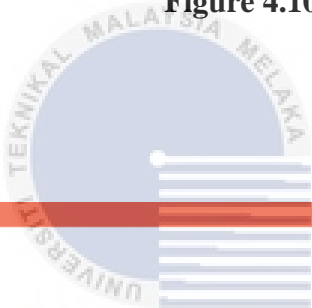
Figure 4.10: User Interface – search question

Figure 4.11: User Interface – Rank Uploader

Figure 4.12: User Interface – Profile teacher

4.4.1 Navigation Design

Navigation design is the design flow and form that involve in the i-Bank Question: Subjective Question Paper (iBQ-SQ). Below is the overview of navigation design for iBQ-SQ.
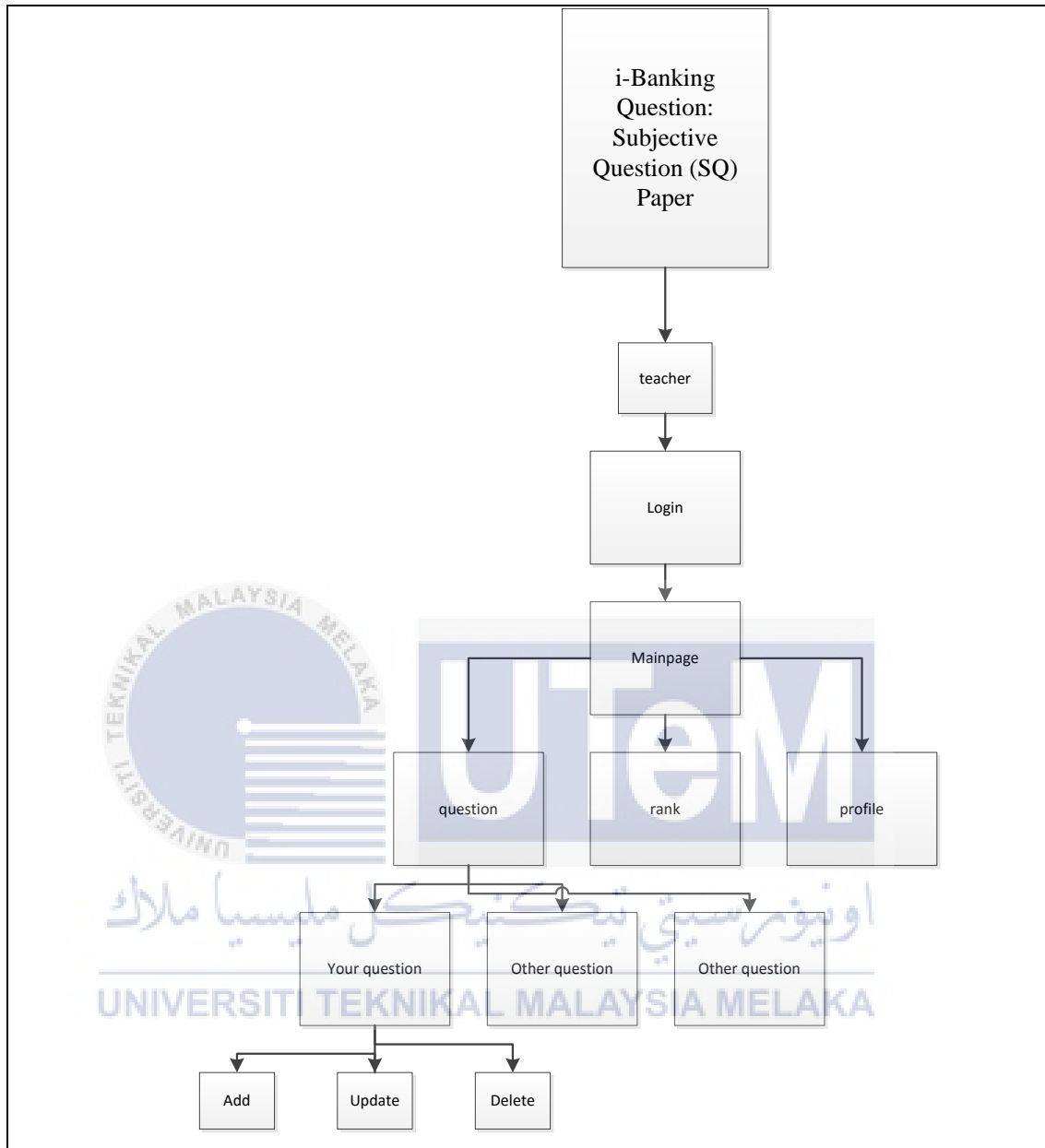
Figure 4.13: Navigation Design i-Bank Question (iBQ-SQ)

4.4.2 Input Design

In iBQ-SQ, input design capture only necessary data. Unnecessary data are avoided in order to get the constant and useful data. This input design contains of the design of user interface.

Table 4.4: Input Design for Add question

Form	Input Component	Type Validation Rules
Question	Question	300 character
	diagram	Blob 2172 kb
	status	30
	level	30
	subject	dropdown
	chapter	dropdown
	subtopic	dropdown
	answer	300 character

Figure 4.14: Input Design add question

Table 4.6: Input Design for search question

Form	Input Component	Type Validation Rules
Search question	subject	300 character
	chapter	30 character
	topic	30 character
	category	dropdown
	form	dropdown
	level	dropdown

Figure 4.15: Input Design for search question

4.4.3 Output Design

As we know, output is a term that show an information either display or printed by the information system. To get an output, there are some method required to be select for present the information. These methods are used to identify the specific output required to meet the information requirements.

Table 4.7: Output Design for i-Bank Question (iBQ-SQ)

Form	Description	
Rank	input	Key in month
	output	Rank uploader
Search question	input	Key in question detail
	output	question

Figure 4.16: View Ranking Uploader

Figure 4.17: View Question after search

4.5   Conclusion

In this chapter, it was discussing on the design that are involve to complete i-Bank Question (iBQ-SQ). System architecture design for this system is about the structure that facilitates the database to complete a transaction. While in the database design, it was divided into three part that is conceptual design that consist of ERD and business rule. Next is logical design that explain about data dictionary of the iBQ-SQ and query involved. Lastly is physical design describing more detail about the DMBS, database object that been used, security mechanism and database contingency. Moreover, there also the Graphical User Interface (GUI) design that explain and show the flow of the system. Next chapter will be the implementation for the iBQ-SQ. It will be the system development environment setup and the database implementation that will be discuss in more detail.

CHAPTER V

IMPLEMENTATION

5.1   Introduction

A complete system must be implement rightly with their requirement. This chapter will be discussing on how this project are implement from the beginning to the end. All this will be explaining in the system development environment setup which is the beginning phase of implementation. Meanwhile, the middle and end process will be explained in the database implementation. In order to produce a good system, all the implementation phase is completed in the best way.

5.2   System Development Environment Setup

the i-Banking Question: Subjective Question (IBQ-SQ) Paper are using java IEE application and its run Eclipse. Eclipse are server that act as the local host. Oracle 9i Edition are also used as the database server. All the data entered by user are stored in this database. The user interface is design using Adobe Dreamweaver CS6.

5.2.1   Installation step

In this installation, it will explain step by step on how Apache Tomcat are installed. Please refer to **APPENDIX B.**

5.2.2   Database and database objects creation

Database for i-Banking Question: Subjective Question (SQ) Paper are create in the **B031310227** database. All the entity in the ERD are create in the database as the single table. All that table are assign with primary key (PK) as their unique key. Not just that, any PK that contains in other are assign as foreign key (FK). All the table are create and using SQL statement.

5.3   Database Implementation

Database implementation will explain about DDL/DCL statement and how the implementation process on the main process base on stored procedure and trigger by using the selected programming language.

5.3.1 DDL/DCL statement (schema level)

Data definition language (DDL) is a syntax that are similar to a computer programming language in order to define the data structure especially data schema. While data control language (DCL) is used to create and destroy databases and database objects. These command are usually use by administration during setup and removal phase of database project.

CREATE STATEMENT

i. Create an empty database name "**B031310227**" on DBMS.
    Command: **CREATE DATABASE B031310227**

ii. Create table based on the entity in the ERD.
    Example command:

```
CREATE TABLE TEACHER (
    TEACHER_ID varchar2 (30) primary key NOT NULL,
    TEACHER_NAME varchar2(30),
    TEACHER_SCHOOL varchar2(30),
    PHONE_NO varchar2(30),
    AGE varchar2(30)
);
```

Figure 5.1 Create table Teacher

```
CREATE TABLE LEVEL_QUEST (
    LEVEL_ID varchar2(30) primary key NOT NULL,
    LVL varchar2(30)
);
```

Figure 5.2 Create table Level_Quest

```
CREATE TABLE SUBJECT (
    SUBJECT_ID varchar2(30) primary key NOT NULL,
    SUB_NAME varchar2(30),
    SUB_CODE varchar2(30),
    CATEGORY varchar2(30)
);
```

Figure 5.3 Create table Subject

```
CREATE TABLE FORM(
    FORM_ID varchar2(30) primary key NOT NULL,
    FORM varchar2(30)
);
```
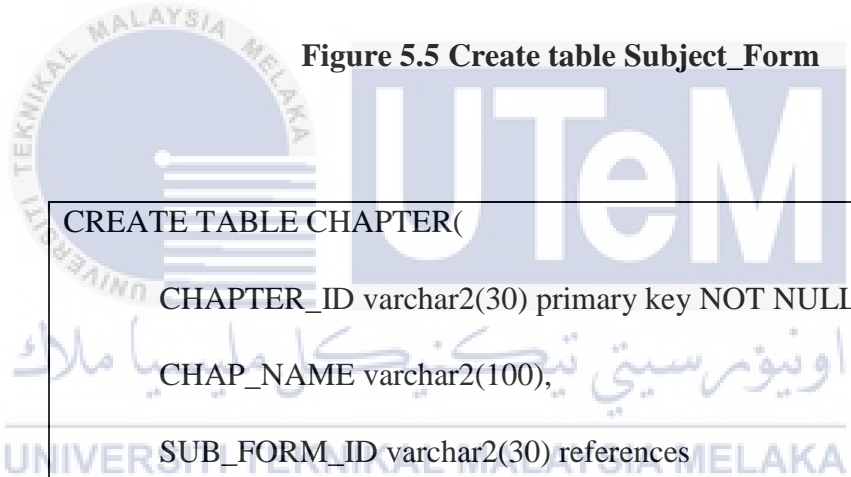
Figure 5.4 Create table Form

```
CREATE TABLE SUBJECT_FORM (

        SUB_FORM_ID varchar2(30) primary key NOT NULL,

        SUBJECT_ID varchar2(30) references
SUBJECT(SUBJECT_ID),

        FORM_ID varchar2(30) references FORM(FORM_ID)

);
```

Figure 5.5 Create table Subject_Form

```
CREATE TABLE CHAPTER(

        CHAPTER_ID varchar2(30) primary key NOT NULL,

        CHAP_NAME varchar2(100),

        SUB_FORM_ID varchar2(30) references

SUBJECT_FORM(SUB_FORM_ID)

);
```

Figure 5.6 Create table Chapter

```
CREATE TABLE SUB_TOPIC(

        TOPIC_ID varchar2(30) primary key NOT NULL,

        TOPIC_NAME varchar2(30),

        CHAPTER_ID varchar2(30) references
CHAPTER(CHAPTER_ID)

);
```

Figure 5.7 Create table Sub_Topic

```
CREATE TABLE QUESTION_SUB(
    QUESTION_SUB_ID  varchar2(30)  primary  key  NOT
NULL,
    QUESTION varchar2(300),
    TOTAL_DOWNLOAD INT,
    STATUS varchar2(30),
    QS_DATE DATE,
    TEACHER_ID          varchar2(30)          references
TEACHER(TEACHER_ID),
    LEVEL_ID          varchar2(30)          references
LEVEL_QUEST(LEVEL_ID),
    TOPIC_ID          varchar2(30)          references
SUB_TOPIC(TOPIC_ID)
);
```

Figure 5.8 Create table Question_Sub

```
CREATE TABLE SUB_ANSWER(
    SUB_ANS_ID varchar2(30) primary key NOT NULL,
    ANSWER varchar2(100),
    MARK NUMBER(5,2),
    QUESTION_SUB_ID varchar2(30) references
    QUESTION_SUB(QUESTION_SUB_ID)
);
```

Figure 5.9 Create table Sub_Answer

```
CREATE TABLE PHOTOUSER(

PHOTO_ID   varchar2(30) primary key NOT NULL,

TEACHER_ID varchar2(30) references TEACHER(TEACHER_ID),

PHOTO_BLOB   BLOB

);
```

Figure 5.10 Create table Photouser

```
CREATE TABLE SUB_DIAGRAM(

SUB_DIAGRAM_ID  varchar2(30) primary key NOT NULL,

QUESTION_SUB_ID varchar2(30) references

QUESTION_SUB(QUESTION_SUB_ID)

DIAGRAM_BLOB   BLOB,

);
```

Figure 5.11 Create table Sub_Diagram

DELETE statement

Delete statement is use to delete a row or all data in the table. To delete the data, it must meet the 'where' condition then it will be deleted.

DELETE FROM Question_Sub WHERE QUESTION_SUB_ID = 'QS105';

Figure 5.12: Example delete data for table Question

UPDATE statement

Update statements is used to modify the values or the data of a row in the table. However, there is the limit using the update statement foe view.

UPDATE QUESTION_SUB SET STATUS = 'ACTIVE' WHERE
QUESTION_SUB_ID = 'QS101';

Figure 5.13: Example update data for table Question_Sub

ALTER statement

Alter statement is used to modify the existing database object in the database. It change the properties of an object in the DBMS. However, it depends on type of DBMS used in develop the system.

ALTER TABLE Question_Sub ADD (Total_Download number);

Figure 5.14: Example alter data for table Question_Sub

DROP statement

Drop statement is use to destroy the existing database objects like table, index or view. In other words, it removes an objects from relational database management system (RDBMS). Most RDBMS support the dropping of tables, users and the database.

DROP TABLE UPLOAD_SUB;

Figure 5.15: Example drop table for table Upload_Sub

B. Implementation main process

The main and all process of the system are using store procedure either for insert, add, update, view or delete. The used of store procedure are to secure the source code from being known by other user. Meanwhile, trigger are also use but only in the database either trigger before or after. It is created in the database. The trigger created are fired when it meet the trigger condition.

Stored procedured

```
create or replace PROCEDURE viewquestion2(
        view_teacher_id IN
QUESTION_SUB.teacher_id%TYPE,
            ref_cursor_view OUT
SYS_REFCURSOR
            )
as
BEGIN
OPEN ref_cursor_view FOR
Select q.QUESTION_SUB_ID,q.question, l.lvl,
st.topic_name, c.chap_name, s.sub_name, s.category,
f.form,
q.status,q.total_download,to_char(q.qs_date,'DD/Mon/
YYYY') as tdate
from question_sub q,level_quest l, sub_topic st,
chapter c, subject_form sf, subject s, form f
where l.level_id =q.level_id
and st.topic_id = q.topic_id
and st.chapter_id = c.chapter_id
and c.sub_form_id = sf.sub_form_id
and sf.subject_id = s.subject_id
and sf.form_id = f.form_id
and  q.teacher_id=view_teacher_id
ORDER BY to_char(q.qs_date,'Mon')
desc,to_char(q.qs_date,'DD') DESC;
 COMMIT;
END;
```

Figure 5.16: Store procedure for query data question

Stored procedure

```
                    CREATE OR REPLACE PROCEDURE insertquestion(
                        question IN QUESTION_SUB.question%TYPE,
                        status IN QUESTION_SUB.status%TYPE,
                        teacher_id IN QUESTION_SUB.teacher_id%TYPE,
                        level_id IN QUESTION_SUB.level_id%TYPE,
                        topic_id IN QUESTION_SUB.topic_id%TYPE
                    )




                    IS
                    BEGIN

                      INSERT INTO QUESTION_SUB
                    (question,status,teacher_id,level_id,topic_id)
                      VALUES (question,status,teacher_id,level_id,topic_id);


                    COMMIT;


                    END;
                    /
```

Figure 5.17: Store procedure for query upload question

Stored procedure

```
CREATE OR REPLACE PROCEDURE updatequestion(
    up_QUESTION_SUB_ID IN
QUESTION_SUB.QUESTION_SUB_ID%TYPE,
    up_question IN
QUESTION_SUB.question%TYPE,
    up_status IN QUESTION_SUB.status%TYPE
)
IS
BEGIN

    UPDATE QUESTION_SUB SET
    question = up_question,
    status = up_status,
    qs_date = sysdate

    where QUESTION_SUB_ID =
up_QUESTION_SUB_ID;

    COMMIT;

END;
/
```

Figure 5.18: Store procedure for query Update question

Stored procedure

```
CREATE OR REPLACE PROCEDURE
deletequestion(p_question_sub_id IN
question_sub.question_sub_id%TYPE)
IS
BEGIN


  DELETE question_sub where question_sub_id =
p_question_sub_id;


COMMIT;


END;
/
```

Figure 5.19: Store procedure for query delete question

Trigger Before insert,update and delete

```
create or replace TRIGGER arif_Before_Question_sub
BEFORE INSERT OR UPDATE OR DELETE ON QUESTION_sub
FOR EACH ROW
BEGIN
        IF INSERTING THEN
        select'QS'||ques_seq.nextval into
:new.QUESTION_SUB_ID from dual;
:new.qs_date :=sysdate;
        ELSIF UPDATING THEN


                :NEW.qs_date := SYSDATE;


        ELSIF DELETING THEN


                DELETE FROM sub_ANSWER
                    WHERE QUESTION_sub_ID =
:OLD.QUESTION_sub_ID;


    DELETE FROM sub_diagram
                    WHERE QUESTION_sub_ID =
:OLD.QUESTION_sub_ID;


    DELETE FROM download_sub
                    WHERE QUESTION_sub_ID =
:OLD.QUESTION_sub_ID;
        END IF;


END;
```

Figure 5.20: Trigger before insert,update and delete

Trigger after insert,update and delete

```
create or replace trigger after_ques_sub

AFTER INSERT OR UPDATE OR DELETE ON question_sub

for each row

declare


plevel varchar2(30);

ptopic varchar2(30);

pteacher varchar2(30);



levelq varchar2(30);

subt varchar2(30);

chap varchar2(100);

subj varchar2(30);

cate varchar2(30);

u_form varchar2(30);


v_counter number;

v_total number;


begin


IF INSERTING THEN


        plevel := :new.level_id;

        ptopic := :new.topic_id;

        pteacher := :new.teacher_id;


        select count(*)
```

```
        into v_counter
        from level_quest l, sub_topic st, chapter c, subject_form sf, subject s, form
f, upload_sub u
        where l.level_id = plevel
        and st.topic_id = ptopic
        and lvl = u.level_quest
        and st.topic_name = u.sub_topic
        and st.chapter_id = c.chapter_id
        and c.sub_form_id = sf.sub_form_id
        and sf.subject_id = s.subject_id
        and sf.form_id = f.form_id
        and u.teacher_id = pteacher;


        select l.lvl, st.topic_name, c.chap_name, s.sub_name, s.category, f.form
        into levelq, subt, chap, subj, cate, u_form
        from level_quest l, sub_topic st, chapter c, subject_form sf, subject s, form
f
        where l.level_id = plevel
        and st.topic_id = ptopic
        and st.chapter_id = c.chapter_id
        and c.sub_form_id = sf.sub_form_id
        and sf.subject_id = s.subject_id
        and sf.form_id = f.form_id;


        IF :NEW.STATUS = 'Active' THEN


            IF (V_COUNTER = 0) THEN
                    insert into upload_sub
(LEVEL_QUEST,SUB_TOPIC,CHAPTER,SUBJECT,FORM,CATEGORY,TEA
CHER_ID) values (levelq,subt,chap,subj,u_form,cate,pteacher);
```

```
                    else
                            update upload_sub
                            set total=total+1
                            where level_quest=levelq
                            and sub_topic=subt
                            and chapter=chap
                            and subject=subj
                            and category=cate
                            and form=u_form
                            and teacher_id=pteacher;

            end if;
        end if;


ELSIF UPDATING THEN

    plevel := :new.level_id;
        ptopic := :new.topic_id;
        pteacher := :new.teacher_id;


        select count(*)
        into v_counter
        from level_quest l, sub_topic st, chapter c, subject_form sf, subject s, form
f, upload_sub u
        where l.level_id = plevel
        and st.topic_id = ptopic
        and lvl = u.level_quest
        and st.topic_name = u.sub_topic
        and st.chapter_id = c.chapter_id
        and c.sub_form_id = sf.sub_form_id
```

```
            and sf.subject_id = s.subject_id
            and sf.form_id = f.form_id
            and u.teacher_id = pteacher;


            select l.lvl, st.topic_name, c.chap_name, s.sub_name, s.category, f.form
            into levelq, subt, chap, subj, cate, u_form
            from level_quest l, sub_topic st, chapter c, subject_form sf, subject s, form

f
            where l.level_id = plevel
            and st.topic_id = ptopic
            and st.chapter_id = c.chapter_id
            and c.sub_form_id = sf.sub_form_id
            and sf.subject_id = s.subject_id
            and sf.form_id = f.form_id;


            IF :OLD.STATUS = 'Non Active' AND :NEW.STATUS = 'Active' THEN


                IF (V_COUNTER = 0) THEN


                        insert into upload_sub
(LEVEL_QUEST,SUB_TOPIC,CHAPTER,SUBJECT,FORM,CATEGORY,TEA
CHER_ID) values (levelq,subt,chap,subj,u_form,cate,pteacher);


                else
                        update upload_sub
                        set total=total+1
                        where level_quest=levelq
                        and sub_topic=subt
                        and chapter=chap
                        and subject=subj
                        and category=cate
```

```
                    and form=u_form

                    and teacher_id=pteacher;




            END IF;




    ELSIF :OLD.STATUS = 'Active' AND :NEW.STATUS = 'Non Active'
THEN
            SELECT TOTAL
            INTO V_TOTAL
            from upload_sub
            where level_quest=levelq
            and sub_topic=subt
            and chapter=chap
            and subject=subj
            and category=cate
            and form=u_form
            and teacher_id=pteacher;




            IF V_TOTAL = 1 THEN
                    DELETE FROM upload_sub

                    where level_quest=levelq
                    and sub_topic=subt
                    and chapter=chap
                    and subject=subj
                    and category=cate
                    and form=u_form
```

```
                                and teacher_id=:OLD.TEACHER_ID;


                ELSE

                        update upload_sub

                        set total=total-1

                        where level_quest=levelq

                        and sub_topic=subt

                        and chapter=chap

                        and subject=subj

                        and category=cate

                        and form=u_form

                        and teacher_id=:OLD.TEACHER_ID;



                END IF;


        END IF;

ELSIF DELETING THEN




   plevel := :old.level_id;

        ptopic := :old.topic_id;

        pteacher := :old.teacher_id;



        select count(*)

        into v_counter

        from level_quest l, sub_topic st, chapter c, subject_form sf, subject s, form
f, upload_sub u

        where l.level_id = plevel

        and st.topic_id = ptopic

        and lvl = u.level_quest
```

```
        and st.topic_name = u.sub_topic

        and st.chapter_id = c.chapter_id

        and c.sub_form_id = sf.sub_form_id

        and sf.subject_id = s.subject_id

        and sf.form_id = f.form_id

        and u.teacher_id = pteacher;


        select l.lvl, st.topic_name, c.chap_name, s.sub_name, s.category, f.form

        into levelq, subt, chap, subj, cate, u_form

        from level_quest l, sub_topic st, chapter c, subject_form sf, subject s, form

f

        where l.level_id = plevel

        and st.topic_id = ptopic

        and st.chapter_id = c.chapter_id

        and c.sub_form_id = sf.sub_form_id

        and sf.subject_id = s.subject_id

        and sf.form_id = f.form_id;


        SELECT TOTAL

        INTO V_TOTAL

        from upload_sub

        where level_quest=levelq

        and sub_topic=subt

        and chapter=chap

        and subject=subj

        and category=cate

        and form=u_form

        and teacher_id=:OLD.TEACHER_ID;


        IF V_TOTAL = 1 THEN

                DELETE FROM upload_sub
```

```
                        where level_quest=levelq
                        and sub_topic=subt
                        and chapter=chap
                        and subject=subj
                        and category=cate
                        and form=u_form
                        and teacher_id=:OLD.TEACHER_ID;


              else

                        update upload_sub
                        set total=total-1
                        where level_quest=levelq
                        and sub_topic=subt
                        and chapter=chap
                        and subject=subj
                        and category=cate
                        and form=u_form
                        and teacher_id=:OLD.TEACHER_ID;

              END IF;


END IF;


end;
```

Figure 5.21: Trigger after insert,update and delete

5.4   Conclusion

This chapter discusses on the activity involve in the implementation phase that use to start up creating the system. It includes the installation software that are used and how to create the database and database objects. For installation step, it was discussed in system development environment setup. While DDL/DCL statement in the database implementation. The next chapter will be discussing on how to carry out the testing plan. In that phase, the overall system will be tested base on the module that are develop.

CHAPTER VI

TESTING

6.1 Introduction

This chapter is about testing phase for i-Bank Question: Subjective Question (iBQ-SQ) Paper. The purpose of testing is to test the capability of the system either it meets all the requirements. It is important to avoid any failures that occurs in the system later. Test plan are used for the testing to carry out. In test plan it included test environment, test organization and test schedule. For the test design, it will explain more on test data and test description for the system. At the end, based on the test result and analysis, it will determine either the system is success or failed.

6.2   Test Plan

Test plan is used in the beginning. It will test on organization, environment and schedule. All this are a way that are taken to test i-Bank Question: Subjective Question (SQ) Paper that are going to be use by user.

6.2.1   Test Organization

Test organization is a people that are responsibility to test the system during the testing process. This people are come from different background. They are Student, System Developer and Project Supervisor. Project Supervisor is the individual who is responsible to supervise the project of system developer works. While student act as the end user of the system. After done test the system, all of them must give the feedback. All these feedbacks can be used as a guide to improve the system.

Table 6.1: Test Organization for iBQ-SQ

Tester ID	Title/Post	Responsibility
Tester 1	System Developer	Responsible to develop, manage, test and documenting the system. Make sure that the system followed the requirement stated and successful completed
Tester 2	Project Supervisor	Responsible to test the system and give the feedback to the system
Tester 3	Student	Responsible to test the system and give the feedback to the system developer.

6.2.2 Test Environment

Test environment is the environment that the system developer uses to develop and maintain the programs. To facilitate the testing process, an optimal environment need to be setup base on the following specification.

Table 6.2: Test Environment for iBQ-SQ

Variable	Requirement
Hardware	Processor : Intel inside core i5 Ram :4gb Hard Disk space :300MB above
Software	Adobe Dreamweaver CS6
Workstation	Microsoft Windows Network
Database	Oracle Database 9i
Training And Preparation	Assigned tester with provided with user manual.

6.2.3 Test Schedule

Table 6.3: Test Schedule

Module Component	Activity	Duration	Start	End
System Login	• System testing • User acceptance	2 days / 3 times	10/07/2016	12/07/2016
Add question	• System testing • User acceptance	4 days / 3 times	12/07/2016	16/07/2016
Searching	• System testing • User acceptance	4 days / 3 times	16/07/2016	20/07/2016
Ranking	• System testing • User acceptance	2 days / 3 times	20/07/2016	22/07/2016

6.3   Test Strategy

Test strategy is the testing that are used in software development cycles. To test for i-Bank Question: Subjective Question (iBQ-SQ) Paper white-box and black-box test are choose.

Testing using Black-box must be conducted based on the functionality testing and requirement in i-Bank Question: Subjective Question (iBQ-SQ) Paper. It's consist of negative testing, positive testing and error guessing. For error guessing, it used to notify if the user enters the valid input by displaying the error message. Positive testing is to test either the system not consisted with the requirements stated or get the expected result. While negative testing is to determine unexpected action that might occur in the system.

White-box testing is the method of testing the software for the working in the application or internal structure, as opposed to its functionality. Normally, it examines the source code that are used to develop the test cases. Through this method, it can uncover errors or the problems because this system might have the potential to miss the unimplemented parts of the specification or missing some requirements stated.

6.3.1 Classes of tests

There are several classes of test that have been carried out. It is divided into security test, unit testing, system testing and user acceptance testing.

- Security Testing

Security testing is a process intended to reveal flaws in the security mechanism of an information system that protect data and maintain the functionality as intended. Actual security required the requirements tested depend on the security requirements that has been implemented by the system.in iBQ-SQ, the quality, reliability and the security are combining together. This testing is using white box testing that are tested by system developer.

- System Testing

System testing is used to make sure that the iBQ-SQ only accept the right input from user. If user enter the invalid input, the system will notify user with display the error message regarding the error. This test is using black box to identify the result for the positive testing, negative testing and error guessing. It is also tested using white box testing to uncover the problem that identified in the black box testing by examine through the source code. This system testing is tested by system developer.

- Unit Testing

  Unit testing are carried out to test the overall of the system to make sure it can be functioned well. This testing is using black box testing and consists of three testers that is system developer, project supervisor and student as the end user.

- User Acceptance Test

  User acceptance test is to identify either user can accept this system and attract user interest to use the system. This test is using black box test strategy and tested by project supervisor and student.

6.4 Test Design

Test design explain on the test description and test data. In test description, it consists of test case identification, test cases, and expected result for each module. While in test data, real life or synthetic data will be selected.

6.4.1 Test Description

i-Bank Question: Subjective Question Subjective Paper (iBQ-SQ) is the system that are develop to store question from teacher. All the data is stored in the database. Test cases are developing to carry out the test process. Table below shows the result of unit testing.

Table 6.4: Login Module

Test Case ID	Description	Testing Type	Expected Result
TC_01-1	Invalid user ID and invalid password	Unit testing	Back to login page
TC_01-2	User ID and blank password	Unit testing	Back to login page
TC_01-3	Valid user ID and password	Unit testing	Succesfully log on

Table 6.5: Add Question Module

Test Case ID	Description	Testing Type	Expected Result
TC_02-1	All fill blank	Unit testing	The record is not save in the database.
TC_02-2	Enter the invalid data or format in question field	Unit testing	The record is not save in the database.
TC_02-3	Enter valid data and format in each field	Unit testing	The record are save in database

Table 6.6: Test Data for Login

Column Name	TD_01-1	TD_01-2	TD_01-3
Test Case ID	TC_01-1	TC_01-2	TC_01-3
User ID	zAbedah	zAbedah	ZABEDAH
Password	T09		T09
Result Test Data	Back to login page	Back to login page	Go to main page

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Table 6.7: Test Data for Add Question

Column Name	TD_02-1	TD_02-2	TD_02-3
Test Case ID	TC_02-1	TC_02-2	TC_02-3
Question			What is Solid?
Diagram			
Status			Active
Level		Knowledge	Knowledge
Subject		Kimia	Kimia
Chapter		Molecule	Molecule
Subtopic		Gas	Solid
Answer		Solid	A complete Molecule
Mark		aaaaa	10
Result Test Data	Fill The Blank Space!	Fill The Blank Space!	Data successfully inserted into database.

6.5   Test Result and Analysis

All the test result is documented describe in the table as show below. Test case is the input to test the system. After you run tests, you can review your test results to see which tests successes and which test are failure. The success or failure when using the actual data for testing process can be the factor to measure the system either it can have worked efficiently or need to be fixed. Please refer table below to see the test result and analysis.

Table 6.8: Test result and analysis for login

Model Component Login			Result	
Test Case ID	Test Data ID	Description	PASS	FAIL
TC_01-1	TD_01-1	User ID and password didn't exist	√	
TC_01-2	TD_01-2	User ID and password didn't exist	√	
TC_01-3	TD_01-3	Valid user ID and password	√	

Table 6.9: Test result and analysis for login

Model Component Login		Result		
Test Case ID	**Test Data ID**	**Description**	**PASS**	**FAIL**
TC_01-1	TD_01-1	User ID and password didn't exist	√	
TC_01-2	TD_01-2	User ID and password didn't exist	√	
TC_01-3	TD_01-3	Valid user ID and password	√	

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Table 6.10: Test result and analysis for Add question

Model Component Login		Result		
Test Case ID	**Test Data ID**	**Description**	**PASS**	**FAIL**
TC_02-1	TD_02-1	All field blank.	√	
TC_02-2	TD_02-2	Invalid data or format in Form	√	
TC_02-3	TD_02-3	Valid data for each field.	√	

6.6 Conclusion

As the conclusion, this chapter that conduct the testing process are the most crucial part to be completed and develop for this system. This is because, developer need to test every single part of the system to know how the system will perform from several aspects. Many aspects need to be consider such as reliability, security of the system and the user efficiency to make sure that the system meets all the requirement. From the testing process that have been done, developer can have fixed any fault and problems that come up. Next chapter will cover on the conclusion of the overall system.

CHAPTER VII

CONCLUSION

7.1   Introduction

This chapter will discuss about the strengths and weaknesses in this project base on the observation and testing that are make. This weakness can be referring if there any person who want to upgrade the system to be better in the future. Other than that is proposition to improve this system to be high level system in order can be used for the long period and effective. The outsider suggestion also takes into consideration. This is because, all that suggestion is taken as the user view through the system and what user want when they use the system. Finally, is the contribution of this project to the university or individual that will be used this system either it brings a lot of goodness or badness.

7.2 Observation on Weaknesses and Strengths

After done completing the i-Bank Question: Subjective Question (iBQ-SQ) Paper, based on the observation that had been made through this system, there are some weaknesses and strengths for this project identified which is;

Table 7.1 Weaknesses and Strength iBQ-SQ system

Weaknesses	Strengths
1.This system does not have auto marking.	1.The system are user friendly and simple as it can attract people to use the system.
2.This system does not accept strange symbol for the question.	2.The system doesn't need user to key in many information. This is because the system is a plugin system.
2.This system does not accept strange symbol for the question notification for user that most uploader question for rewarding.	3.The system automatically arrange the chapter and subtopic for the user.
	4.The system automatically arrange the chapter and subtopic for the user.

7.3 Proposition for Improvement

i-Bank Question: Subjective Question (iBQ-SQ) Paper System can be improved better in order can be used for the long period of time The database can be backup using database logical backup. The data must be backup every day and only backup the new data by using differential backup (backup all changes since last full backup). However, full and incremental backup can be used because full back up all the data and incremental backup all changes since last full or incremental backup and it's doesn't much different with differential backup.

The last improvement that can be made are make the system to be online system and implement these system as the application on the smartphone. The reason why the system need to be online and become an application is to reduce time taken for teacher to upload question anytime. This is because teacher need to open computer for upload question. By using online system, it can be done less than one minute.

7.4 Contribution

This project contributes a lot to secondary school teacher in terms of reduce the use of paper, easier manage the student record and transform the manual system to the complete system. This system is built to easier the teacher manages the question that are uploaded. In other side, it can be seen as save time, papers and space. Not just that, it also helps teacher that have uploaded many question because it rewards the teachers.

7.5 Conclusion

As a conclusion, this project is completed according to the objective and project scope. The development process for this system are based on the project schedule and milestone that have been planned earlier. This system is make to be a user friendly and easier to be user either on user or admin side even though there some weaknesses need to be repair.

REFERENCES

Scott Burkett (2007). "SCALING YOUR TECHNOLOGY WITH YOUR BUSINESS." Retrieve on January 2007 from http://www.scottburkett.com/technology/scaling-your-technology-with-your-business-530.html

Ajax | Jason | java |
 http://www.programming-free.com/2013/03/ajax-fetch-data-from-database-in-jsp.html

How to create dynamic dropdown list
http://stackoverflow.com/questions/2896730/how-to-generate-dynamic-drop-down-lists-using-jquery-and-jsp

Contextual Label Classes
http://www.w3schools.com/bootstrap/tryit.asp?filename=trybs_labels2&stacked=h

Upload Image into Oracle Blob Image
http://www.srikanthtechnologies.com/blog/java/uploadimagetoblob.aspx

APPENDICES

APPENDIX A

Dictionary Table Teacher

Attribute Name	Data Type	Length	Key	Description
Teacher_Id	Varchar2	30	Pk	Teacher Id
Teacher_Name	Varchar2	30		Teacher Name
Teacher_School	Varchar2	30		Teacher School
Phone_No	Varchar2	30		Phone Number
Age	Number	30		Age

اونيۏمرسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Data Dictionary Table Upload_sub

Attribute Name	Data Type	Length	Key	Description
Upload_Sub_Id	Varchar2	30	PK	Upload_Id
Level_Quest	Varchar2	30		Level Question
Sub_Topic	Varchar2	30		Subtopic Question
Chapter	Varchar2	30		Chapter Question
Subject	Varchar2	30		Subject Question
Category	Varchar2	30		Category Question
Total	Varchar2	30		Total Upload Question
Teacher_Id	Varchar2	30	FK	Reference From Table Teacher

Data Dictionary table Photo_User

Attribute Name	Data Type	Length	Key	Description
Photo_Id	Varchar2	30	PK	Photo_Id
Teacher_Id	Varchar2	30	FK	Reference From Table Teacher
Sub_Topic	Blob			Photo Teacher

Data Dictionary table Question

Attribute Name	Data Type	Length	Key	Description
Question_Sub_Id	Varchar2	30	PK	Question Id
Question	Varchar2	30		Question
Total_Download	Number	38		Total Question Download
Status	Varchar2	30		Status Question
Qs_Date	Date	30		Date Question
Teacher_Id	Varchar2	30	FK	Reference From Table Teacher
Level_Id	Varchar2	30	FK	Reference From Table Level_Quest
Topic_Id	Varchar2	30	FK	Reference From Table Teacher

Data Dictionary table Sub_answer

Attribute Name	Data Type	Length	Key	Description
Sub_Ans_Id	Varchar2	30	PK	Photo_Id
Answer	Varchar2	30		Answer
Mark	Number	(5,2)		Photo Teacher
Question_Sub_Id	Varchar2	30	PK	Reference From Table Question_Sub

Data Dictionary Table Sub_diagram

Attribute Name	Data Type	Length	Key	Description
Sub_Diagram_Id	Varchar2	30	PK	Sub_Diagram_Id
Diagram_Blob	Blob			Question Diagram
Question_Sub_Id	Varchar2	30	PK	Reference From Table Question_Sub

Data Dictionary Table Sub_diagram

Attribute Name	Data Type	Length	Key	Description
Level_Id	Varchar2	30	PK	Level_Id
Lvl	Varchar2	30		Level Question

Data Dictionary Table Subtopic

Attribute Name	Data Type	Length	Key	Description
Topic_Id	Varchar2	30	PK	Topic_Id
Topic_Name	Varchar2	30		Topic_Name
Chapter_Id	Varchar2	30	FK	Reference From Table Chapter

Data Dictionary Table Form

Attribute Name	Data Type	Length	Key	Description
Form_Id	Varchar2	30	PK	Form_Id
Form_Name	Varchar2	30		Secondry School,Primary School

Data Dictionary Table Subject

Attribute Name	Data Type	Length	Key	Description
Sub_Form_Id	Varchar2	30	PK	Sub_Form_Id
Subject_Id	Varchar2	30	FK	Reference From Table Subject
Form_Id	Varchar2	30	FK	Reference From Table Form

Data Dictionary Subject_Form

Attribute Name	Data Type	Length	Key	Description
Sub_Form_Id	Varchar2	30	PK	Sub_Form_Id
Subject_Id	Varchar2	30	FK	Reference From Table Subject
Form_Id	Varchar2	30	FK	Reference From Table Form

Data Dictionary chapter

Attribute Name	Data Type	Length	Key	Description
Chapter_Id	Varchar2	30	PK	Sub_Form_Id
Chapter_Name	Varchar2	30	FK	Reference From Table Subject
Sub_Form_Id	Varchar2	30	FK	Reference From Table Form

APPENDIX B

Apache Tomcat server and Eclipse Java EE Configuration

STEP 1: Lunch the eclipse and select workspace location in file system.
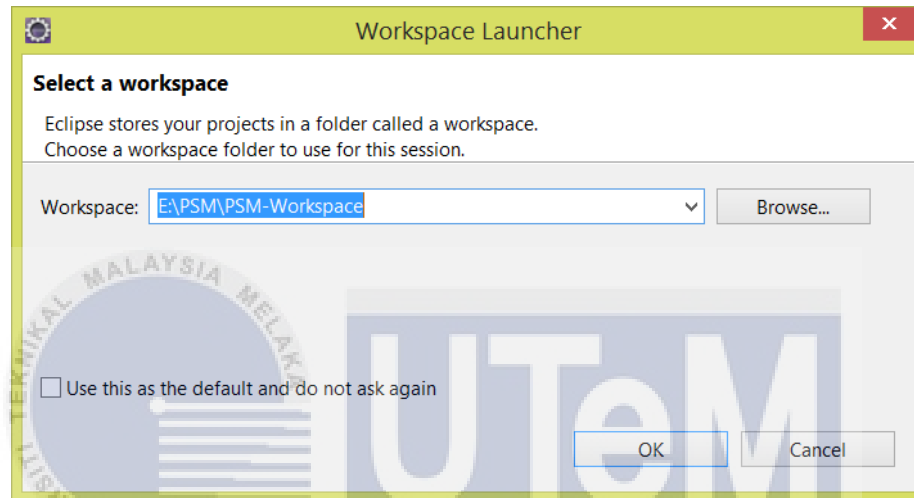
Server Installation Step 1

STEP 2: Click on File – New – Dynamic
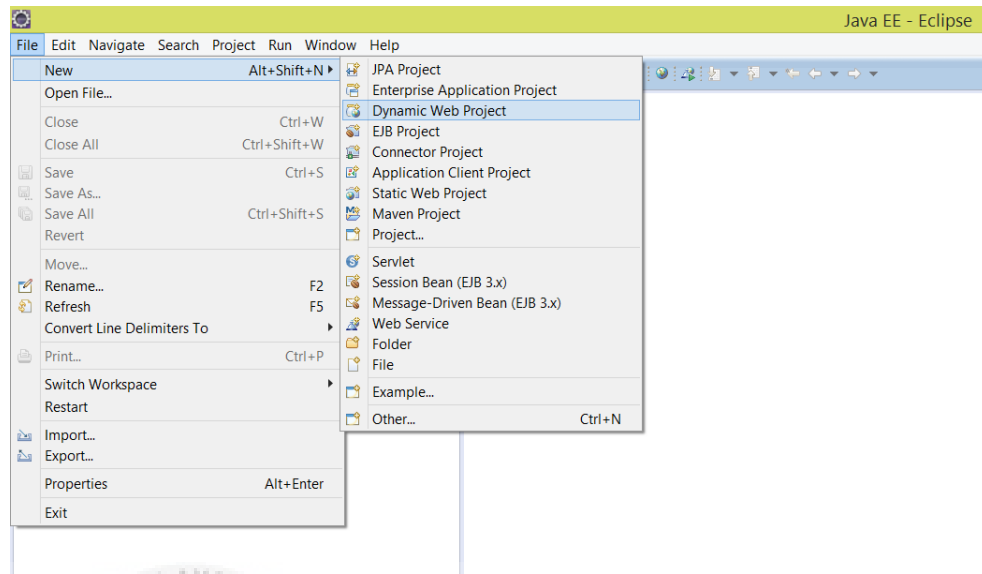
Server Installation Step 2

STEP 3: Choose Apache Tomcat v8.0 and click next button.
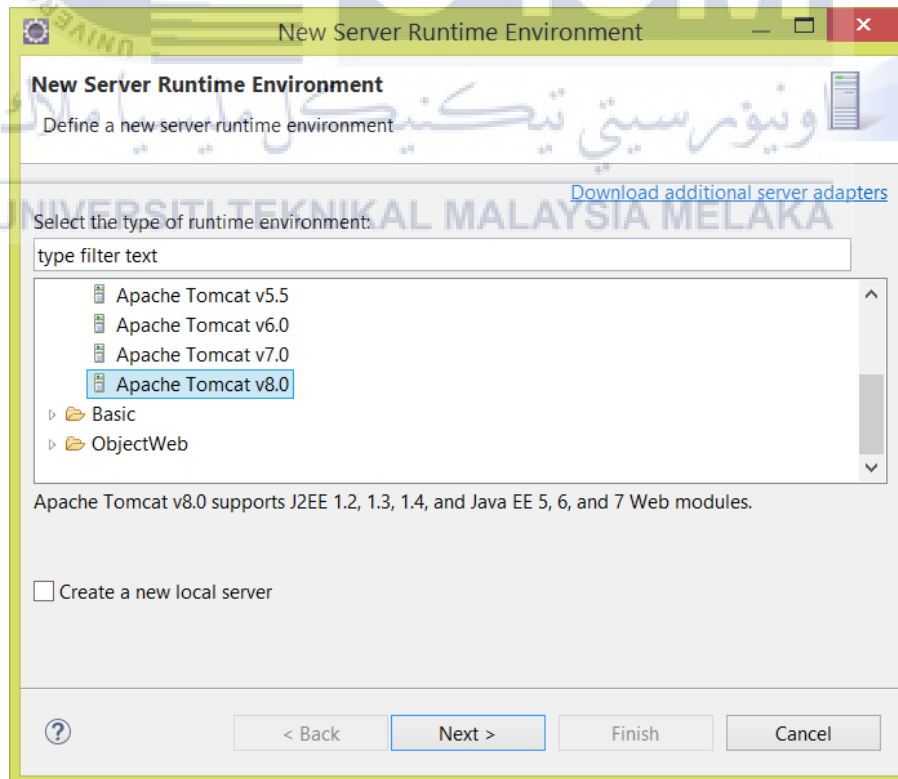
Server Installation Step 3

STEP 4: Find the apache-tomcat-8.0.27 and click on OK button.
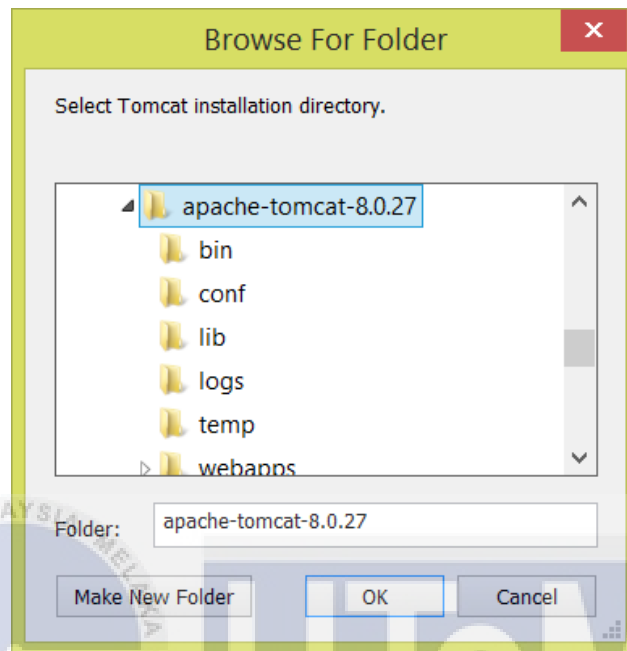
Server Installation Step 4

STEP 5: Click Finish Button
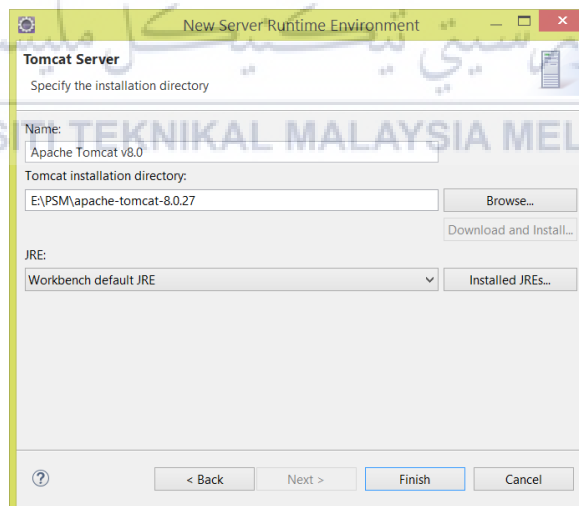
Server Installation Step 5

STEP 6: Enter the project name and tick Use default location and click on Next button

: Server Installation Step 6

STEP 7: Click on FInish Button
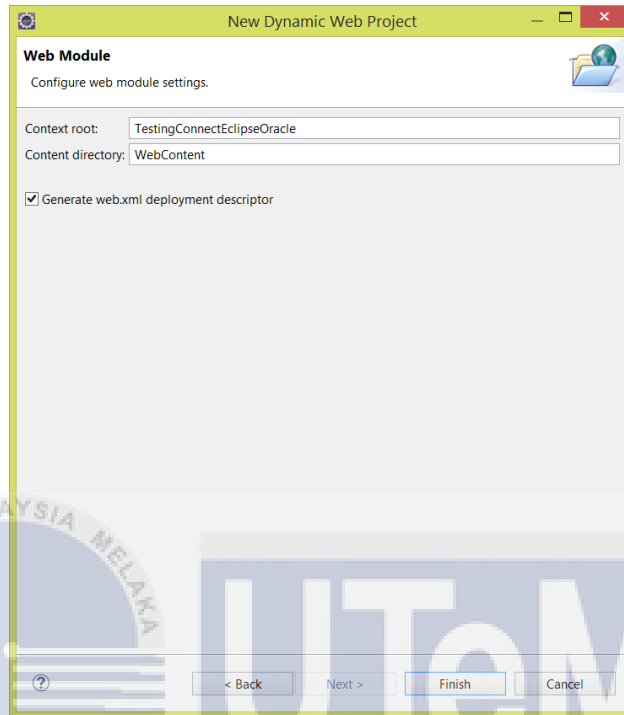
Server Installation Step 7

STEP 8: Click on Server and click the blue text to create server.
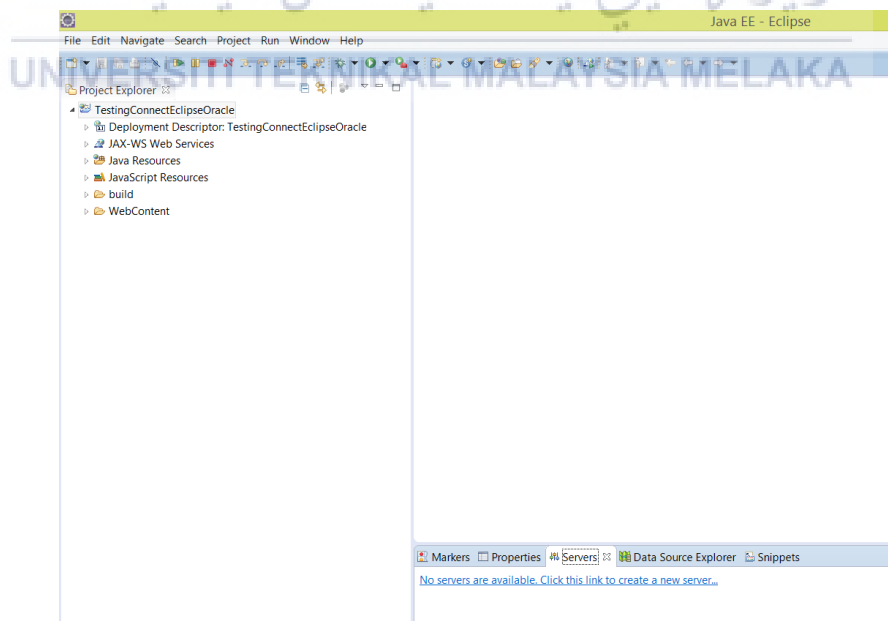
Server Installation Step 8

STEP 9: Enter the host name as localhost and Server runtime as Apache Tomcat v8.0 and click Next button
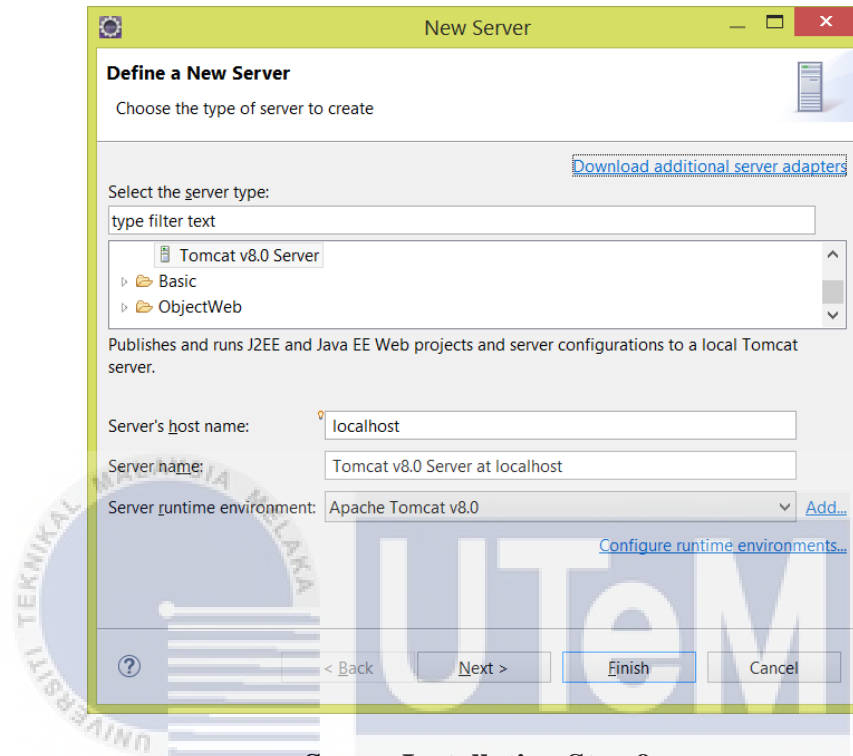
Server Installation Step 9

STEP 10: Click Add all Button and click Finish button.
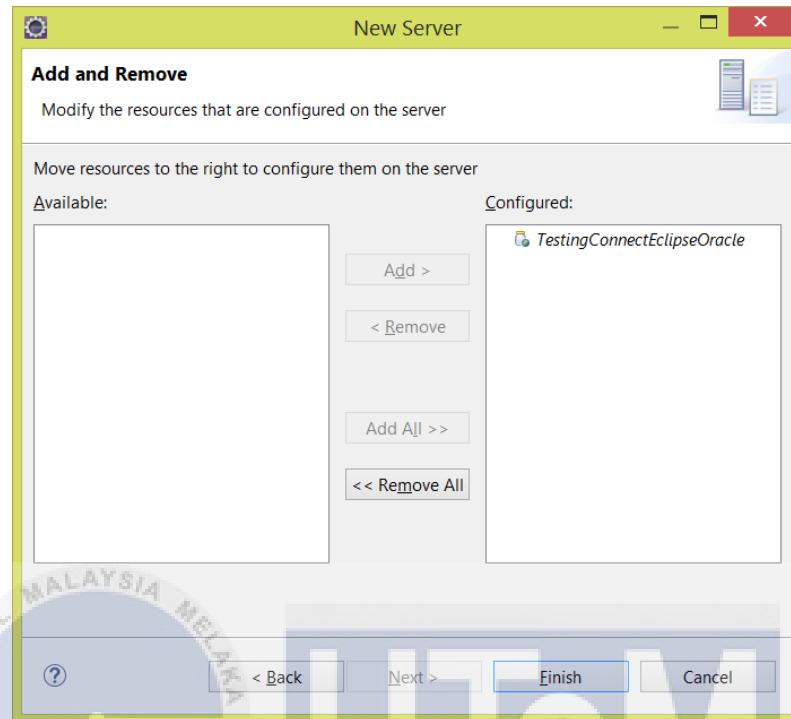
Server Installation Step 10

STEP 11: Click on server and click the green button to start the server.

Server Installation Step 11

STEP 12: Right click on the project and Run As – Run on Server
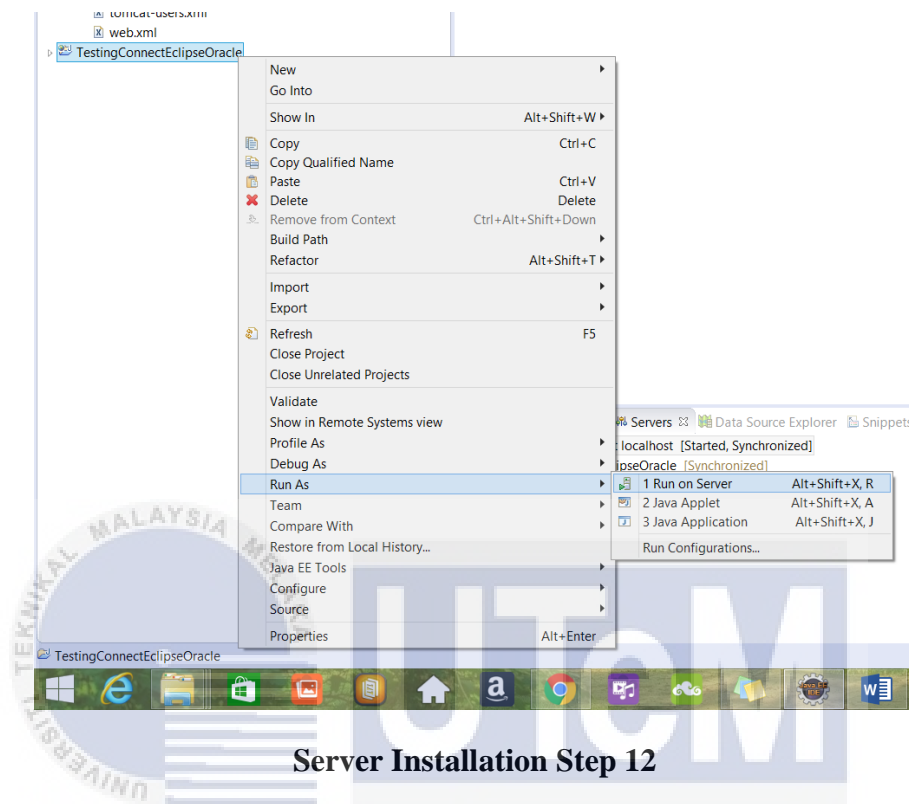
Server Installation Step 12

STEP 13: Open any web browser and go to localhost:8080 to test the server.

Server Installation Step 13

STEP 14: Right click the project and go to New- SQL file
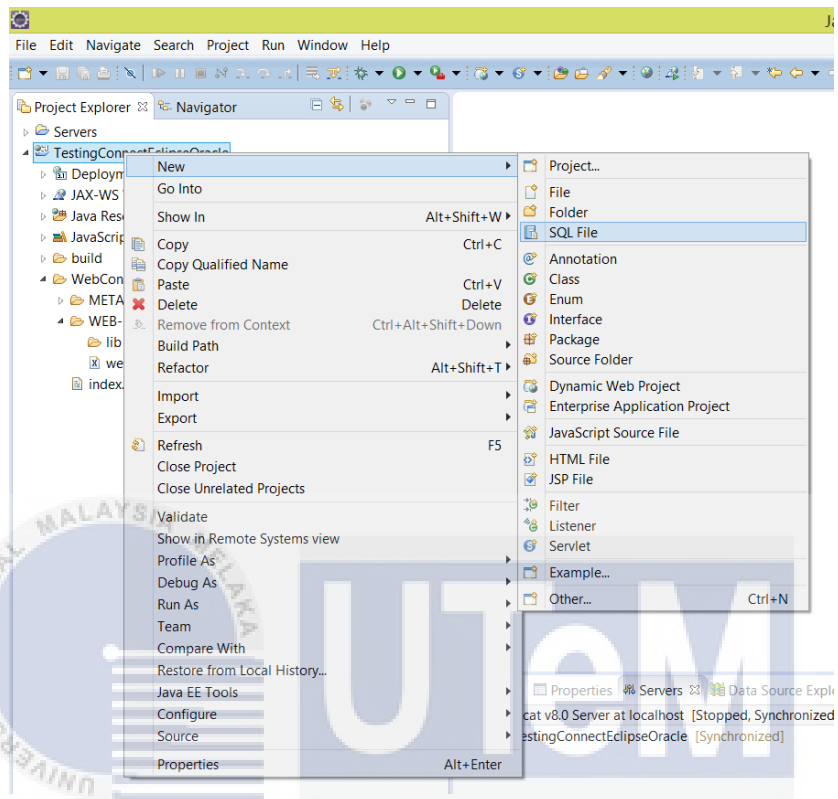
: Server Installation Step 14

STEP 15: Click on Oracle and enter the name as DBOracle and click on Next button.
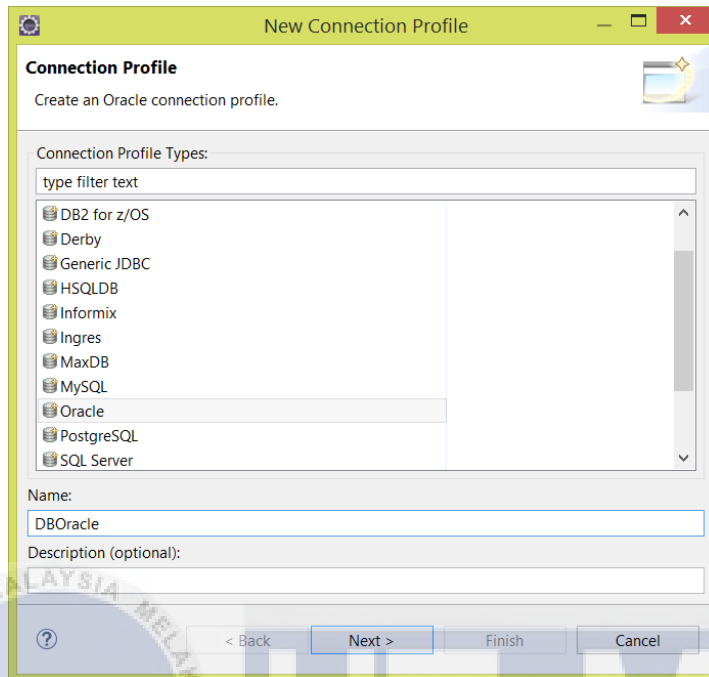
Server Installation Step 15

STEP 16: Click JAR List tab and click on Add JAR/Zip Button and find the driver in the file system and click Properties tab.
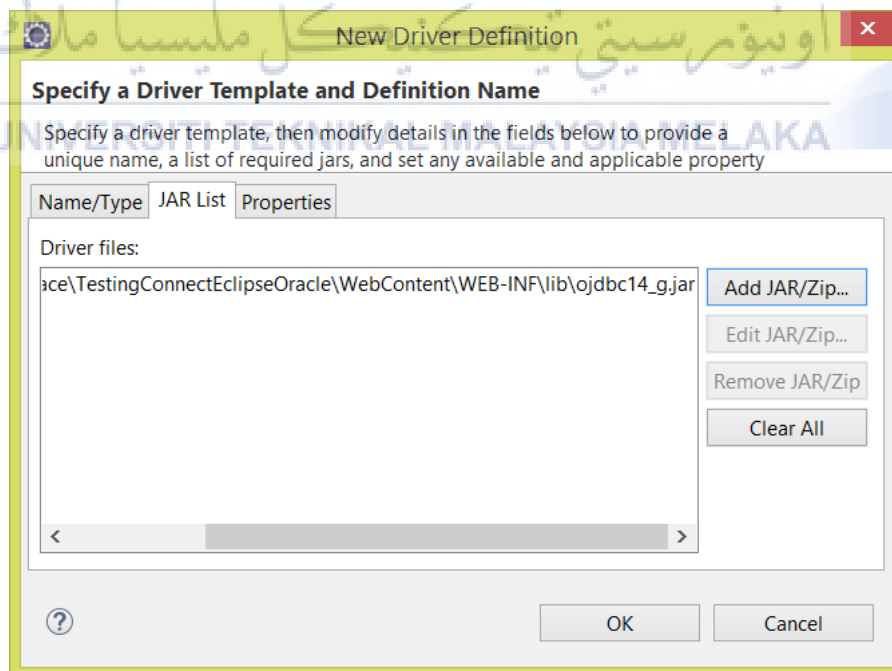
Server Installation Step 16

STEP 17: Edit the value as follow and click on OK button.

Server Installation Step 17

STEP 18: Click Test Connection button to test the connection with database.
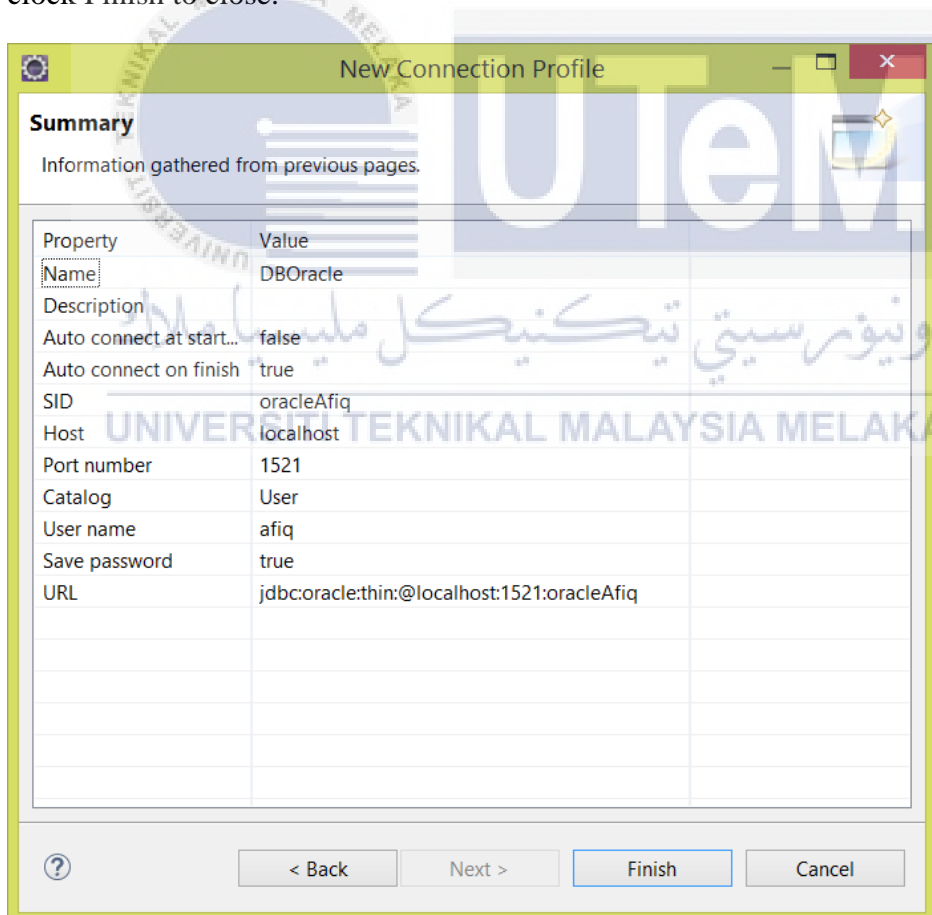

Server Installation Step 18

STEP 19: This popup will display if successful ping to the database and click Ok button to close
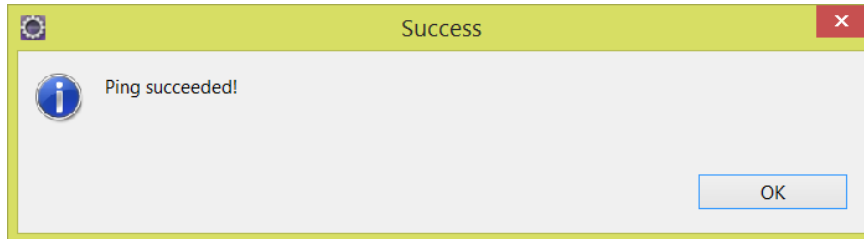
Server Installation Step 19

STEP 20: After all process successful, the summary of the connection will display and clock Finish to close.

Server Installation Step 20