ARITHMETICNORM FOR LEARNING DATABASE NORMALIZATION

AISHA HASAN SAIF AL-BAKRI

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

ARITHMETICNORM FOR LEARNING DATABASE NORMALIZATION

AISHA HASAN SAIF AL-BAKRI

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

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2021

DECLARATION

I hereby declare that this project report entitled

ARITHMETICNORM FOR LEARNING DATABASE NORMALIZATION

is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT : _____ Date : 12 /9 /2021

AISHA HASAN SAIF AL-BAKRI

I hereby declare 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 Honours.

SUPERVISOR : _____ Date : 12 /9 /2021

TS.HIDAYAH BINTI RAHMALAN

DEDICATION

I dedicate this project to my beloved parents who's mean so much to me and helped me in all things great and small.

I also dedicate this work to my respected supervisor, Ts. Dr. Hidayah binti Rahmalan who encouraged me and pushed me to complete the project.

All of you have been my best cheerleaders. Without them after Allah, I would not have achieved this success.

I will always appreciate all they have done; may Allah grant you all Jannah Firdaus.

ACKNOWLEDGEMENTS

In the name of Allah, the most gracious and the most merciful. Alhamdulillah, all praises to Allah for the strengths and his blessing in completing my final project.

First and foremost, I express my heartfelt gratitude to my beloved father and my beloved mother for their unwavering love and support throughout the years. Thank you for guiding me in learning and doing everything in my power to help me walk along the path of greatness. You sacrificed many things for me to provide me with all the happiness. Whatever success I have now is due to both of you. Both of you aided me emotionally and monetarily.

Special appreciation goes to my respected supervisor Ts. Dr. Hidayah Binti Rahmalan for assisting me in developing my technical abilities and supporting me throughout my project with her patience and knowledge while enabling me to work in my own way. Her support and persistence are responsible for my bachelor's degree, and without her, this project would not have been accomplished. There is not a better or kinder supervisor in the world.

Special thanks to my wonderful family for being a part of my life and being another huge part in my study and my rock when I needed it the most and for their supporting for every decision I make.

In my daily study, I have been blessed with a friendly group of friends and lectures. Thanks to all my lecturers and friends for their cooperation and their invaluable assistance.

ABSTRACT

Normalization, which is a systematic way of decomposing tables, is the most extensively used process for examining relational databases. Its purpose is to design a set of relational tables with the least amount of data redundancy possible while retaining consistency. It also prevents undesirable features like insertion, update, and deletion anomalies. Thus, it is important for students to correctly implement the database normalization process. However, some students may have trouble with database normalization, which can lead to data duplication and the creation of databases that are larger than they need to be, slowing access. From our preliminary survey, 60% among FTMK lecturers agree that normalization is the most difficult topic in database and 90% agree that students have difficulties in learning normalization topis. While 81% among students agree that normalization is a difficult topic and 22% agree that normalization is the most difficult topic in database. Due to this matter, our study proposed a normalization tool named as ArithmeticNorm. Our ArithmeticNorm tool provides an environment for learning normalization of relational database to give students an interactive hands-on experience in database normalization process. From a total of 32 ArithmeticNorm users, 93% found that our tool will increase their productivity and 84% agree that ArithmeticNorm system will be useful in their studies.

ABSTRAK

Normalisasi, yang merupakan cara yang sistematik untuk menguraikan jadual, adalah proses yang paling banyak digunakan untuk memeriksa pangkalan data relasi. Tujuannya adalah untuk merancang satu set jadual hubungan dengan sekurang-kurangnya jumlah data yang mungkin dan kemungkinan mengekalkan konsistensi. Ia juga menghalang ciri yang tidak diingini seperti penyisipan, kemas kini, dan penghapusan anomali. Oleh itu, adalah penting bagi pelajar untuk melaksanakan proses normalisasi pangkalan data dengan betul. Walau bagaimanapun, sesetengah pelajar mungkin menghadapi masalah dengan normalisasi pangkalan data, yang boleh menyebabkan duplikasi data dan penciptaan pangkalan data yang lebih besar daripada yang mereka perlukan, melambatkan akses. Daripada tinjauan awal kami, 60% daripada pensyarah FTMK bersetuju bahawa normalisasi adalah topik yang paling sukar dalam pangkalan data dan 90% bersetuju bahawa pelajar mengalami kesukaran dalam mempelajari topis normalisasi. Walaupun 81% di kalangan pelajar bersetuju bahawa normalisasi adalah topik yang sukar dan 22% bersetuju bahawa normalisasi adalah topik yang paling sukar dalam pangkalan data. Oleh kerana perkara ini, kajian kami mencadangkan alat normalisasi yang dinamakan sebagai ArithmeticNorm. Alat aritmetik kami menyediakan persekitaran untuk pembelajaran normalisasi pangkalan data relasi untuk memberi pelajar pengalaman yang interaktif dalam proses dalam proses normalisasi pangkalan data. Daripada sejumlah 32 pengguna aritmetik, 93% mendapati bahawa alat kami akan meningkatkan produktiviti mereka dan 84% bersetuju bahawa sistem aritmeticnorm akan berguna dalam pengajian mereka.

Table of Contents

CHAPTER III ANALYSIS

CHATER IV DEGIGN

CHATER V IMPLEMENTATION

CHATER VI TESTING

CHATER VII CONCLUSION

LIST OF TABLES

PAGE

LIST OF FIGURES

PAGE

CHAPTER I

INTRODUCTION

1.1 Project Background

Data is essentially the plain facts and statistics collected during the operations of a business. They can be used to measure/record a wide range of business activities - both internal and external. While the data itself may not be very informative, it is the basis for all reporting and as such is crucial in business.

Data normalization removes a variety of irregularities that might make data analysis more difficult. Anomalies might arise because of losing data, adding new information, or changing current data. Different benefits can be realized through other uses of the data and data analytics once the errors have been identified and eliminated from the system. Through data normalization, the information in a database can be formatted in such a way that it can be visualized

and analyzed.[1] Without it, a business can collect all the data it wants, but most of it would go unused, taking up space and providing little value to the organization. As a result, while attempting to load an integrated conceptual model into a database management system (DBMS), several issues may arise. These inconsistencies are caused by relationships that are created straight from user views.

There are several studies show the importance of database normalization and confirm the benefits of normalization for database. Mendjoge (2016) highlighted that it is critical for students to correctly implement the database normalization process. Codd (1970) confirm that the value of database normalization has been demonstrated and recommended in the literature for decades. While Ringlead (2019) highlighted that normalize your database before take action has numerous advantages for your sales. A database must be normalized to reduce redundancy (duplicate data) and ensure that only related data is stored in each table Li (2019). Winters (2021) said "normalization is important to protect data, correct data, and clean the database". While Muñoz (2019) said "It is more crucial to organize your company or catalog information than it is to find that pair of socks in your drawer every morning". Yee (2016) highlighted that the benefits of employing normalization are substantial, while the disadvantages are unavoidable in some situations. Normalization is important because it is extremely difficult to store items in a relational database that keeps the same information in many places Wikipedia (2018).

Therefore, various methods for improving teaching and learning database normalization were designed and developed. ArithmeticNorm system gives students an interactive hands-on experience in database normalization process starts from 1NF, 2NF, and finally 3NF by using certain techniques to determine a set of keys that exist in the relational database schema and proceed with the process of normalization. The goal of this study was to overcome the lack of knowledge and the teaching strategy used to improve learning database normalization.

[1] Import.io. (May 2019).

1.2 Problem Statement

This research starts with identifying problems and issues related to normalization. In this section, the problem statement was based on research materials such as journals or academic papers, and also preliminary findings through our survey.

There has been a lot of research done to show the complexity in the database normalization process, and how this is a tough topic not just for new database designers and developers, but also for university students studying this process. In this regard, Shanardi (2018) confirmed that normalizing a database is a complex and tough topic since analysts must understand the database's purpose. Brumm (2017) highlighted that database normalization concept can be hard to understand. Narvekar et al. (2016) observed that while taking a database management course, students find the processes of normalization and the ideas of normal forms is difficult to. Cooper (2021) said, “I found normalization quite difficult early in my job, likely because of my lack of experience”.

According to Georgiev, N. (2008), computer science students have difficulty learning the fundamentals or concepts of database normalization, and difficulties in motivating them to learn database normalization because the subject appears to students to be dry and tough. Wang et al. (2010) highlighted that, because of the dry and academic manner in which database normalization is presented in textbooks and classes, it can be tough to motivate students to learn normalization topics.

Meanwhile, from our finding, normalization tends to be complex for new students and designer which could lead to difficulties that students face in learning database normalization. Thus, there are several problems that occur such as:

i. Lose interest and trust in programming.

Due to that student may get poor database design, students will lose interest and trust in their programming and system they develop. Thus, that could lead students to do their work with less confidence.

ii. Current normalization tools do not provide a clear understanding of normalization.

Current algorithms that are used in available database normalization tools do not provide a clear understanding of normalization.

iii. Current normalization tools do not provide a visual aid for normalization.

The functional dependencies (FDs) that user input in available database normalization does not has data that show the user why his table needs to be separated into a set of tables. As a result, these tools do not provide a visual aid for normalization process.

According to the teacher's and student's preliminary survey results that can be referred in Section 2.4, normalization was confirmed the toughest and the most difficult topic in the database subject for students and for teachers as well. From a total of 10 respected teachers who teach database subjects, majority of them have chosen that normalization is a difficult and tough topic. While from a total of 97 students who took database subject, 79 students have chosen that normalization is a difficult topic. This becomes our interest to help students in learning normalization.

1.3 Objective

ArithmeticNorm system aims to overcome students' difficulties that they face while learning database normalization, the objectives of this project are as follows:

i. To investigate students' and lecturers 'opinions about normalization topic.
ii. To develop a tool that help student visualize normalization.
iii. To evaluate the suggested normalization tool that has been developed.

1.4 Project Scope

The scope of this study is related to the topic of normalization. In this section, the scope also includes target users and module of system as shown in Section 1.4.1 and 1.4.2 respectively.

1.4.1 Target Users

The target user of ArithmeticNorm are computer science students, database designers, and those who are interested in the educational purposes of database normalization as well. Thus, the position of user that the system has is divided into two which are:

I. Student

Students can register to the system and start learning database normalization and practice their own database as well. so that users allow to insert and delete tables that are in CSV format, delete or update profile that include image media data, and view exams results.

II. Admin

The administrator manages all functions in the system and manages media data as well that could be added, updated, or

deleted video and audio, do the functions that normal users have. In addition to getting an analysis of all users, tables, results, and media data.

1.4.2 Modules of System

There are 5 modules of ArithmeticNorm system which are:

I. Registration Module

This module will assist the student in registering and will make the manual registration process easier.

II. Login module

The authentication of user accounts will be aided by the login module. Users can access their accounts if they have a valid login username and password.

III. Assessment Module

This module will provide a testing environment where users can examine their knowledge gained in the database normalization.

IV. Theoretical Module

Theoretical module consists of theoretical explanations of normalization topics covering the normalization phase of the database design.

V. Practice module

This module will help students to practice database normalization through the system. ArithmeticNorm system guides students to perform a good database and normalize their table.

1.5 Project Significant

Due to the difficulties in learning normalization, this study intended to help and give a visual aid to the student in learning normalization. The suggested prototype that this study with to develop will benefit the students in their study and also in their career therefore, most of them would like to check the normalize of their database or to normalize their database as well.

According to the literature study table that will discover in Table 2.1, it has found that there are few commercial design tools are available to help students understand and learn normalization. These tools could be difficult for students to find out since it is not easily access. Therefore, ArithmeticNorm system expected to be more accessible, timely, and easy to understand by students in addition to solve student's problems that shown in Section 1.3.

1.6 Expected Output

ArithmeticNorm system expected to assess students' normalization knowledge by providing self-test questions. The system expected to provide media data in the education field to teach student normalization theories. ArithmeticNorm system also expected to accept students' CSV files, and push students to think about how to normalize their table by guide them to correct answers, and generate 1NF, 2NF, and 3NF as output. Students may have to practice or test his/her table.

1.7 Conclusion

As a conclusion of this chapter, the initial analysis part of developing ArithmeticNorm system which has project background, problem statement, objectives, user scope, develop module, project signification, and expected output of this research have covered. ArithmeticNorm system is more helpful for students who are interested in learning database normalization.

CHAPTER II

PROJECT METHODOLOGY AND PLANNING

2.1 Introduction

A methodology will consist of phases, themselves consisting of sub-phases[2], which will guide to the choice of the techniques that might be appropriate at each stage of this project. During the development of ArithmeticNorm system, these phases helped to plan, manage, control, and evaluate information system projects. This chapter describes in detail about system method and system development life cycle that used in this system.

[2] Ukessays, three information systems development methods information technology essay. (Jan 2015).

2.2 Project Methodology

In this project, the methodology used in the system divided into two which could be easier to develop the system more efficient and effective. The methodologies that have been used are as the following:

2.2.1 Development Methodology

The first methodology that has been used is rapid application development (RAD). RAD was conceived in the 1980s, so it is not something new. But unlike the waterfall model, it is not singular.[3] It is a continuous evolution of development philosophies according to the system requirements until the end of this semester. RAD is a rapid development method that results in a high-quality solution and contains four phases: requirements planning, user design, construction, and cutover.

According to ArithmeticNorm system which use research prototyping, it found that Rapid application development methodology in the Figure 2.1, is more suitable for this system unlike DevOps, and Agile development methodology because of that design, and construction phases of this project will be repeated until the confirmed that the system meets all requirements which include generating normalization forms using arithmetic science.

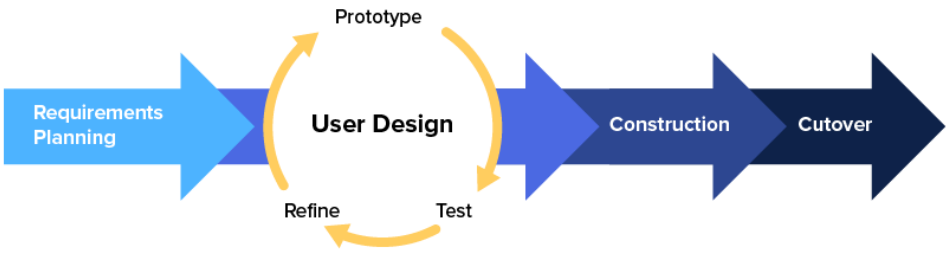


Figure 2. 1: Rapid Application Development

[3] Kiss flow, Rapid Application Development (RAD). (March 2021).

2.2.2 Database Methodology

Starting with requirements analysis and finishing with monitoring and modification, the DBLC specifies the steps involved in implementing the database. Furthermore, the DBLC never ends because database monitoring change and maintenance are all part of the life cycle and continue long after the database has been implemented. Simply said, the DBLC refers to the system database whole life cycle.

As shown in the Figure 2.2, there are five stages in the database life cycle which are requirements analysis, logical design, physical design, implementation, and monitoring that include modification, and maintenance.

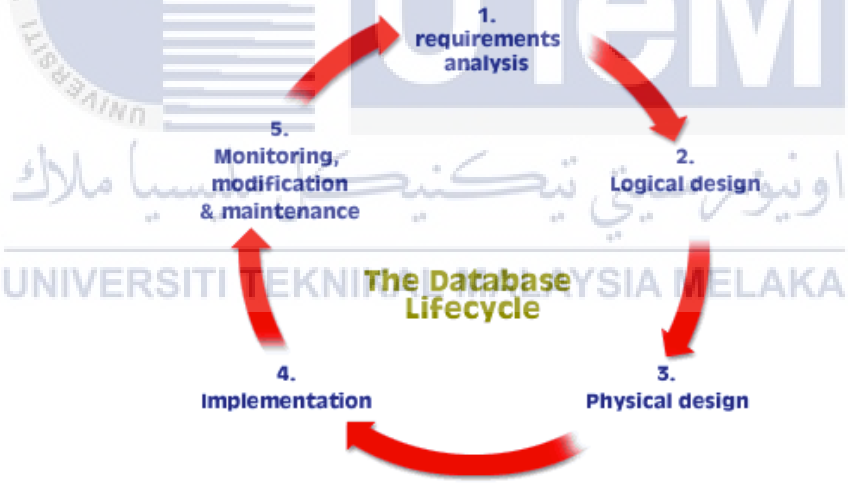


Figure 2. 2: Database Life Cycle (DBLC) [4]

[4] Relational DB Design, (DBLC) Database Life Cycle.

2.2.3 Requirements Analysis Phase

Requirement's analysis is to collect information that needs to be stored and manipulated. This stage involves assessing the informational needs to develop ArithmeticNorm system so that a database can be designed to meet those requirements. Since this study is to develop a tool in helping students in learning normalization, thus requirements will be gathered through literature review on several materials and tools related to normalization. This can be referred to Section 2.3. The requirements of this system are:

- The users need to upload CSV files to generate normal forms and show the table in GUI after normalizing into normal forms.
- Media data which include videos and images to be used in education field.
- Users need to evaluate themself in normalization topics.
- System must generate set of user table attributes and set of rows.

2.2.4 Logical Design Phase

During the initial portion of logical design, a conceptual model is produced based on the requirements assessment that has been done in the first stage. A conceptual model is typically an entity-relationship (ER) diagram that shows system database columns, fields, and primary keys, as well as how the tables are associated together (linked). The ER diagram can be referred in Section 4.3.1.

2.2.5 Physical Design Phase

The physical design stage's goal is to improve database performance. This requires determining how to increase the RDBMS's performance of the system. The two most time-consuming tasks in an RDBMS are getting data from and writing data to a database, both of which can be sped up by tweaking some database design components. The physical design can be referred in Section 4.3.3.

2.2.6 Implementation Phase

In this phase, represents the work done to meet the system requirements of the scope of work in Section 2.2.3 and fulfill the charter. During the implementation phase, DBMS installation must be performed. The DBMS can be installed on a new server or an existing server. After installing the DBMS, the next step is to create a database using the standard database language, structured query language (SQL), in the selected DBMS. Next, the data must be loaded into database tables. The implementation phase can be referred in Chapter 5.

2.2.7 Testing and Evaluation Phase

The testing phase of the development lifecycle is where you focus on investigation and discovery.[5] During the testing phase, we determine whether system code and programming work according to the system requirements that are in Section 2.2.3. Test scripts ensure consistency while testing while evaluation involves evaluating the system to stand-alone system, integrated system, and, if appropriate. The testing phase can be referred in Chapter 6.

2.2.8 Maintenance Phase

To support system operational effectiveness, the maintenance process includes making improvements to hardware, software, and documentation. It entails making changes to improve the system performance, fixes bug, improves protection, or meet system requirements. Since this is research prototype, we still need more lengthy time to be tested among students and lecturers before being in the maintenance phases.

[5] Study, testing phase in SDLC.

2.3 Literature Review

The purpose of this study is to develop a tool that can help students to check their database normalized or not. We start with identifying the most difficult topic among database and through our preliminary survey among teachers and students, it is as our prediction that normalization is the most difficult topic. In addition, the survey also manages to identify the reason why normalization is tough. This is to measure the first and second objective and the results can be referred in Section 2.4.

In order to develop the tool, this study also investigate some existing materials and tools as displayed in Table 2.1 which display an analysis of published works by other researchers related to the topic of normalization. This analysis covers the information on the methods, the topics, the advantages, and how the tool works.

A few authors have been developed a database normalization tool. Because database normalization improves data integrity for students and for developers as well, many of these authors said that database normalization tools are the most important and helpful for students to do their programming with more confidence. This study also includes some definition related to normalization as displayed below.

Definition

There are many definitions of database normalization which include:

- Normalization is a method of systematically deconstructing tables in order to remove data redundancy (repetition).[6]
- The process of arranging a database is known as database normalization.[7]
- Database normalization is a database schema design approach that involves modifying an existing schema to reduce redundancy and data dependencies.[8]

[6] Study tonight, normalization of database.

[7] Wikipedia, database normalization. (March 2018).

- The technique of efficiently structuring data in a database is called normalization.[9]
- Normalization, also known as canonical synthesis, is a technique for creating database structures to hold data.[10]

Normalization could be a complex process with many specific rules and different intensity levels. In information technology, database normalization is part of successful database design which organizing data in a database while also allowing the database to be more adaptable and flexible.

This study expected to advance normalization learning by helping students discover more about database normalization and by helping them do their programming with more confidence.

Through the finding in our literature review, we think that eliminate partial and transitive dependencies automatically and find normal forms based on the functional dependencies will be effective and efficient in designing user databases which are important in ensuring consistent data, elimination of data redundancy, efficient execution of queries and high-performance application.

[8] W3schools, database normalization.

[9] Chapple, Mike. Life wire, database normalization basics.

[10] Tutorial ride, Database Normalization.

No	Reference/ Year	Category	For whom	Topics used	Advantages	How to work	Name of application
1.	[4], 2008	Research	Employees.	Subset property, augmentation, transitivity, dependency graph, dependency matrix, and directed graph matrix.	1. Automated relational database normalization method. 2. Produces the dependency matrix and the directed graph matrix. 3. Generating the 2NF, 3NF, and BCNF normal forms. 4. Automatically distinguish one primary key for every final table.	Employing a mathematical normalization approach based on dependency matrix generation This approach begins by initializing functional dependencies using a set of simple symbols and keys, followed by determining all potential paths between all pairs, revealing all transitive dependencies between determinant keys. Then, using the Circular-Dependence technique, recognize dependency closure and find the direct dependency. Then, instead of transitive dependency, original direct dependency is used. After that, it moves on to regular forms 2NF, 3NF, and BCNF.	Automatic Database Normalization and Primary Key Generation.
2.	[18], 1978	Research	No specific user.	Projectivity, transitivity, additivity.	1. Find set of all minimal keys. 2. Deciding whether or not there is a key of cardinality less than or equal to a specified integer. 3. Find a third normal form.	This tool will use Codd to find keys for sub-relations, which will then be used to find third normal forms. Tables with indivisible data values are used to represent data. The table's rows correspond to the entities being described, while the columns contain the attributes used to describe them. Each column has an attribute name associated with it that uniquely identifies it.	Candidate Keys for Relations
3.	[26], 2018	Research	For users who need help on decomposing datasets of medical records.	Minimality pruning, right-hand side pruning, and key pruning.	2 Defines a rule constraint between two sets of attributes. 3 Find set of function dependency.	This tool accepts sets of FDs and infers candidate keys from the database, then assigns the left-hand side attribute combinations of a set of FDs to dictionary keys and their closures to the appropriate values, and finally minimizes the set of FDs to a minimum coverage using Python.	A Python application to mine for functional dependencies and candidate keys in tabular data.
4.	[14], 2008	Research	Students and lecturers to teach and test students.	Queries, and join.	1. Candidate decomposition of a given relation. 2. Check normal forms. 3. Decomposition of Relations. 4. Use of efficient data structures. 5. Use of advanced tools for code optimizations such as the GWT's Java-to-JavaScript	Decomposition algorithms for deconstructing a normalized database schema into a specified normal form utilizing functional dependencies, and Test algorithms for testing whether a given relational schema violates certain normal forms can be used with GWT and JavaScript. Generate a sample solution to a given assignment and present it to the user, then run a series of tests to ensure the solution is correct before	A Web-Based Environment for Learning Normalization of Relational Database Schemata.

					compiler.	storing it in the database.	
5.	[13], 1988	Research	users who need help in specify functional dependencies.	Queries, and transactions.	1. projecting functional dependencies on to sub relations. 2. Computing minimal covers and synthesizing relations into third normal form. 3. Benchmarks to validate theoretical foundations a which derived from a "real" database. 4. Removing redundant dependencies. 5. Computing the closure of a set of attributes. 6. Improve compute minimal covers. 7. Eliminate redundant dependencies.	This tool employs an r-closure, a type of closure and a high-performance algorithm for calculating the closure of a set of attributes. To begin, this approach locates sets of attributes, individual attributes, and limits the right/left sides of FD to a single attribute. Then generate third normal form relations to lower the total number of closures computed.	New Methods and Fast Algorithms for Database Normalization.
6.	[16], 2011	Research	Users who want to Normalize their database in 1NF, 2NF, and 3NF.	Linked list nod, linked list, and queries.	1. Normalizing relations within the shorter time. 2. Requires less space. 3. Determiners prime attributes and non prime attributes.	This technique employs a novel approach in which a connection and its functional dependencies are represented by a single linked list/ node structure (which is a node identifier, a unique number provided to each newly formed node and stored inside the node itself). This utility normalizes a relation before constructing a table and populating it with records. First, a node is added to a linked list, and then all of the prime and non-prime attributes are entered, followed by information on all of the FDs that hold the relation. The last stage is to produce 1NF, 2NF, and 3NF.	Possible Algorithms of 2NF and 3NF for DBNorma-A tool for Relational Database Normalization.
7.	[32], 1989	Research	Users who in academic circles, marketplace, and fields of application.	Hierarchical, network, and relational.	1. Accurate, up - to – date information is available on demand at any time. 2. The machine can retrieve and change data fast. 3. define the relational schema. 4. Find set of functional dependence.	This tool defines the contents of a Relational Schema using the PROLOG programming language. Data is entered into an existing file, which is then checked for redundant lines and excluded, as well as whether the values of various domain attributes are recorded as sets or lists. If such records exist, they must be converted to a usable form. After that, perform the into 1NF, 2NF, and 3NF.	Automated normalization tool.
8.	[36], 2007	Research	Users whose interest in educational purposes.	Algorithm BCNF, and Mathematica.	1. Find primary key. 2. Find a set of functional dependencies (FD's) among the attributes of each schema. 3. Elimination of redundant	Obtaining the minimal cover of the functional dependency set and evaluating an attribute for a candidate key using fundamental set operations such as closure and complete closure in conjunction with	JMathNorm: A Database Normalization Tool Using Mathematica.

					attributes. 4. Set full closure. 5. Testing for primary key and obtaining minimal cover. 6. Allow student to check their manual work in studying the normalization algorithms and to normalize schemas for the database design.	modules. This tool converts business rules into a set of entities with a set of properties and relationships between them using the Java programming language. The Java Link (JLink) function in Mathematica is used to run Mathematica modules.	
9.	[30], 2015	Tool	Students	Functional dependencies, candidate keys, and queries.	1. Find candidate keys. 2. Check normal form. 3. Normalize to 1NF, 2NF, 3NF, and BCNF. 4. Find minimal cover. 5. Show normalization steps. 6. Show functional dependency.	Ask user to enter their attribute and all functional dependency for that attribute, then, the system determine candidate keys and check for 1NF, 2NF, 3NF, and BCNF. This system also normalize into normal forms by showing functional dependencies.	Normalization Tool.

Table 2. 1: Summary of Literature Review

Based on the literature review study that discussed in Table 2.1, it is found that current algorithms that are used in available database normalization tools do not provide a clear understanding of normalization. However, as shown in Figure 2.3 these tools do not engage users to use the tools by providing a good environment such as provide related topics that allow users to go through and learn about the subject, use media data as a part of the education field, and provide exam environment that could be challenges for users to test their knowledge.

Most normalization tools consider database relations and functional dependencies (FDs) as two distinct inputs and thus lead to partial automation. For example, as shown in Figure 2.3 the functional dependencies (FDs) that user input does not has data that show the user why his table needs to be separated into a set of tables.

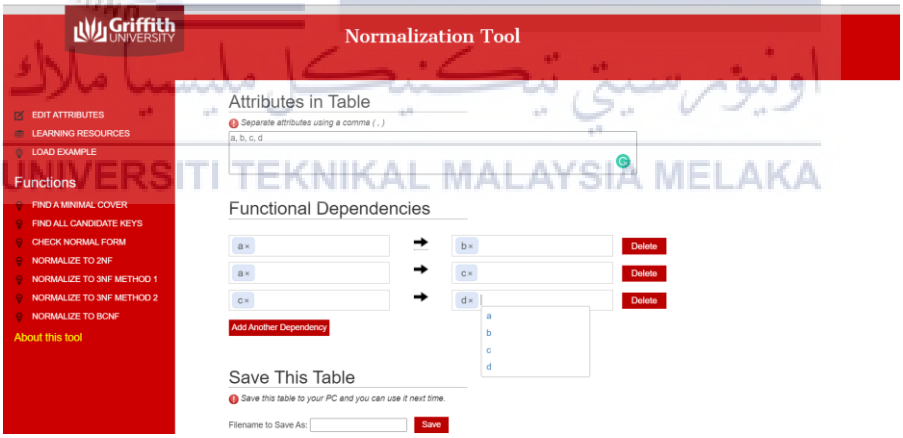


Figure 2. 3: Normalization Tool by Griffith University [30]

2.4 Preliminary Survey

We conduct a preliminary survey in identifying the toughest topics in database subjects from students' and teachers' perspective.

a. Student's survey

In this survey, the questionnaire is used to collect data and getting opinions from students regarding the database normalization through Google form among different universities. The link and list of questions can be referred in Appendix B.

We received 97 responses from different universities and most of the respondents are from FTMK students at UTeM university. As shown in Figure 2.4 up to Figure 2.6, there are 46% of the respondents are male while 53% of them are female. 23% with a range of age 25 - 27 and 44% with a range of age 22- 24. 74% of the responses are from lectures with a bachelor's degree/ graduate degree while 11% of them with a diploma degree.

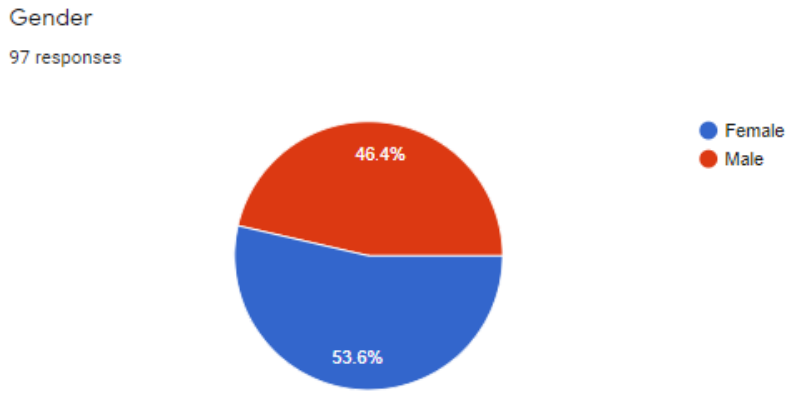


Figure 2. 4: Gender of the Respondents

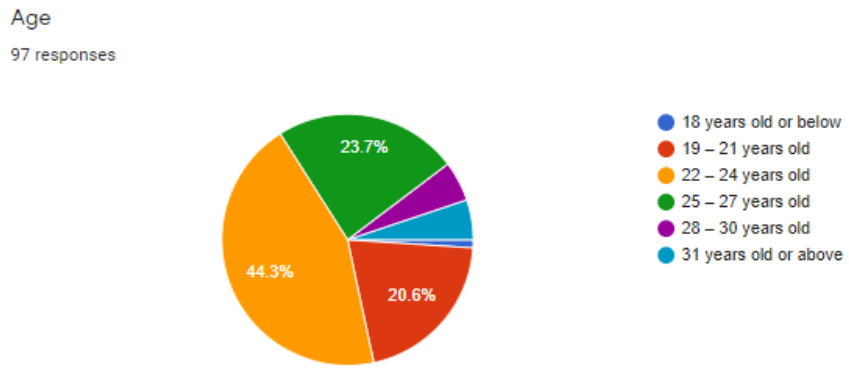


Figure 2. 5: Range of Age of the Respondents

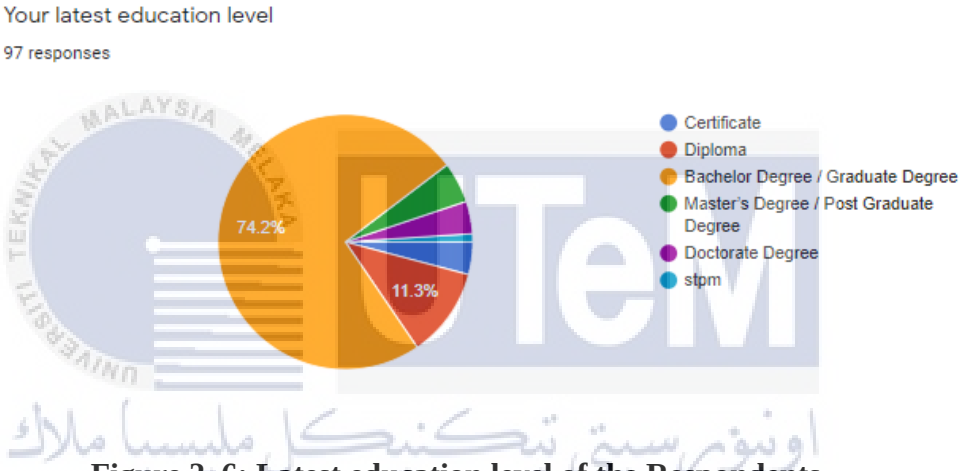


Figure 2. 6: Latest education level of the Respondents

27% of the respondents have taken database subject last three semesters, and 22% of them have taken database subject last two semesters as shown in the Figure 2.7.

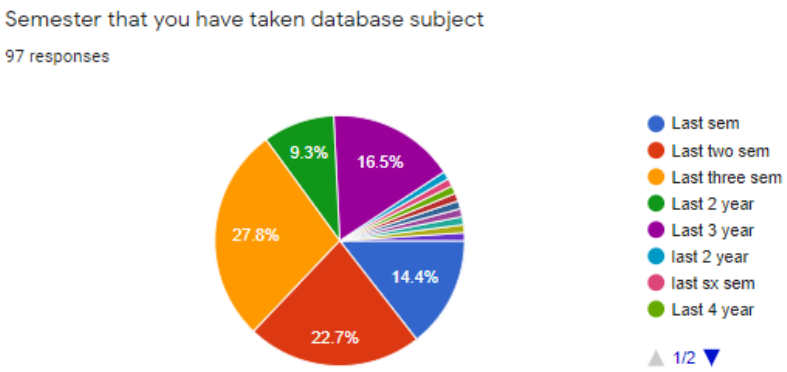


Figure 2. 7: Semester that Respondents Taken Database Subject

As shown in Figure 2.8, normalization has been agreed by 81% respondents as a difficult topic in database. They gave reason that this is difficult topic because they get confused when they normalize their database and it is difficult to understand, while some of them said because they do not know when normalize their database and it takes time to identify normal forms, normalization is difficult to implement, and because less of knowledge. Some of them also admit that normalization is very important, but it is very difficult, and they spent a lot of time learning and understanding it.

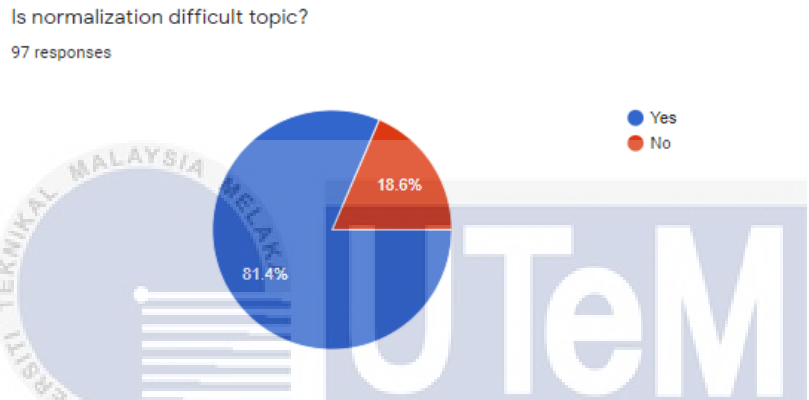


Figure 2. 8: Difficulty of Normalization

There are 63% of the respondents found that normalization is a tricky topic when they implement normalization process. And 57% of them found that normalization is complex topic as shown in the Figure 2.9.

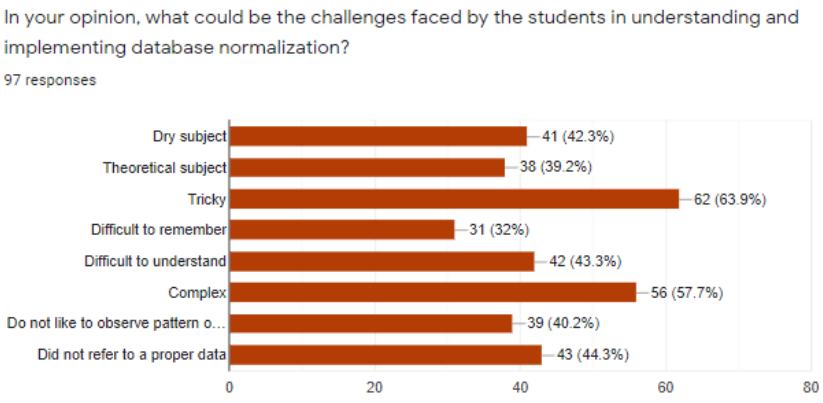


Figure 2. 9: The Challenges Faced by The Students

There are 22% of the respondents have chosen that normalization topic is the most difficult topic of database topics, and 13% of them have chosen that data manipulation language topic is the most difficult topic of database topics as shown in the Figure 2.10.

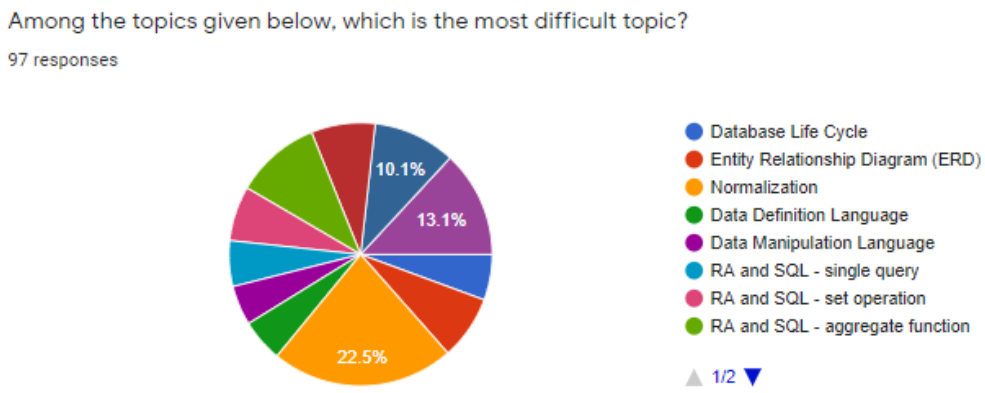


Figure 2. 10: The Most Difficult Topic of Database Topics

Figure 2.11 shows the agreement of having a normalization tool that checks from normal form. There are 48% of the students want to have that tool, and 20% of them want to have it if they are available. In total 68% would be happy to use the normalization tool, giving us good support indirectly as we plan to develop this ArithmeticNorm tool.

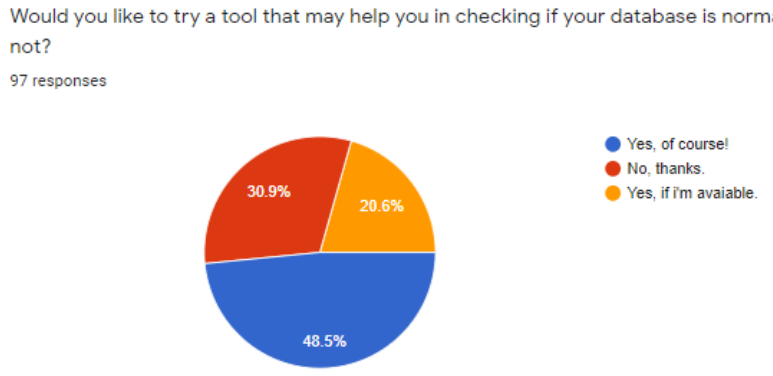


Figure 2. 11: Agreement of Having a Normalization Tool

Figure 2.12 shows the agreement of the benefits of a normalization tool. There are 74% of the students said that a normalization tool will help students learn the normalization concepts, and 25% of them said a normalization tool can be helpful for students to learn the normalization concepts. Some of them said, by using a tool, the student can know their mistake, can help a lot of students, can save a lot of time and effort, can make databases easier to design, and students can study normalization topics with more interesting.

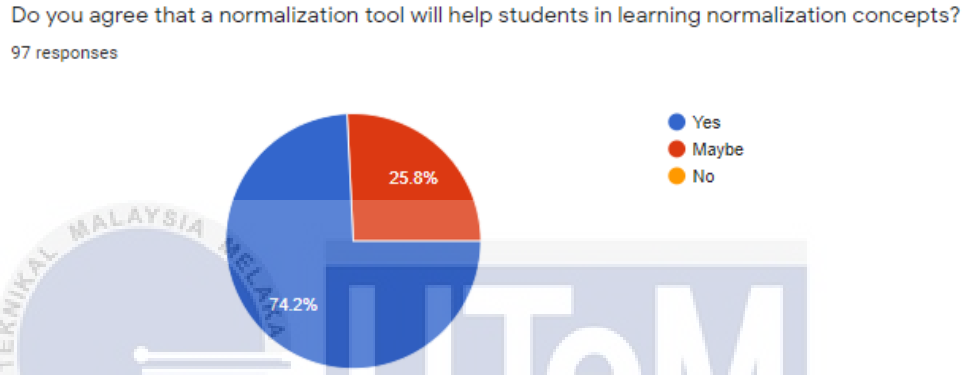


Figure 2. 12: Agreement of the Benefits of a Normalization Tool

In summary, according to the student's preliminary survey, normalization has become the toughest and the most difficult topic in the database subject. From a total of 97 students who took database subject, 79 students have chosen that normalization is a difficult topic. This becomes our interest to help students in learning normalization.

b. Teacher's survey

In this survey, the questionnaire is used to collect data and getting opinions from teachers regarding the database normalization through Google form among the respected lectures of UTeM university. The link and list of questions can be referred in Appendix C.

We received 10 responses from respected lecturers at different universities and most of them from FTMK lecturers at UTeM university. As shown in Figure 2.13 up to Figure 2.15, there are 80% of the respondents are female while 20% of them are male. 40% with a range of age 40 – 43, 20% with a range of age 32- 35 and 30% of them 44 old year and above. 60% of the responses are from people with a master's degree while 40% with a doctorate degree.

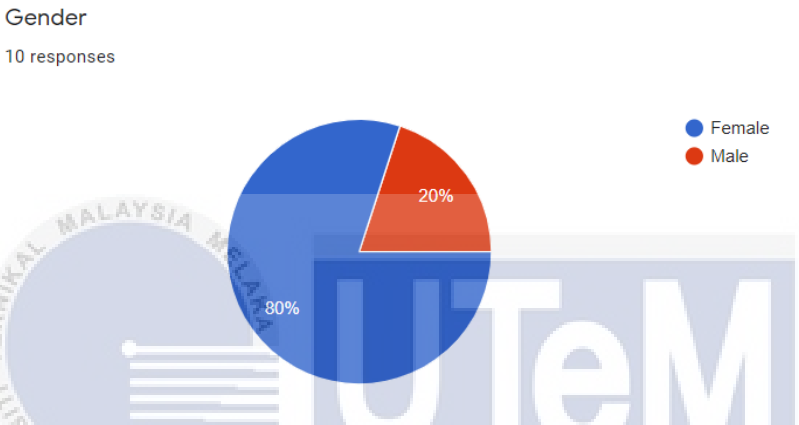


Figure 2. 13: Gender of the Teacher's Respondents

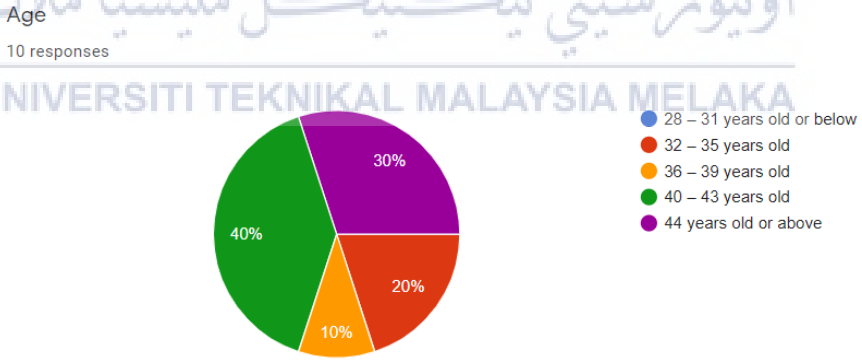


Figure 2. 14: Range of Age of the Teacher's Respondents

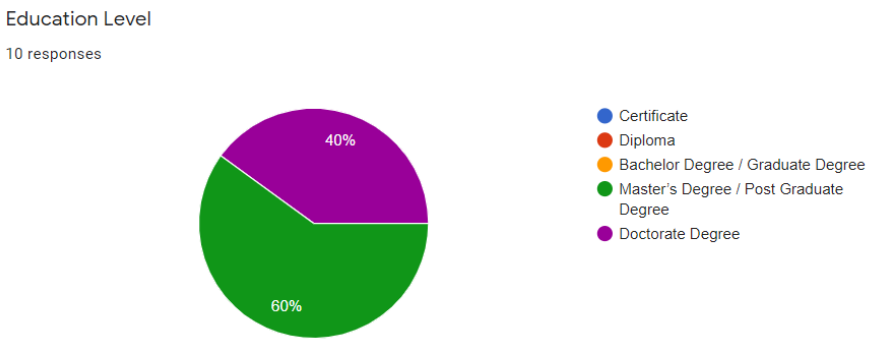


Figure 2. 15: Education Level of the Teacher's Respondents

Figure 2.16 shows the duration of the teaching database subject of the respected teachers. 30% of them are teaching database subject in a range of 17 to 20 years. And 20% are teaching database subject in a range of 3 to 6 years.

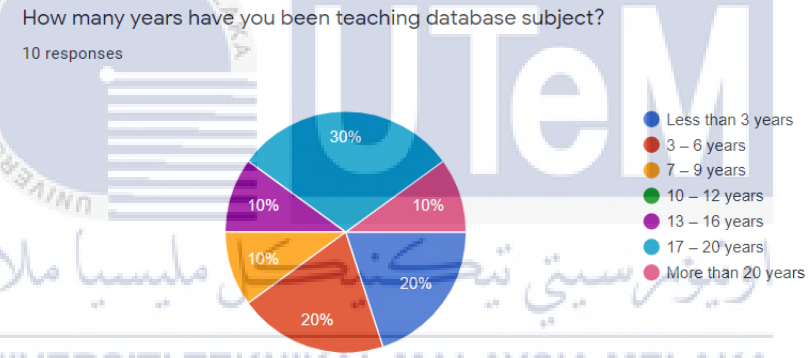


Figure 2. 16: Duration of Teaching Database Subject

As shown in Figure 2.17, normalization database is a topic that all the respondents teach.

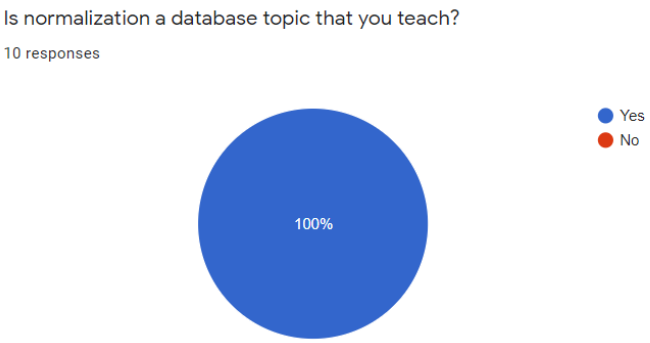


Figure 2. 17: Teaching Normalization Topic

As shown in Figure 2.18, 90% of the respondents agree that students have difficulties in learning normalization topics. Respondents have confirmed many reasons which are it is because some students cannot grab the concept, it is because they are unable to identify the dependency and because that student finds it is difficult to relate the normalization concept with the real situation. There is one respondent highlighted that student needs to master the theory of functional dependency to aid in understanding the normalization topics.

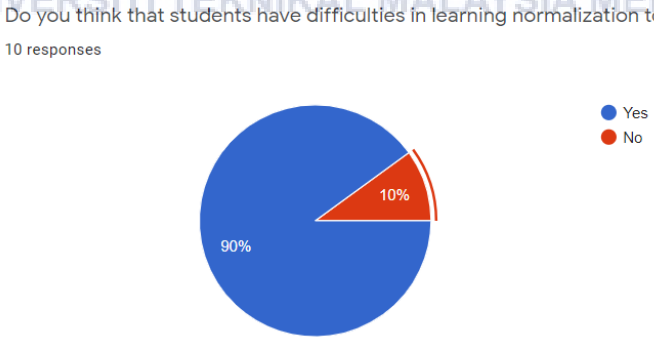


Figure 2. 18: Agreement on Facing Difficulty in Learning Normalization by Students

The weighted average was evaluated to detect each database topics average of experience. Table 2.2 presents the weighted averages of the scale experience of database topics.

Table 2. 2: Weighted Averages of the Scale Experience of Database Topics

Database Topics	Very difficult	Difficult	Neutral	Easy	Very easy	Not applicable	weighted averages
	1	2	3	4	5	6	
1. Database Introduction			1	2	6	1	4.70
2. Database Life Cycle.			1	3	5	1	4.60
3. Entity Relationship Diagram	1	3	1	3	1	1	3.30
4. Normalization	4	1	1	2	2		2.70
5. Data Definition Language		1	4	1	3	1	3.90
6. Data Manipulation Language		1	4	1	3	1	3.90
7. RA and SQL - single query		2	3	1	2	2	3.90
8. RA and SQL - set operation		4	1	1	2	2	3.70
9. RA and SQL - aggregate function		4	1	2	1	2	3.60
10. RA and SQL - inner Join		5	1	1	1	2	3.40
11. RA and SQL - outer Join		5	1	1	1	2	3.40
12. RA and SQL - subquery		5	1		2	2	3.50

There are 4 respondents find that is very difficult to teach normalization topic, while 1 of them find that it is difficult to teach normalization topic as shown in the Figure 2.19. The numbering below represents the scale of experience.

1 - Very difficult.
2 - Difficult.
3 - Neutral.
4 - Easy.
5 - Very easy.
6 - Not applicable (Never teach).

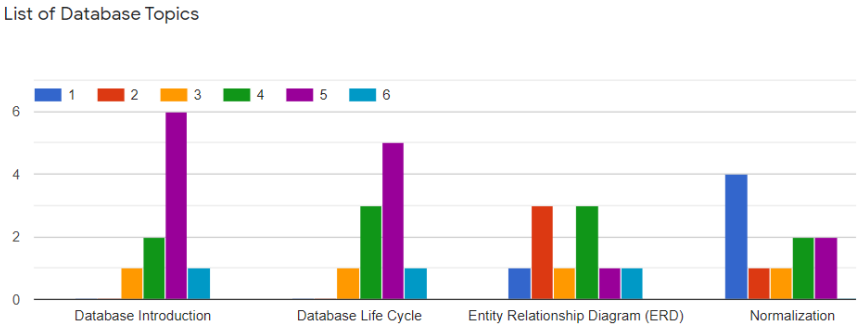


Figure 2. 19: Scale of Experience

There are 60% of the respondents have chosen that normalization topic is the most difficult topic of database topics because sometimes students do not know to relate normalization with real world and because it requires more explanation of FDs theory. One respondent shows the importance and the difficulties of normalization and highlighted that the wrong process of normalization could lead to bad database design. 10% of respondents have chosen that data manipulation language topic is the most difficult topic of database topics as shown in Figure 2.20.

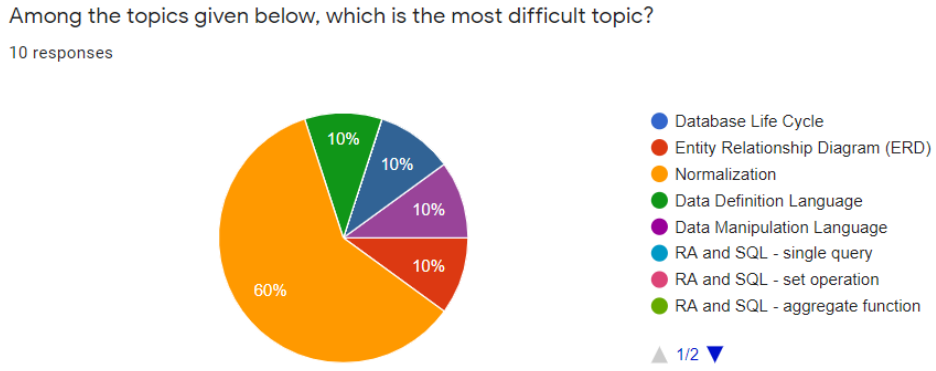


Figure 2. 20: The Most difficult Topic of Database Topic for the Teachers

As shown in Figure 2.21, 90% of the respondents do not have experience using normalization tool.

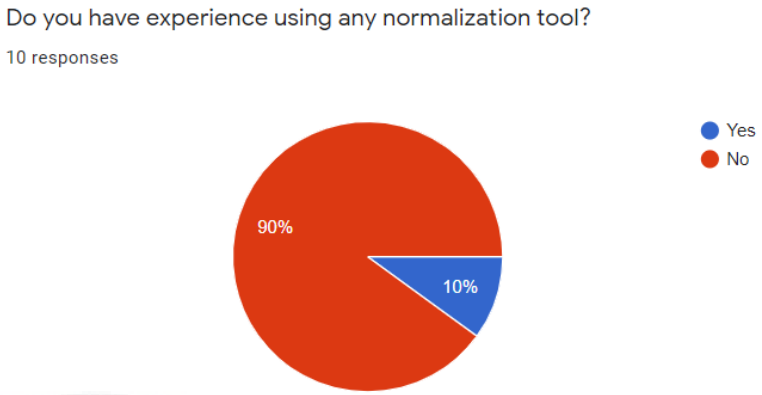


Figure 2. 21: Experience using Normalization Tool

There are 50% of the respondents said that a normalization tool will reduce the time needed to teach normalization concepts, while 50% of them said that normalization tool could be reduce the time needed to teach normalization concepts as shown in Figure 2.22. The respondents show the benefits of a normalization tool. Some of them said a normalization tool can help to normalize a database easily and students can visualize clearly if they can play around with a tool. In addition to confirming that a normalization tool will help in teaching the concept of normalization.

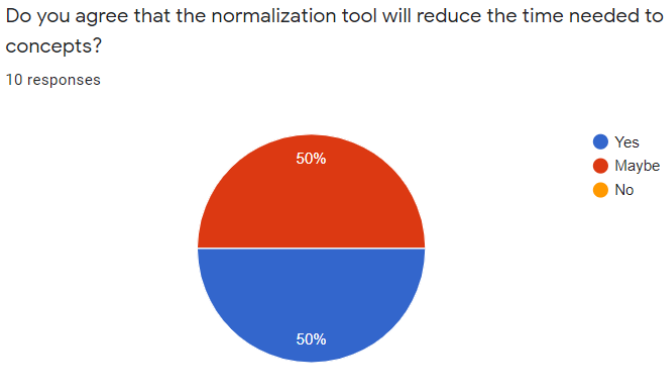


Figure 2. 22: Agreement of the benefit of a Normalization Tool

In summary, from a total of 10 respected teachers who teach database subjects, majority of them have chosen that normalization is a difficult and tough topic.

2.5 Project Schedule and Milestones

In this section, project schedule and milestones tool are provided to delineate a point in a project schedule. These points can note the start and finish of the project and mark the completion of a major phase of work. The purpose of this section is to ensure effective management and keeping the work on track as shown in Table 2.3 and Figure 2.23.

Table 2. 3: project Milestones

Milestones	Documents	Date
1. Project plan. **2. Problem identification and analysis.**	1. Function, and nonfunction requirements. 2. Flow chart of the proposed system.	21-28 March 2021
Conceptual design.	Entity-Relationship Diagram.	29 March - 4 April 2021
Physical design.	System interface design	5- 18 April 2021
Implementation and testing of the proposed system.	System prototype	19 April – 30 May

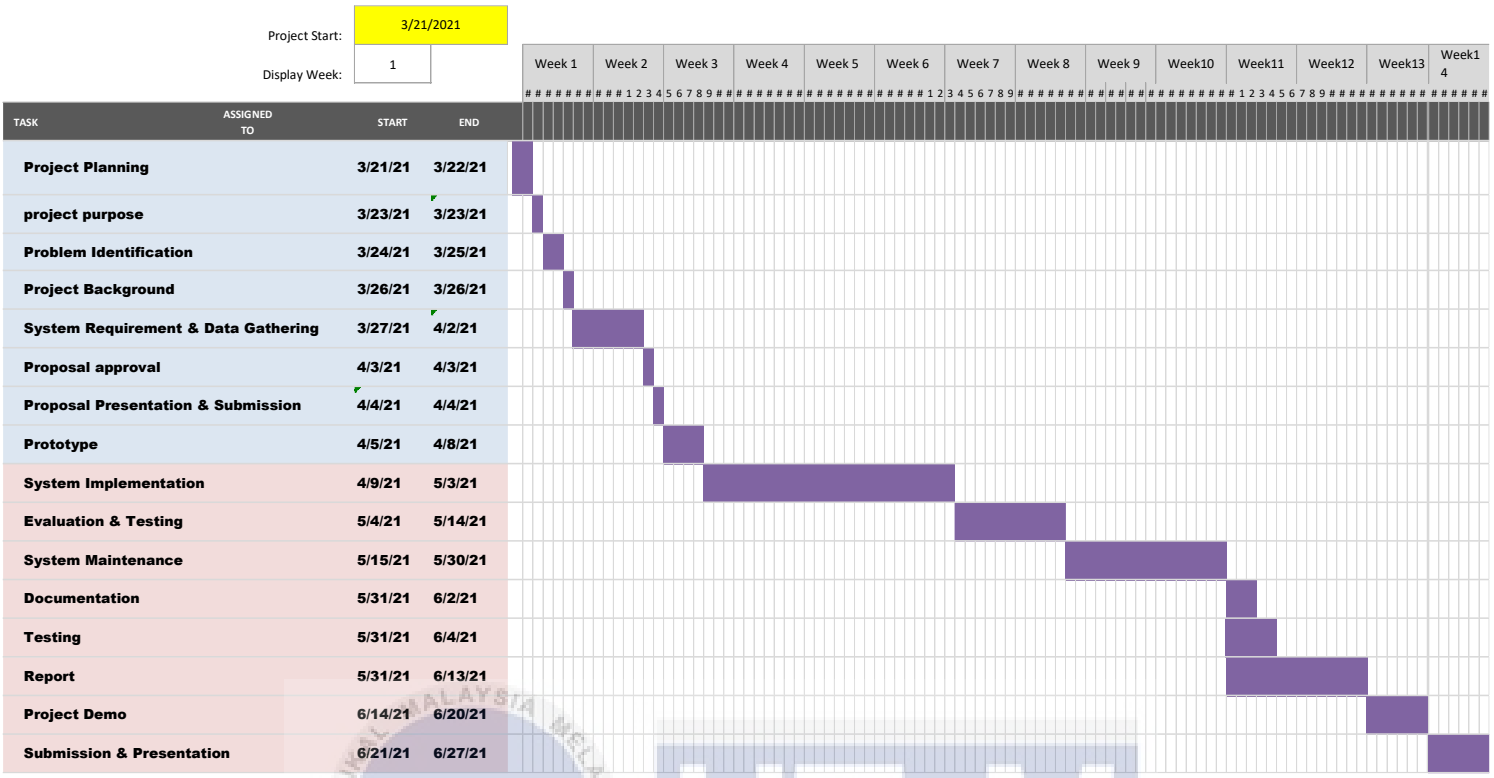


Figure 2. 23: Gantt Chart of ArithmeticNorm system

2.6 Conclusion

At the end of this chapter, we have discussed about project methodology that expected to achieve in ArithmeticNorm system and project planning of the ArithmeticNorm performance system. This chapter is a means of improving the management and control of the software development process, structuring and simplifying system processes by specifying system activities to be done and techniques to be used.

CHAPTER III

ANALYSIS

3.1 Introduction

This chapter aims to identify ArithmeticNorm goals, purposes, and procedures that will be achieved efficiently by collecting and interpreting facts, identifying the problems, and decomposition of the system into its components. System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. This chapter will provide description about functional, and non-functional requirements of ArithmeticNorm system.

3.2 Problem Analysis

In this section, we will discuss the tools of database normalization that are available for students and have been done by other researchers, and then will discuss about the proposed improvement or solutions that have been used in ArithmeticNorm system and explain the proposed technique that can develop in order to provide an easier and understood system for students to help them get more knowledge of database normalization as well as for the developer that can help better understanding the code.

Although normalization algorithms have been developed, very few commercial design tools are available to assist the normalization satisfactorily. Based on the literature review study that has been done in Table 2.1, it is found that current algorithms that are used in available database normalization tools do not provide a clear understanding of the university's research performance.

As shown in Figure 3.1, most of these normalization tools consider database relations and FDs as two distinct inputs and thus lead to partial automation and then represent these attributes, functional dependencies, and keys of a relation in the form of sets that stored as an array of strings. As a result, these tools do not provide a visual aid for normalization. Most of the authors of these tools claimed that this tool is having a positive impact on students.

One of these tools is normalization tool which developed by Griffith university. This tool asks the user to enter their attribute and all functional dependency for that attribute, then, the system determines candidate keys and check for 1NF, 2NF, 3NF, and BCNF. This system also normalizes into normal forms by showing functional dependencies (please refer to Table 2.1 no 9).

The current system does not have a clear understanding of normalization concepts due to that there are some students who do not have a clear understanding of the concept of functional dependencies. Therefore, they may face difficulties on find functional dependency for their database.

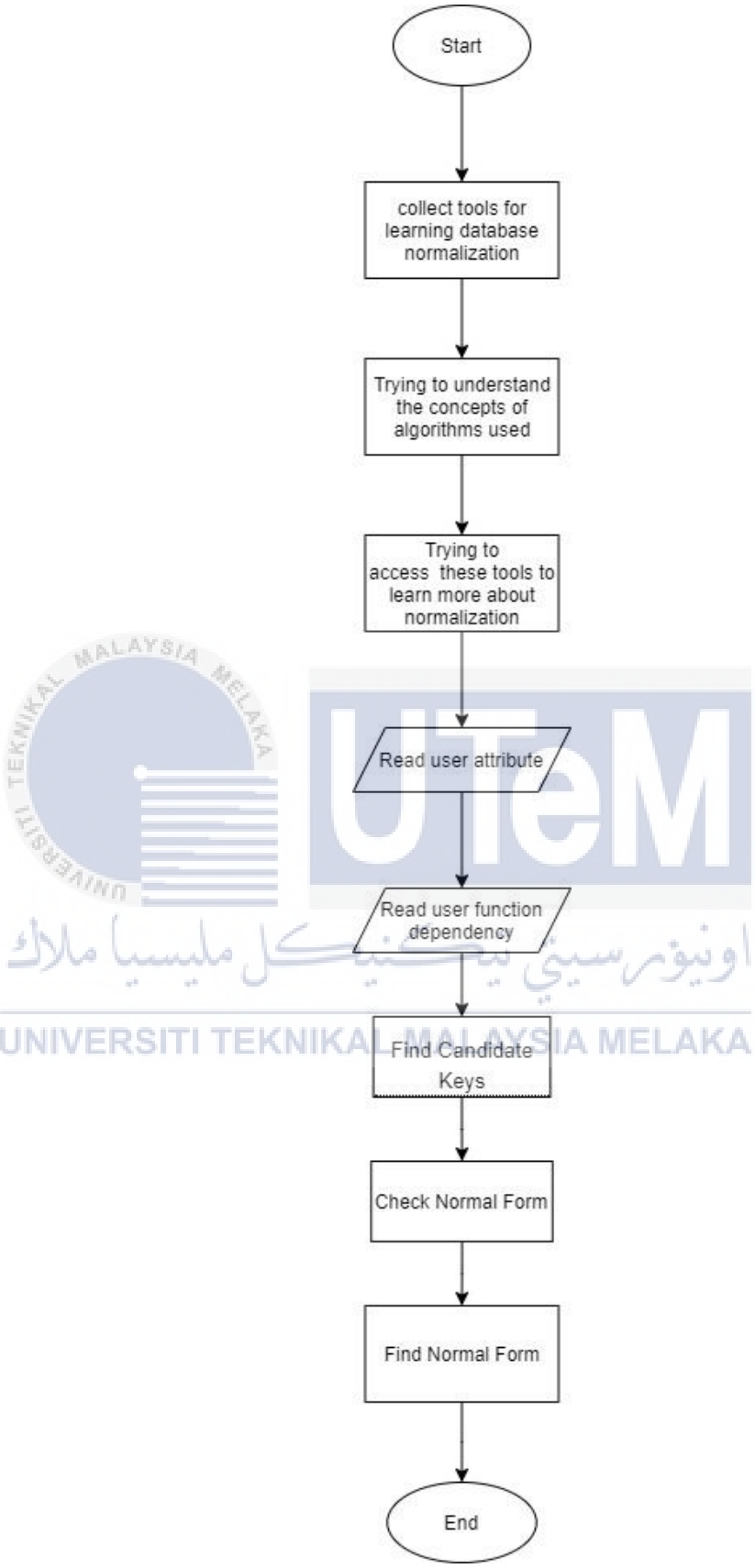


Figure 3. 1: Current System Flowchart

3.3 Proposed Improvements

The proposed solution in ArithmeticNorm system is to provide techniques performance for both programmers and users. These techniques depend on handling with Microsoft Excel arrays to find normal forms of the user's tables and to allows users upload their tables to provide a visual aid for normalization and provide a clear understanding of database normalization, ArithmeticNorm system expected to find normal forms based on the functional dependencies that will provided by users.

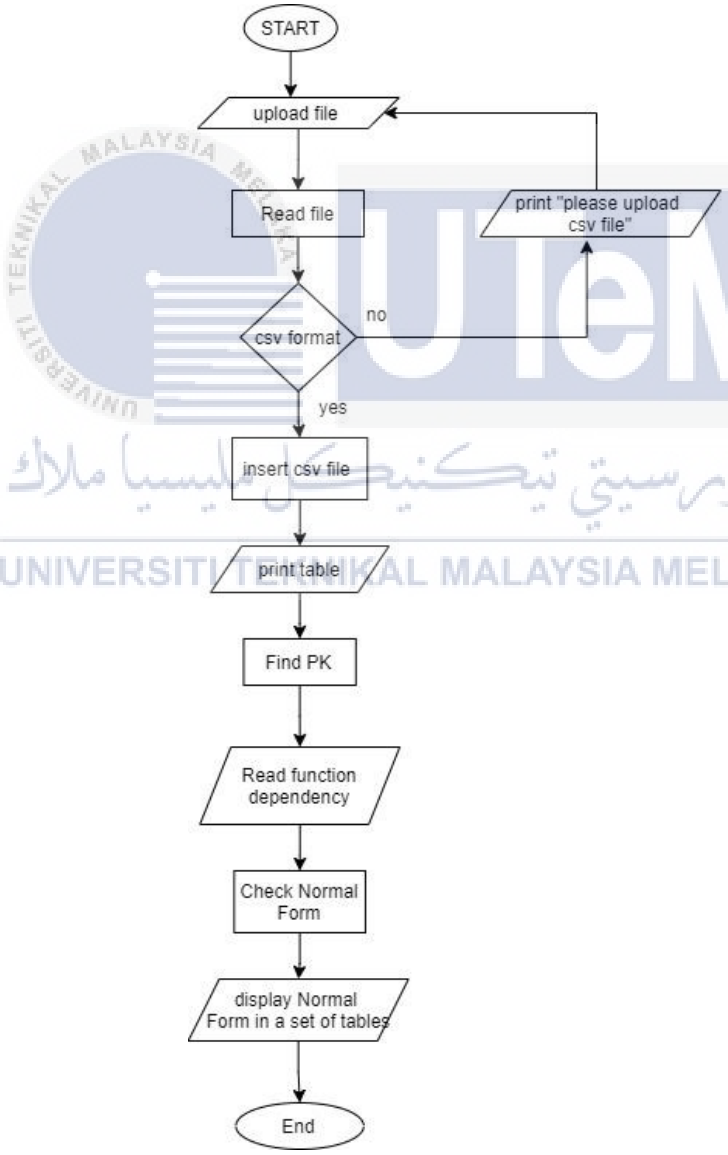


Figure 3. 2: Proposed Improvements Flowchart

3.3.1 Upload Table

To allow users to upload their table in CSV format and start check for the first normal form (1NF), there are some techniques that used in this system to achieve that. The first technique is to initialize the CSV format in the system which can determine if the user uploads another format except for the CSV format. The system starts to calculate the row and column number and go through all cells of the user table to check if the cell having null values or not. As shown in the Figure 3.3, if the table does not have null values, the second technique will start which is used to find whether the table has repeating columns' names or not.

These techniques help the system to find a repeating group and repeating columns name which is not allowed in the first normal form (1NF). After done of these techniques, the system uploads the user table and starts to determine primary key which is a rule of the first normal form (1NF).

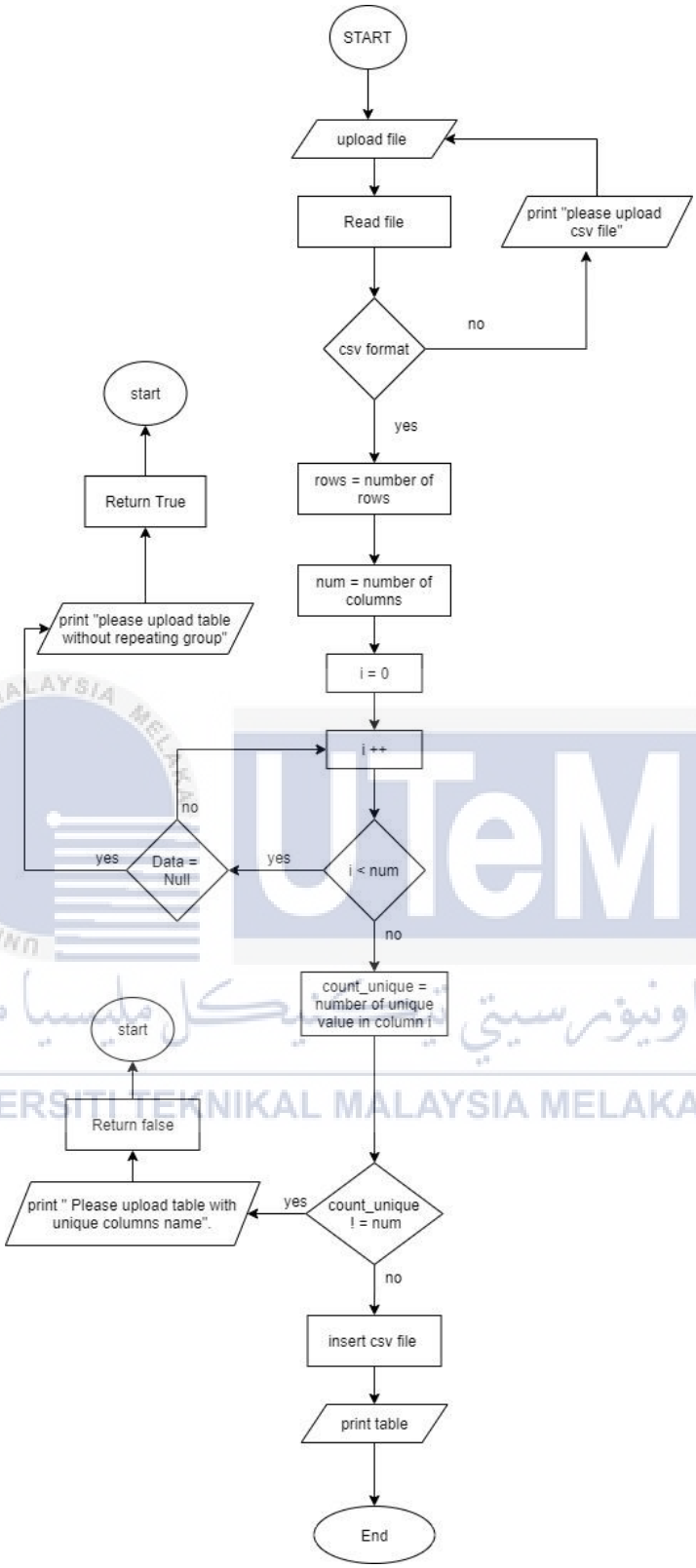


Figure 3. 3: Upload Table Flowchart

3.3.2 Find Primary Key

Technically, in order to identify if the attribute is unique, the approach that has been done in the literature review study (please refer to Table 2.1) is to calculate the total of rows for the selected attribute. If the total of the selected attribute is equal to the total rows of the table, therefore there is no redundancy value which also indicates to suitable candidate attribute for primary key as shown in the Figure 3.4.

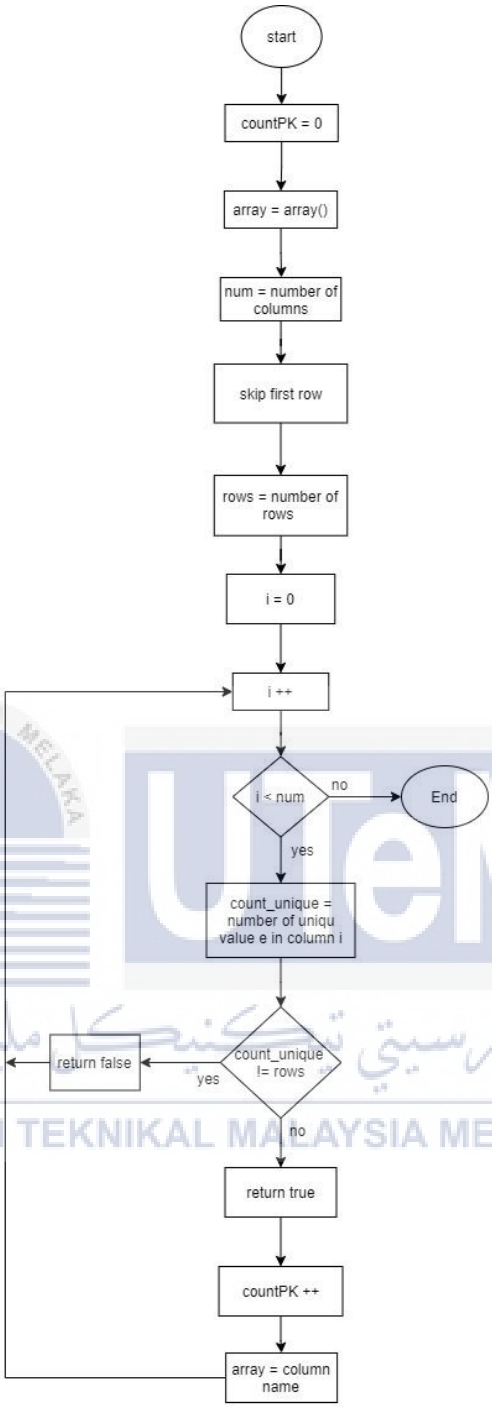


Figure 3. 4: Find Primary Key

3.3.3 Find Composite Key

A composite key is finding when there is no primary key in the table or when the primary key that finding by the system is not suitable to be the primary key. Technically, to determine the correct composite key, by referring to approach that has been done in the literature review study (see Table 2.1), for every two columns in the table the composited process performed which collect it in one value and store in an array. The unique value of these sets of composite keys has been counted. If the unique value equal to the number of rows, then the composite indicates to suitable candidate attribute for the new composite primary key.

Partial Dependency occurs when a non-prime attribute is functionally dependent on part of a candidate key. The 2nd Normal Form (2NF) eliminates the Partial Dependency. A set of partial dependency can be found using the same technique as the composite key technique. Except that the number of unique values of a set of composite keys for every two columns in a table is less than the rows number to indicate that these columns are a partial dependency.

As shown in the Figure 3.5, the flowchart has been developed to find composite key in order to find the suitable primary key.

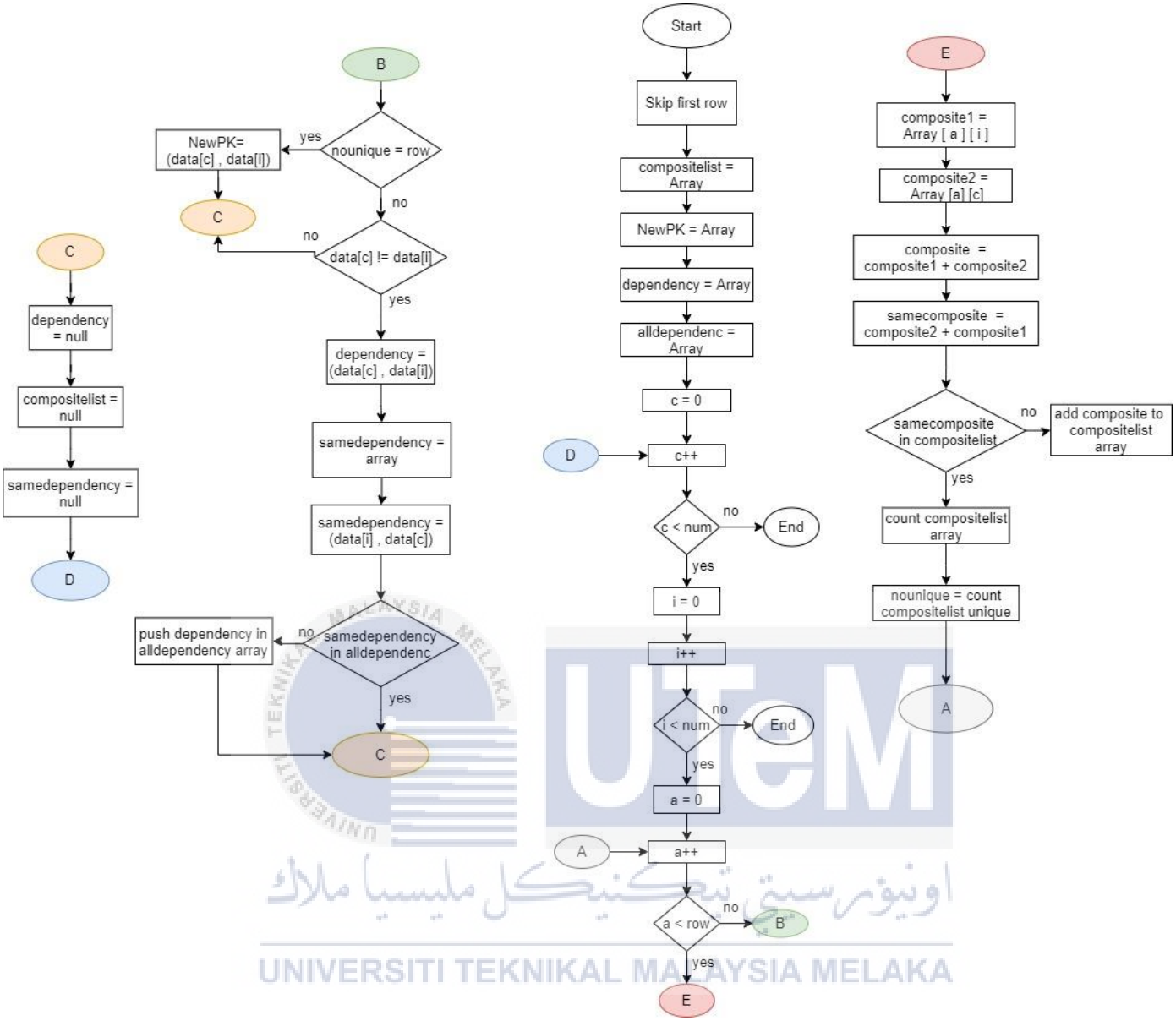


Figure 3. 5: Find Composite Key Flowchart

3.4 Requirement Analysis of the To-Be System

The analysis of system requirements is presented in this part. The system requirements analysis' goal is to structure the ArithmeticNorm system in a way that is independent of any implementation environment. The behavior and constraints of the system can be determined during this phase. The system requirements analysis activity is the overall process's second major development phase. Focusing solely on items and their relationships is a typical approach to object-oriented approaches. This section contains information on function requirements, non- function requirements and other requirements as written in Section 3.4.1, 3.4.2 and 3.4.3 respectively.

3.4.1 Function Requirement

The main function requirement of this system has been written in Section 3.3 explaining the process of how ArithmeticNorm tool works. In this section, explanation on additional and supporting features for this system is displayed.

This system consists of features that can manage media data which are images, audio, videos, and others, register new account, login to the system, design tables into normal forms, update user profile, and generate report.

i. Manage media data

Media data comes in a variety of formats, each of which has an impact on how students learn and comprehend information. The usage of media data in education allows students to gain access to more relevant information.

ii. User Registration and Login

The user must be able to register to the system with a valid username. If the username has been added to the system, the user can log in to the system. The user should be able to log in to the system once he/she has successfully registered to the system by his username.

iii. Design Tables into Normal Forms

To help students learn database normalization, the system must be able to guide students to design their tables into normal forms. The https://www.youtube.com/watch?v=Hy4eJaI6vXs is the guideline steps on what to do for this section.

iv. Generate Report

System is able to do report of all the system and managed by the admin.

3.4.2 Non-Function Requirement

This section describes the system operational capabilities and constraints that enhance its functionality. Non-Function requirements are as the following:

i. Scalability

Assesses the highest workloads under which the system will still meet the performance requirements.

ii. Reliability

The system must be able to have an 85 percentage for the system and its element to run without a failure for a given period.

iii. Maintainability

Maintainability is the amount of time it takes to fix, change, or adapt a solution or one of its components to improve performance or other attributes or to adapt to a changing environment.

iv. Usability

The user must be able to use the system easily and the system must be able to provide an easy system that helps the user to handle or use it easily.

3.4.3 Other Requirement

There is other two requirements that have been discussed in this section which are hardware requirement and software requirement.

3.4.3.1 Hardware Requirement

Below are the details of hardware requirements that used in order to develop ArithmeticNorm system**.**

- Processor Ryzen 4 and above.
- Memory 1000 GB.
- Hard Disk 350 GB.
- Monitor.
- Internet access.
- 9 GB RAM and above.

3.4.3.2 Software Requirement

Below are the details of software requirements that used to develop ArithmeticNorm system.

- XAMPP Server

  XAMPP is an abbreviation for cross-platform, Apache, MySQL, PHP and Perl, and we used to build WordPress sites offline, on a local web server on a computer.

- Notepad ++

  Notepad ++ are software for editors that used to design system interface in PHP and HTML.

- Windows 8/10

  Windows is one of the platforms that is used in developing the system. Windows is a very famous platform that is frequently used in public.

3.5 Research Methodology

The strategies or techniques used to locate, select, process, and analyze information regarding normalization topic have been discussed in this section. The methodology part of a research prototype allows the reader to assess the overall validity and reliability of the study. Two important questions are addressed in the methodology section: What method was used to gather or create the data? What method was used to analyze it? Figure 3.6 show the phases of research methodology that has been followed in this project step by step.

Figure 3. 6: Research Methodology Phases

i. Related Work

Related work allows us to demonstrate our understanding of the field while also allowing others to connect our current work to other scientific fields. By clearly describing previous work in Section 2.3, that section includes techniques that defined the problem, addressed a central or related problem, or applied a similar methodology to a similar challenge to our work.

ii. Preliminary Survey

The purpose of the preliminary survey is to gather sufficient data that show the goal of planning and design normalization tool as shown in Section 2.3, a brief inquiry has been done to investigate students' and lecturers 'opinions about the normalization topic using an online survey through Google form. This phase meets the first objective of this system.

iii. Develop Normalization Tool

Normalization tool has been developed to meets the second objective of this project which can be referred in Chapter 5. There are some algorithms that have been used in this phase based on the related work that has been done by other researchers. In addition to implements proposed improvements for current normalization tools. The normal forms algorithms that have used can be found in Appendix A.

iv. Evaluate Normalization Tool

Based on Chapter 6, the testing phase of this system has been done to evaluate the overall system and evaluate the normalization tool that has been developed as well. The results of users' survey who have been chosen to evaluate the normalization tool have been shown in Section 6.5.

3.6 Conclusion

As a conclusion of this chapter, description about functional, non-functional, and other requirements of ArithmeticNorm system has been clearly written. In addition to clearly explain of research methodology that has been followed in this project.

CHAPTER IV

DESIGN

4.1 Introduction

This chapter discusses the design of ArithmeticNorm system which consists of the design of UI components and a clearly defined visual style, architecture of a system that describes its major components, their relationships (structures), and how they interact with each other. This chapter also discusses database design which is a collection of processes that facilitate the designing, development, implementation, and maintenance of enterprise data management systems. The main objectives of database designing are to produce logical and physical design models of the proposed database system. In addition to discusses the data dictionary that used in the system.

4.2 System Architecture Design

ArithmeticNorm system uses three-tier architecture which are presentation tier, client tier, and database tier. Each tier can run on a separate operating system and server platform. In addition, each tier runs on at least one dedicated server hardware or virtual server, so the services of each tier can be customized and optimized without impact the other tiers.

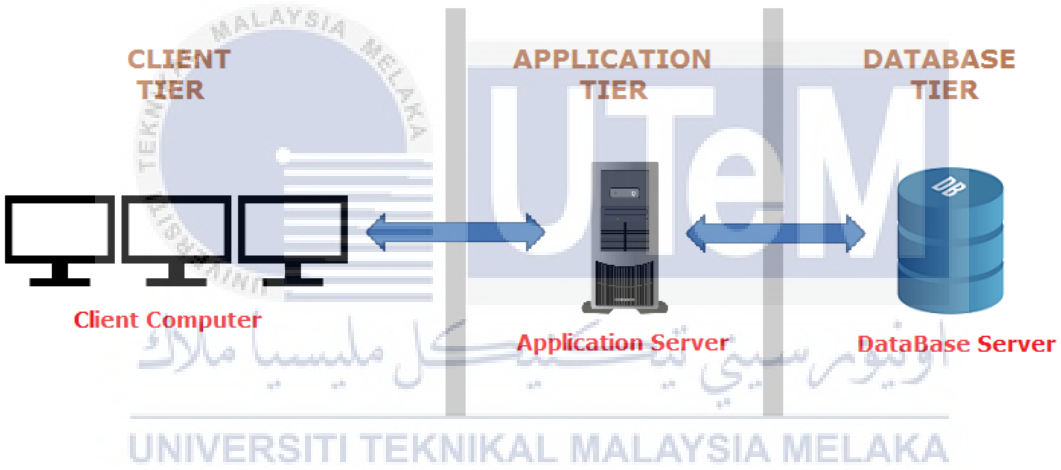


Figure 4. 1: Three-Tier Architecture

As shown in the Figure 4.1, the client tier is the application user interface and communication tier, where the end-user interacts with it to show information and gather data. It can be accessible by a web browser, a desktop application, or a graphical user interface (GUI). HTML, CSS, and JavaScript are used to create web presentation levels. The PHP language is used to create the application tier for the system, which may add, delete, and edit data in the data layer. The information processed by the application is stored and controlled in the database tire. The relational database management system that uses in the system is MySQL Database Service.

4.3 Database Design

There are three types of database design that will discuss in this section which are conceptual design, logical design, and physical design.

4.3.1 Conceptual Design

In this section, entity relation diagram (ERD) models will be described which is the business objects that should exist in the system and the relationships between them. The purpose of this section is to present an overall picture of the system by recognizing the business objects that involved in the system.

Figure 4.2 shown that the entity relation diagram (ERD) that has been developed for the system based on the following business rule:

a. A user can be either a normal user or an admin. Both have the same user information required by the system and can upload, update, or delete only one image that uses in their profile photo. They also can upload more than one CSV file that uses by the system to design their normal forms.

b. Admin users can manage only one audio file and can write more than one theory that helps students better-understanding database normalization. Each of these theories may contain one or more video files managed by the admin.

c. Admin can create more than one exam paper that contains at least one question, the questions must be multiple choice that contains many choices. Once students finish attends the exam the results and the exam details will be recorded automatically on the user page. All users are allowing to access their results record that can be more than one result.

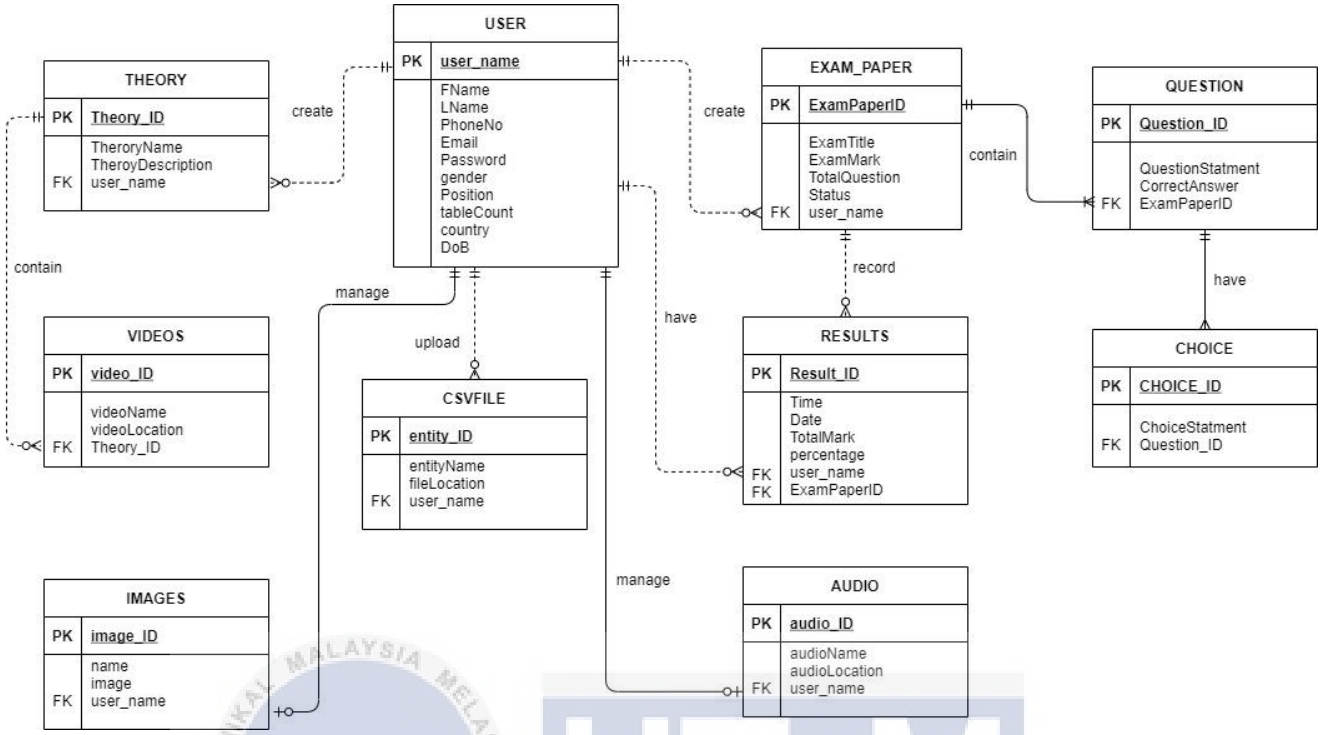


Figure 4. 2: Entity Relation Diagram (ERD) For ArithmeticNorm System

4.3.2 Logical Design

Logical database design is performed to determining the logical data structures that are required to support information resources within the system.[11] This section describes the implementation of the database that satisfies the requirements of the system and helps to ensure a successful implementation and discuss the data dictionary which is a collection of the names, definitions, and attributes for data elements and models that used as part of the database for the system. After that, database normalization will be shown at the end of this section.

[11] Broadcom, introduction to logical design.

4.3.2.1 Data Dictionary

Based on the entity relation diagram (ERD) that has been developed in conceptual design Section 4.3.1, there are 10 tables shown in that section, each of these tables must consist of the entity name, attribute name, data type, and constraint for each attribute (please refer to Table 4.1 up to Table 4.10).

Table 4. 1: Data Dictionary for User Table

USER					
ATTRIBUTE NAME	**CONTENT**	**DATA TYPE & LENGTH**	**FORMAT**	**PK/ FK**	**REFERENCE TABLE**
user_name	Username for user account	VARCHAR (20)	X999999999	PK	
FName	User first name	VARCHAR (10)	X999999999		
LName	User last name	VARCHAR (10)	X999999999		
phoneNo	User phone no	INT (20)	9999999999		
email	User email	VARCHAR (20)	X999999999		
password	Password for user account	VARCHAR (20)	X999999999		
position	User position	VARCHAR (10)	X999999999		
gender	User gender	VARCHAR (10)	X999999999		
tableCount	No of table that user created	INT (10)	9999999999		
country	User country	VARCHAR (10)	X999999999		
DoB	User date of birth	DATE	MM/DD/YYY		

Table 4. 2: Data Dictionary for Exam_paper Table

EXAM_PAPER					
ATTRIBUTE NAME	**CONTENT**	**DATA TYPE & LENGTH**	**FORMAT**	**PK/ FK**	**REFERENCE TABLE**
examPaperID	Auto generate exam paper id	VARCHAR (20)	9999999999	PK	
examTitle	Exam title name	VARCHAR (100)	X999999999		
examMark	Exam total mark	FLOAT (4,5)	99.99		
totalQuestion	Total questions	INT (20)	9999999999		
status	The status of exam paper	Varchar (10)	Open		
user_name	Username for user account	VARCHAR (20)	X999999999	FK	USER

Table 4. 3: Data Dictionary for Results Table

RESULTS					
ATTRIBUTE NAME	**CONTENT**	**DATA TYPE & LENGTH**	**FORMAT**	**PK/ FK**	**REFERENCE TABLE**
resultID	Auto generate result ID	INT (5)	9999999999	PK	
time	Exam time	TIME	11:00		
date	Exam date	DATE	MM-DD-YYY		
totalMark	User total mark	FLOAT (5,2)	99.99		
percentage	Percentage of user correct answers to all questions.	FLOAT (4,2)	99.99		
User_name	Username for user account	VARCHR (20)	X999999999	FK	USER
examPaperID	Exam paper ID	INT (5)	9999999999	FK	EXAM_PAPER

Table 4. 4: Data Dictionary for Question Table

QUESTION					
ATTRIBUTE NAME	**CONTENT**	**DATA TYPE & LENGTH**	**FORMAT**	**PK/ FK**	**REFERENCE TABLE**
questionID	Auto generate question ID	INT (5)	9999999999	PK	
Question_statment	Question statement	TEXT	X999999999		
CorrectAnswer	The Correct answer of the question	VARCHAR (200)	X999999999		
examPaperID	Exam paper ID	INT (5)	9999999999	FK	EXAM_PAPER

Table 4. 5: Data Dictionary for Choice Table

CHOICE					
ATTRIBUTE NAME	**CONTENT**	**DATA TYPE & LENGTH**	**FORMAT**	**PK/ FK**	**REFERENCE TABLE**
choiceID	Auto generate choice id	INT (5)	9999999999	PK	
choiceStatment	Choice statement	VARCHAR (100)	X999999999		
questionID	Question ID	INT(5)	9999999999	FK	QUESTION

Table 4. 6: Data Dictionary for Theory Table

THEORY					
ATTRIBUTE NAME	**CONTENT**	**DATA TYPE & LENGTH**	**FORMAT**	**PK/ FK**	**REFERENCE TABLE**
TheoryID	Auto generate theory ID	INT (10)	9999999999	PK	
theoryName	theory name	VARCHAR (250)	X999999999		
theoryDescription	Theory description	LONGTEXT	X999999999		
User_name	Username for user account	VARCHAR (20)	X999999999	FK	USER

Table 4. 7: Data Dictionary for Images Table

IMAGES					
ATTRIBUTE NAME	**CONTENT**	**DATA TYPE & LENGTH**	**FORMAT**	**PK/ FK**	**REFERENCE TABLE**
image_ID	Auto generate image ID	INT (10)	9999999999	PK	
name	Image name	VARCHAR (255)	X999999999		
Image	Image code	LONGTEXT	X999999999		
User_name	Username for user account	VARCHAR (20)	X999999999	FK	USER

Table 4. 8: Data Dictionary for Videos Table

VIDEOS					
ATTRIBUTE NAME	**CONTENT**	**DATA TYPE & LENGTH**	**FORMAT**	**PK/ FK**	**REFERENCE TABLE**
Video_ID	Auto generate video ID	INT (5)	9999999999	PK	
videoName	Video name	VARCHAR (255)	X999999999		
VideoLocation	Video location	VARCHAR (255)	X999999999		
Theory_ID	Theory ID	INT (10)	9999999999	FK	THEORY

Table 4. 9: Data Dictionary for Csvfile Table

CSVFILE					
ATTRIBUTE NAME	**CONTENT**	**DATA TYPE & LENGTH**	**FORMAT**	**PK/ FK**	**REFERENCE TABLE**
Entity_ID	Auto generate entity ID	INT (5)	9999999999	PK	
entityName	Table name	VARCHAR (10)	X999999999		
User_name	Username for user account	VARCHAR (20)	X999999999	FK	USER
Location	The location of Csv file	VARCHAR (255)	X999999999		

Table 4. 10: Data Dictionary for Audio Table

AUDIO					
ATTRIBUTE NAME	**CONTENT**	**DATA TYPE & LENGTH**	**FORMAT**	**PK/ FK**	**REFERENCE TABLE**
audio_ID	Auto generate audio ID	INT (10)	9999999999	PK	
audioName	Audio name	VARCHAR (10)	X999999999		
User_name	Username for user account	VARCHAR (20)	X999999999	FK	USER
audioLocation	The location of audio file	VARCHAR (255)	X999999999		

4.3.2.2 Database Normalization

In order to reduce data redundancy and improve data integrity in the system, database normalization is used which is the process of structuring the database and the process of simplifying the design of the database so that it achieves the optimal structure of this system.

Based on the entity-relationship diagram (ERD) that has been normalized in Section 4.3.1 (conceptual design), there are some normalizations that have been done for that entity-relationship diagram which include:

i. Upload csv file

1st Normal form

User (user_name, FName, LName, PhoneNo, Email, Password, gender, Position, entityName, fileLocation)

2nd Normal form

User (user_name, FName, LName, PhoneNo, Email, Password, gender, Position)

Csvfile (entity_ID, entityName, fileLocation, user_name)

ii. Create exam paper

1st Normal form

User (user_name, FName, LName, PhoneNo, Email, Password, gender, Position, ExamTitle, ExamMark, TotalQuestion, QuestionStatment, CorrectAnswer, ChoiceStatment)

2nd Normal form

User (user_name, FName, LName, PhoneNo, Email, Password, gender, Position)

Exam_paper (ExamPaperID, ExamTitle, ExamMark, TotalQuestion, QuestionStatment, CorrectAnswer, ChoiceStatment, user_name)

3rd Normal form

Exam_paper (ExamPaperID, ExamTitle, ExamMark, TotalQuestion, user_name)

Question (Question_ID, QuestionStatment, CorrectAnswer, ExamPaperID)

Choice (choice_ID, ChoiceStatment, Question_ID)

iii. Upload video

1st Normal form

User (user_name, FName, LName, PhoneNo, Email, Password, gender, Position, TheroryName, TheroyDescription, video_ID, videoName, videoLocation)

2nd Normal form

User (user_name, FName, LName, PhoneNo, Email, Password, gender, Position)

Theory (theory_ID, TheroryName, TheroyDescription, video_ID videoName, videoLocation, user_name)

3rd Normal form

Theory (theory_ID, TheroryName, TheroyDescription, user_name)

Video (video_ID, videoName, videoLocation, theory_ID)

iv. Record results

1st Normal form

User (user_name, FName, LName, PhoneNo, Email, Password, gender, Position, Time, Date, TotalMark, Grade, ExamPaperID)

2nd Normal form

User (user_name, FName, LName, PhoneNo, Email, Password, gender, Position)

Results (Result_ID, Time, Date, TotalMark, Grade, ExamPaperID, user_name)

4.3.3 Physical Design

During the physical design that have been implement in the system, the transformation of the conceptual design and logical design (entities into tables, the instances into rows, and the attributes into columns) has been implemented in this phase.

ArithmeticNorm system uses MYSQL database to translates logical data models into a set of SQL statements. Since MySQL is a relational database system, it is relatively easy to translate from a logical data model, such as the one that has been described in conceptual design and logical design (refer to Section 4.3.1 and Section 4.3.2). The purpose of using MYSQL in ArithmeticNorm system is that it provides comprehensive support for every system development need.

Some of the stored procedure and trigger have been implemented in the system using MYSQL as shown in Table 4.11 and Table 4.12.

Table 4. 11: Stored Procedure

Stored procedure	**Name of procedure**	**Table involved**
Basic DML	insert_results	Results
	Insert_user	User
	Insert_theory	Theory
	Insert_video	Videos
	Insert_ csvfile	csvfile
	Insert_question	Question
Total		**6**

Table 4. 12: Stored Trigger

Trigger Type		**Trigger Name**	**Table involved**	**Total**
After	Insert Update	countTotalQuestion	Questions, Exam_Paper	2
After	Delete			1
After	Insert	countUserTable	Csvfile, user	1
Total				**4**

4.4 Graphical User Interface Design (GUI)

ArithmeticNorm system allows users to directly interact with their devices and complete certain tasks such as upload media data, display normal form design, learn about database normalization, and so on by manipulating elements like icons and scroll bars, which is one-way designers make their digital devices more efficient and usable.

The system uses PHP and HTML language to design a suitable user interface for users based on the system requirements as shown in Figures 4.3 up to Figure 4.23.

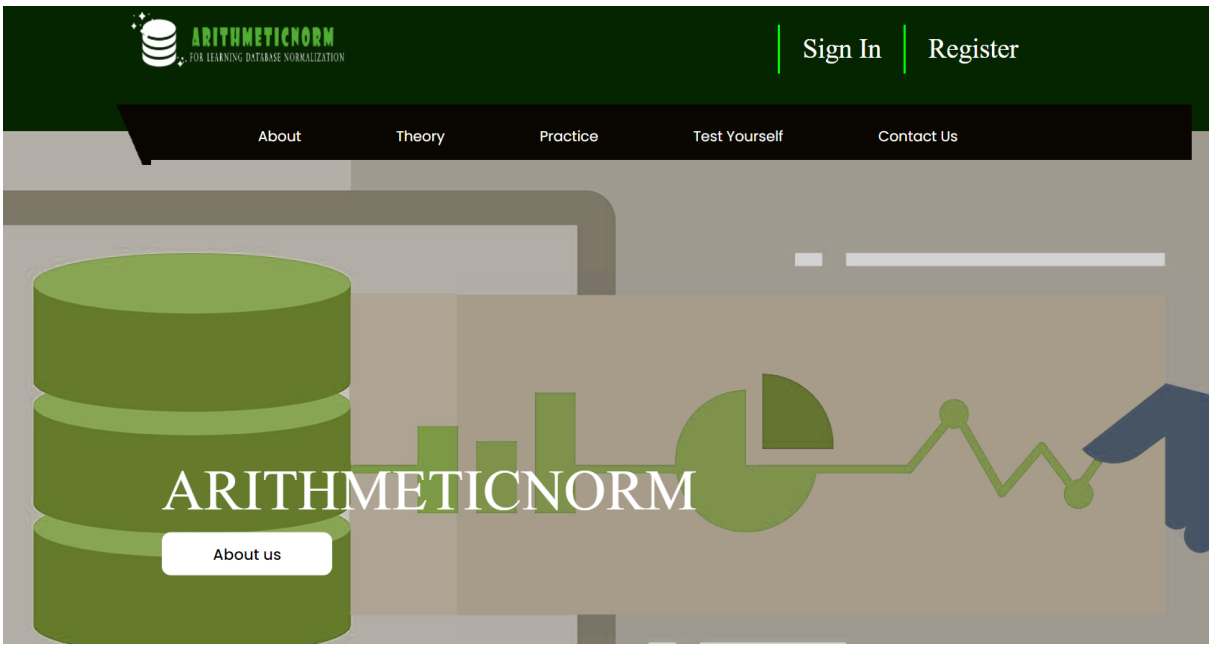


Figure 4. 3: ArithmeticNorm System Home Page

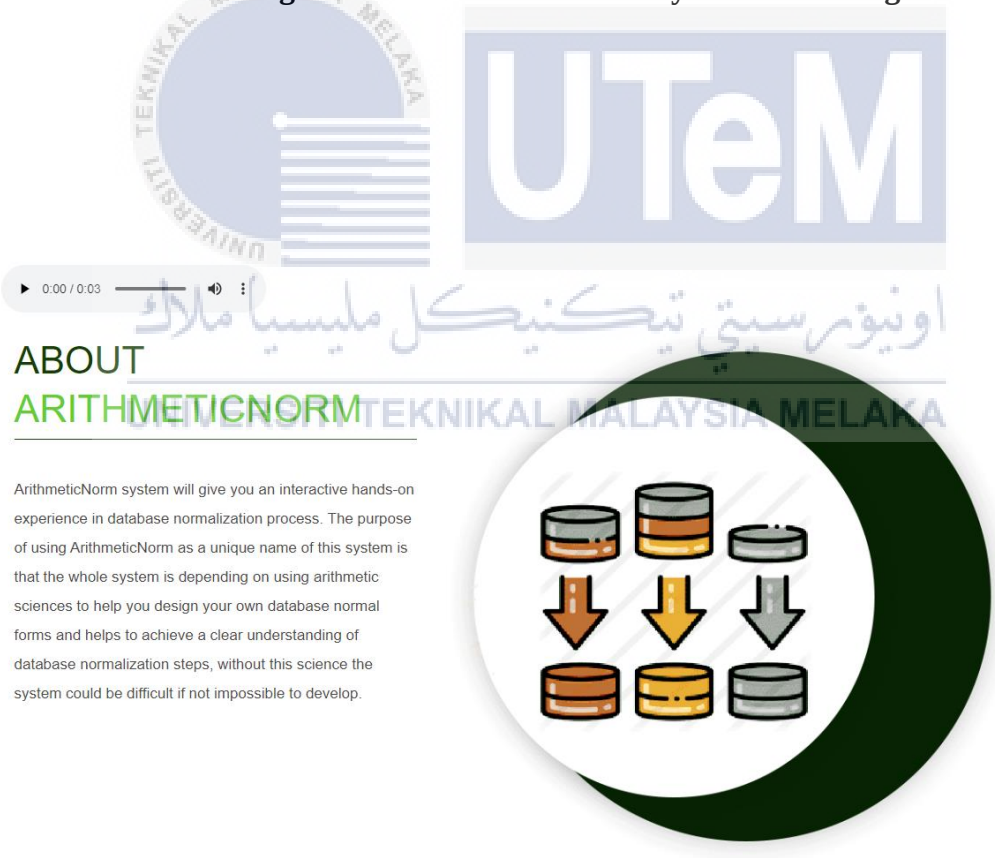


Figure 4. 4: ArithmeticNorm System Home Page (About)

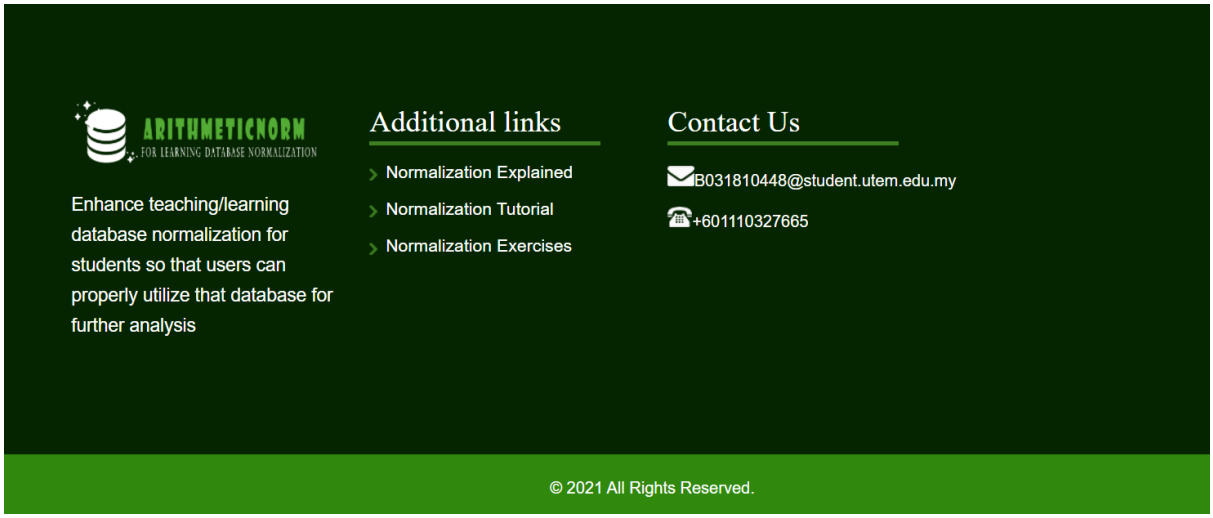


Figure 4. 5: ArithmeticNorm System Home Page (Contact Us)

Figure 4. 6: Sign In page

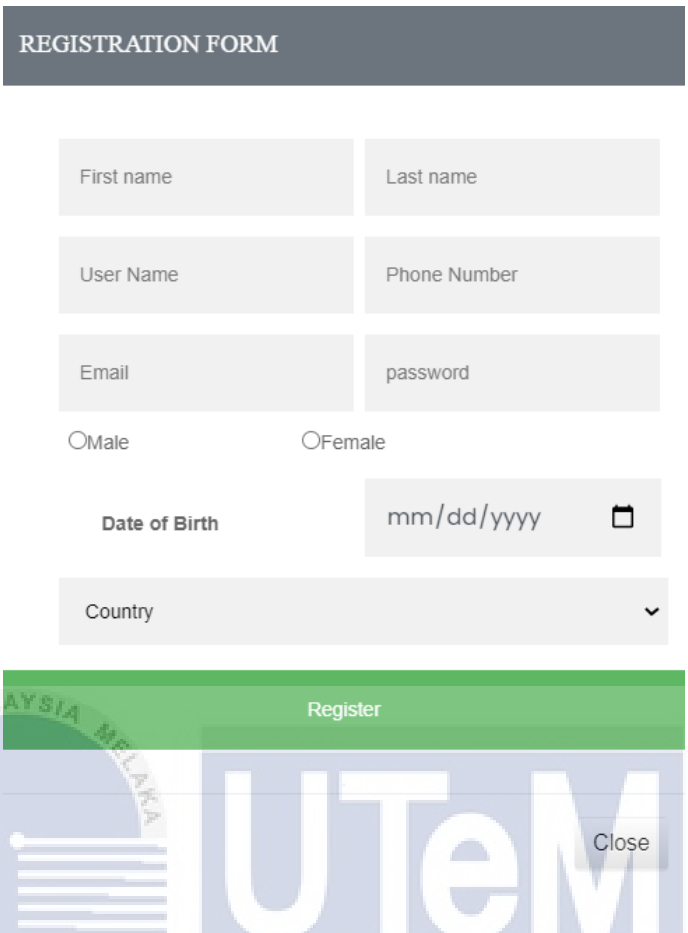


Figure 4. 7: Registration page

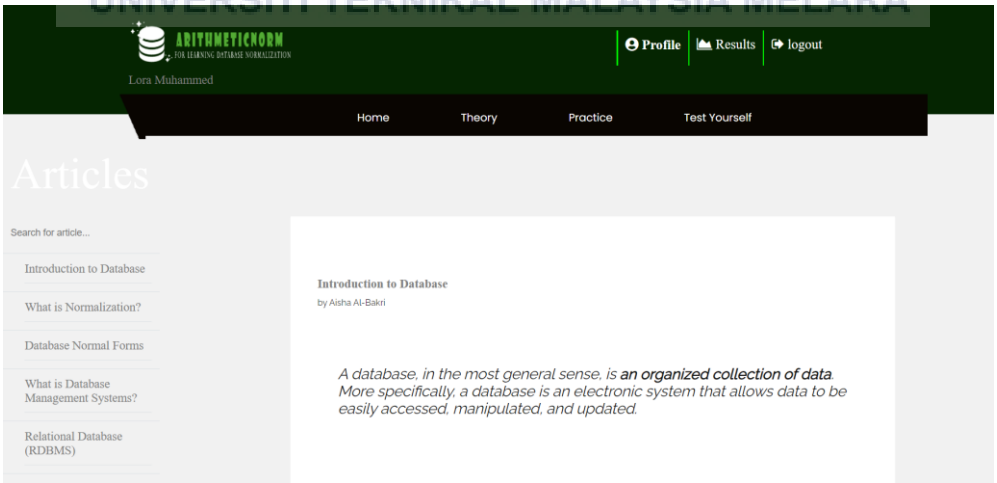


Figure 4. 8: Theory Page

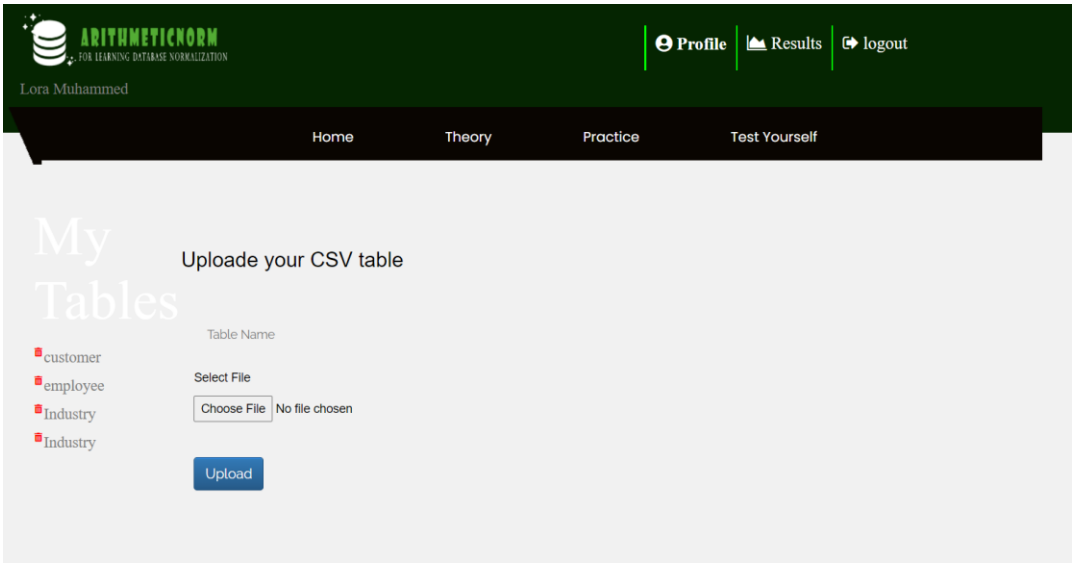


Figure 4. 9: Practice Page

Figure 4. 10: Student Home Page

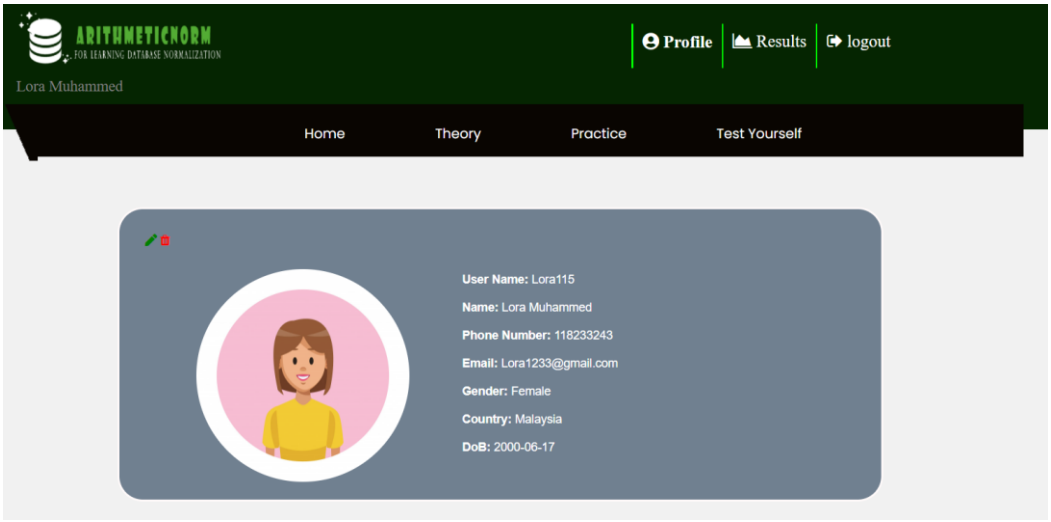


Figure 4. 11: User Profile Page

Figure 4. 12: Admin Home Page

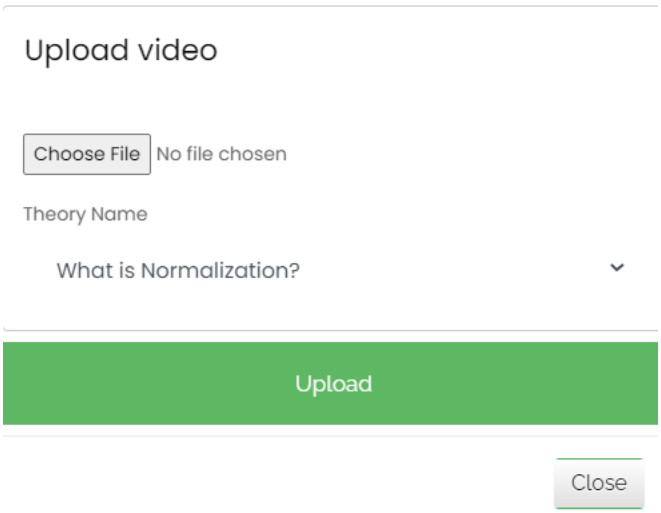


Figure 4. 13: Upload Video Page

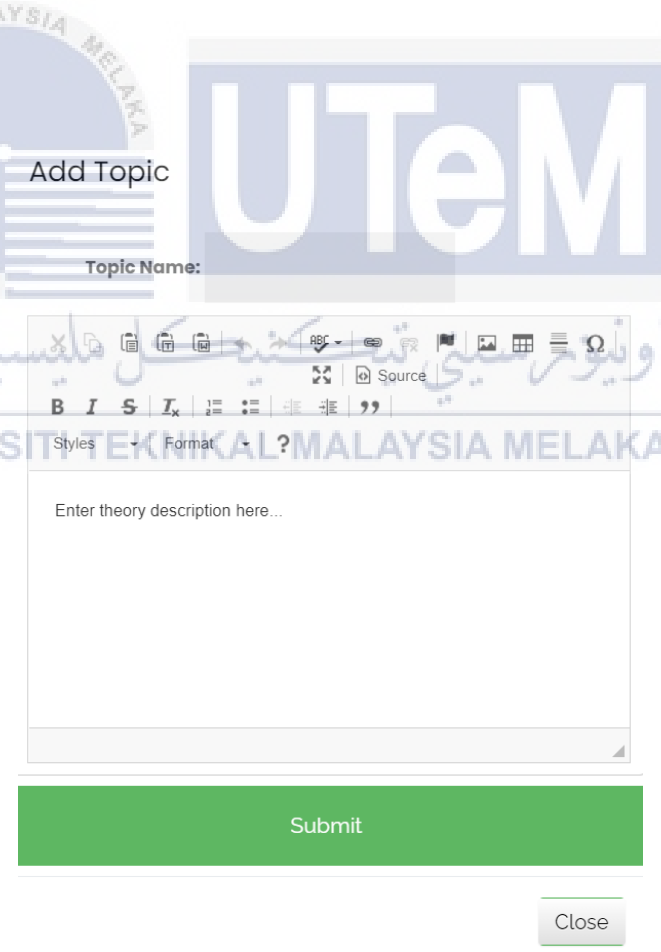


Figure 4. 14: Add Theory Page

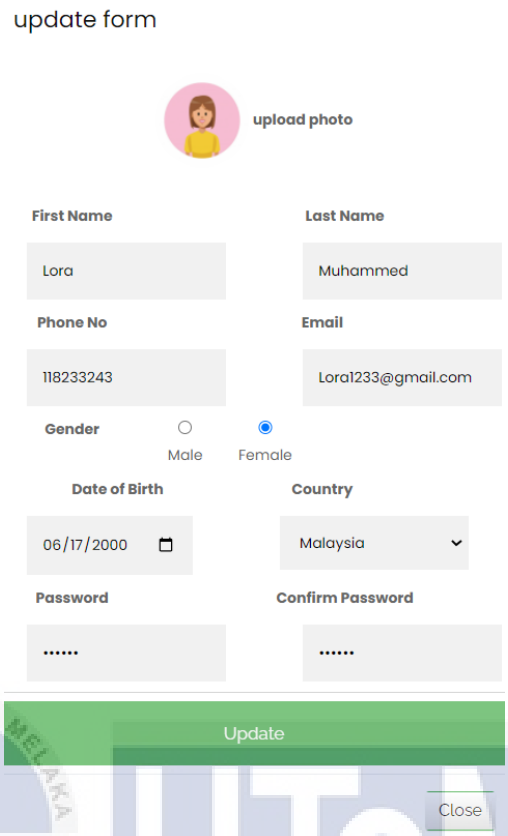


Figure 4. 15: Update Profile Page

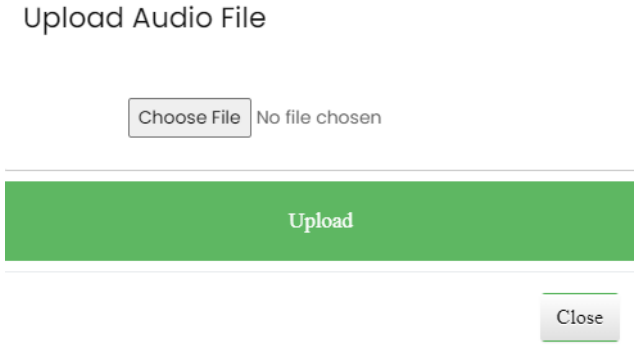


Figure 4. 16: Upload Audio Page

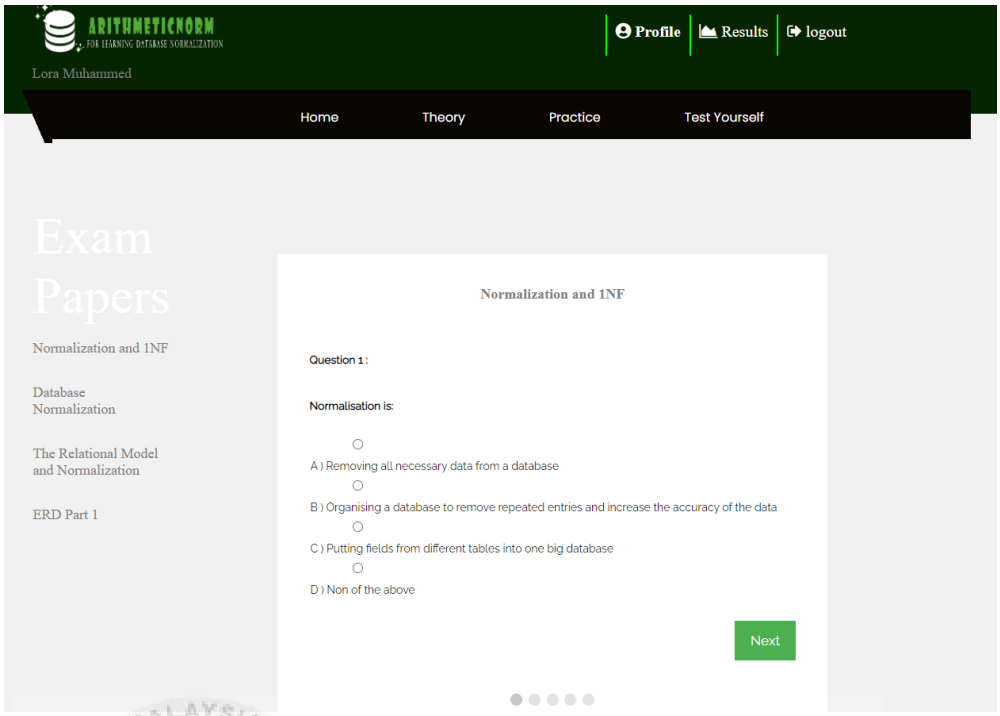


Figure 4. 17: Test Yourself Page

Figure 4. 18: Results Page

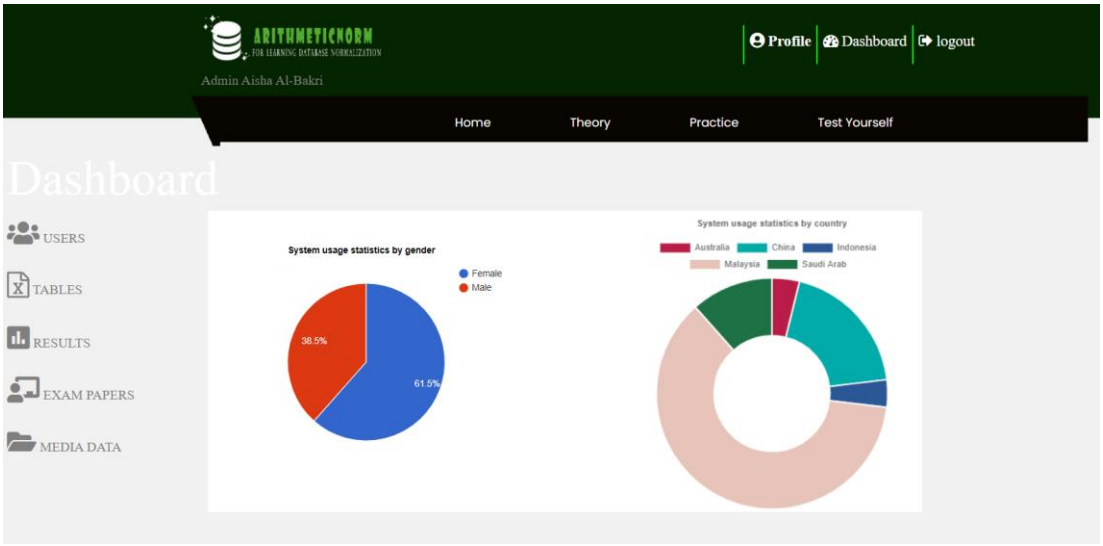


Figure 4. 19: Dashboard Page

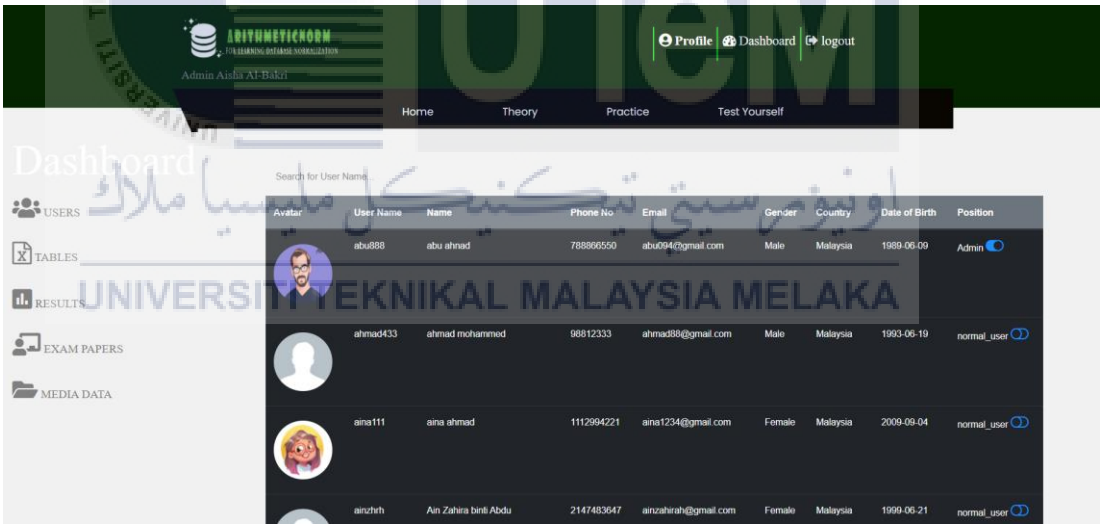


Figure 4. 20: Analysis of Users

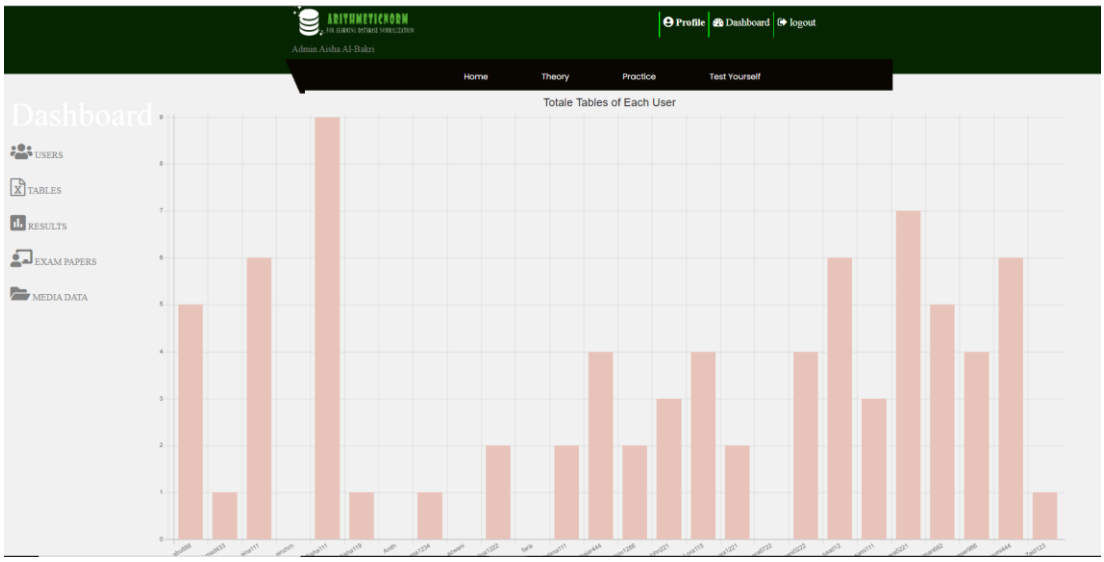


Figure 4. 21: Analysis of Tables

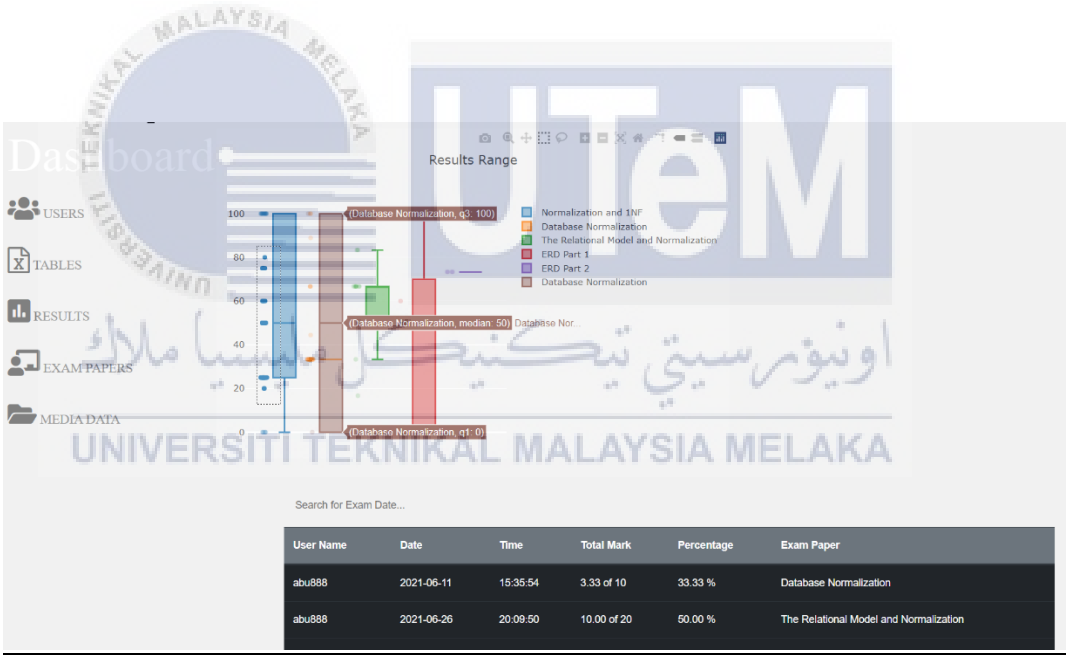


Figure 4. 22: Analysis of Results

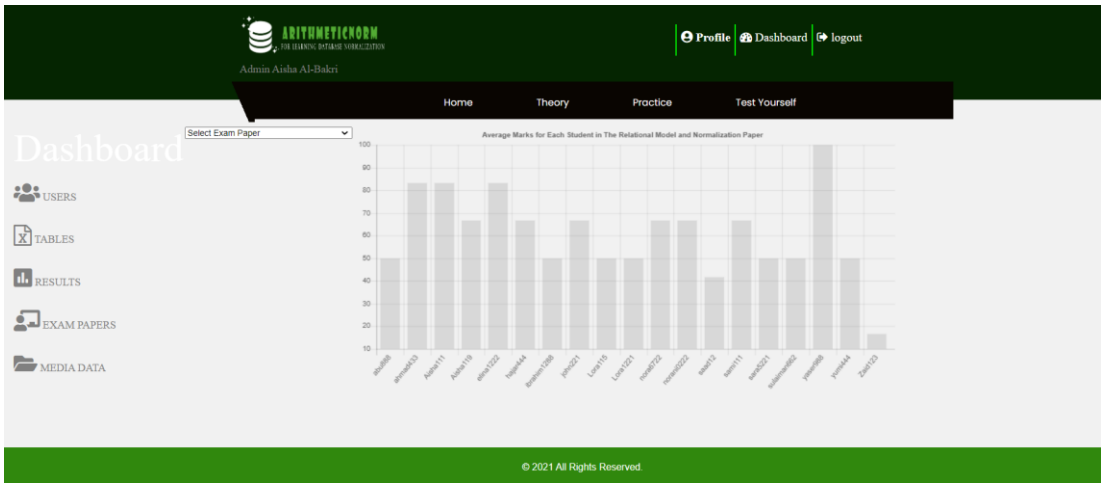


Figure 4. 23: Analysis of Exam Papers

Figure 4. 24: Analysis of Media Data

4.5 Conclusion

In a conclusion, system design is the process of defining the architecture, product design, modules, interfaces, and data for the system to satisfy system requirements. PHP is the most popular scripting language for web development, and MySQL is a relational database management system (RDBMS) that uses structured query language (SQL). PHP and MYSQL working together to building ArithmeticNorm system that is both suitable to meet system requirements which is a better choice to design graphical user interface design (GUI) and build system database that has been discussed in this chapter.

Chapter 5 will describe the system implementation and explain how the information system should be built.

CHAPTER V

IMPLEMENTATION

5.1 Introduction

This chapter is about the implementation phase that has been involved in developing ArithmeticNorm system which includes all the products or services that perform for the customer and all the project management documents. The explanation of this chapter will include database installation steps, database implementation that include DDL statements and MYSQL database, system development environment, and database software which include stored procedures and triggers using SQL language.

5.2 System Development Environment Setup

This section will describe the initial setup of the ArithmeticNorm system which includes all the components the system needs to develop. Based on the architecture design that has been described in Chapter 4, ArithmeticNorm system uses a three-tier architecture which is CSS, HTML, and JavaScript as the presentation layer, PHP as the application layer, and MYSQL as the database layer. This system uses Apache web server services as localhost and phpMyAdmin tool to store system data, both of them are included in the XAMPP package.

5.2.1 Software Environment Setup

XAMPP software is used to set up the ArithmeticNorm system environment which collection of software contains a web server include Apache, MariaDB, MYSQL server, and programming languages. XAMPP can be downloaded freely from the site www.sourceforge.net/projects/xampp/files/latest/download. Figure 5.1 up to Figure 5.6 show the steps of installing XAMPP software.

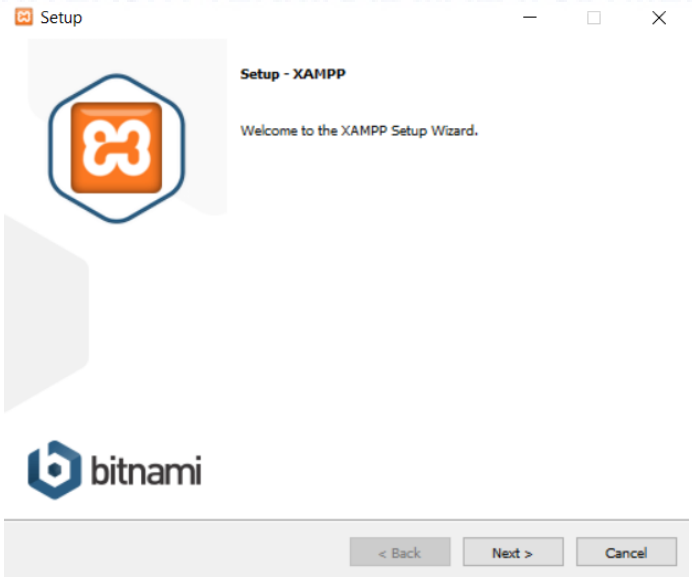


Figure 5. 1: XAMPP Installation Start

Figure 5.1 shows first step of XAMPP installation process, user have to click Next button to start the installation.

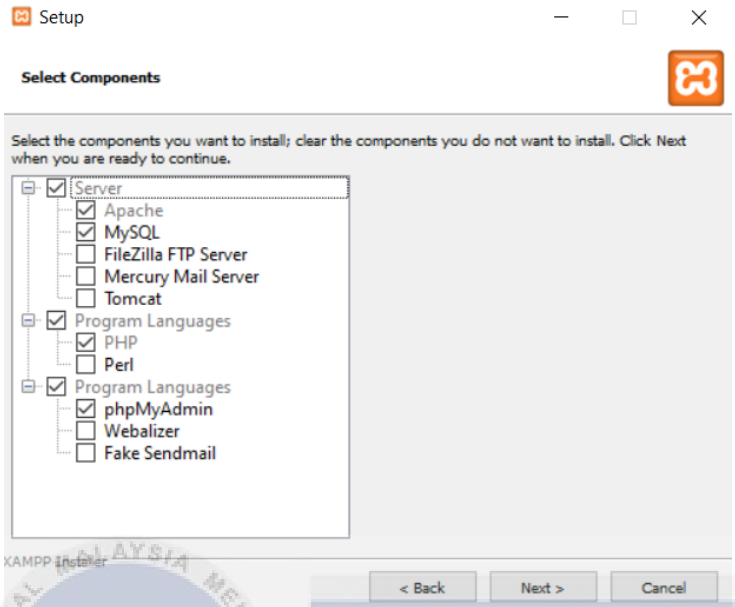


Figure 5. 2: Select Components of Server

Figure 5.2 shows the services that can install. This system uses Apache web server, MYSQL server, and phpMyAdmin as database interface. Then, click the Next button.

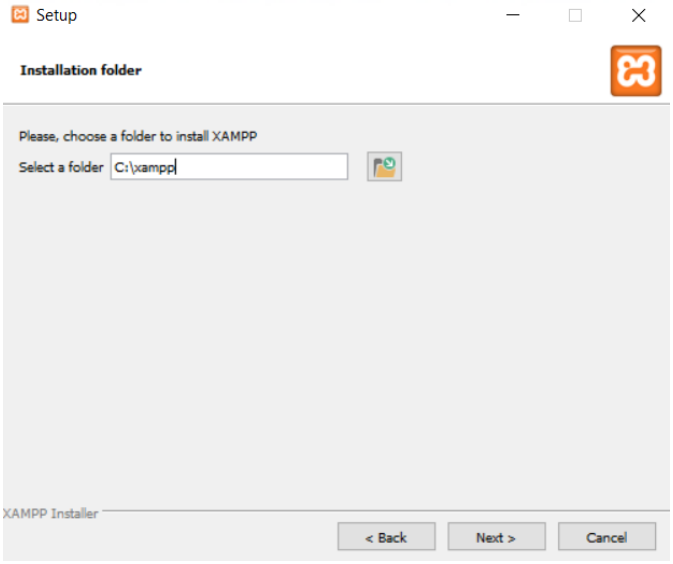


Figure 5. 3: Select the Installation Folder Location

Figure 5.3 shows the selection of installation folder location. As default, the installation of XAMPP is placed c:/xampp folder location.

Figure 5. 4: Installation Process

Figure 5.4 shows step to continue the installation after all the required settings have been done. The installation will take several minutes to complete.

Figure 5. 5: Completing the XAMPP setup

Figure 5.5 shows the success of XAMPP installation process. User have to click finish button to start XAMPP services.

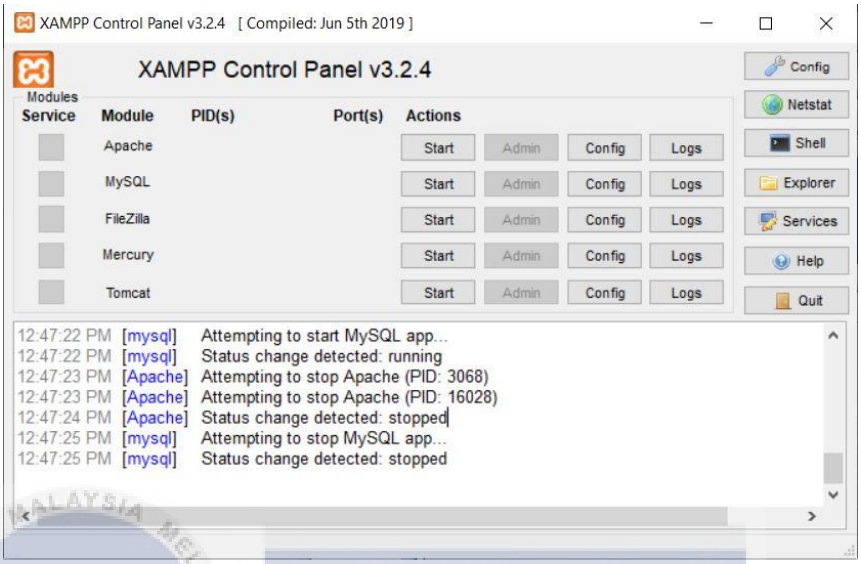


Figure 5. 6: XAMPP Control Panel

Figure 5.6 shows XAMPP control panel that use to running and to start or stop XAMPP package.

5.2.2 Database Environment Setup

The database will organize structured data, which is normally stored electronically in a computer system. A database management system is usually in charge of a database (DBMS). A database system is made up of the data, the database management system, and the applications that go with it. PhpMyAdmin has been used as DBMS in this system which is one of the packages that exists in XAMPP software, and it is one of the most widely used MySQL database administration apps.

After download and install XAMPP software, PhpMyAdmin will be available to use by access it through XAMPP control panel or by typing localhost/phpMyAdmin on the browser as shown in the Figure 5.7.

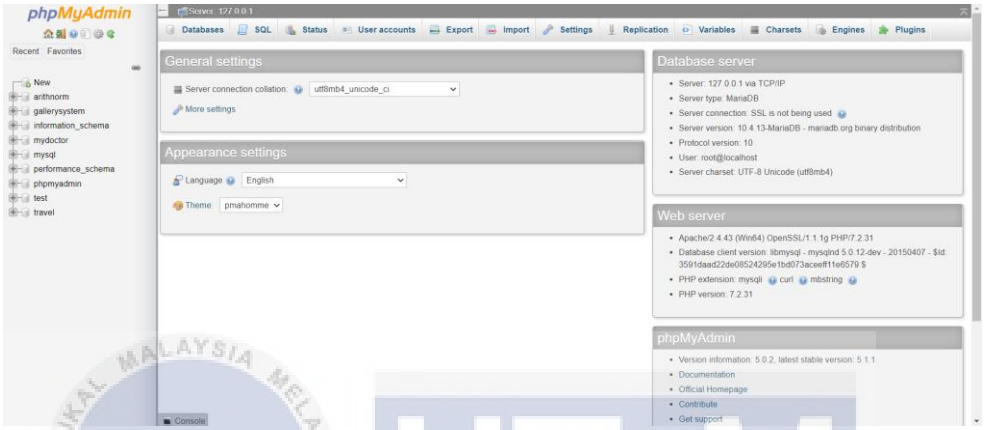


Figure 5. 7: PhpMyAdmin Interface

Figure 5.7 shows the PhpMyAdmin graphical user interface (GUI) which is a PHP-based HTML interface for establishing and maintaining MySQL databases that has been used in this system. PhpMyAdmin also used to create tables, stored procedures and create trigger.

5.3 Database Implementation

This section will describe how MYSQL query and command are implemented during the development of the system. All the tables, the procedures and the triggers that have been created in this system will be clarify in this section

i. CREATE TABLE Clause

The tables have been created based on the ERD that have been discussed in Chapter 4. Figure 5.8 shows example of create table syntax.

```
CREATE TABLE `user` (
        `user_name` varchar(20) PRIMARY KEY NOT NULL,
        `FName` varchar(10) NOT NULL,
        `LName` varchar(10) NOT NULL,
        `phoneNo` int(20) NOT NULL,
        `email` varchar(20) NOT NULL,
        `password` varchar(10) NOT NULL,
        `position` varchar(20) NOT NULL,
        `gender` varchar(10) NOT NULL,
        `totalimages` int(10) NOT NULL,
        `country` varchar(10) NOT NULL,
        `DoB` date NOT NULL
      ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Figure 5. 8: Create Table Clause

ii. SELECT Statement

There are many SELECT statements queries that have been used in this project to catch data from the database as shown in the Figure 5.9.

```
SELECT * FROM user;
```

Figure 5. 9: Select Statement for user Table

iii. WHERE Clause Statement

WHERE statement is uses to fetch data with condition from the database. Figure 5.10 shows example of WHERE statement that have been used in this system.

```
SELECT * FROM user WHERE user_name = “Aisha111”;
```

Figure 5. 10: SELECT Statement with WHERE Clause

iv. TRIGGER Clause

The TRIGGER clause will be executed automatically when a certain event occurs in a certain table. Based on the triggers list that has been done in Table 4. 12, there are 4 triggers that have been implemented in this system. Figure 5.11 up to Figure 5.14 show these triggers.

```
BEGIN
DECLARE vcount INT;
DECLARE vexampaperID INT;

SELECT examPaperID INTO vexampaperID
FROM question
WHERE questionID = NEW.questionID;

SELECT count(questionID) INTO vcount
FROM question
WHERE examPaperID = vexampaperID
GROUP BY examPaperID;

UPDATE exam_paper
SET totalQuestion= vcount
WHERE examPaperID = vexampaperID;
END;
```

Figure 5. 11: Trigger After Insert to Question Table

```
BEGIN
DECLARE vcount INT;
DECLARE vexampaperID INT;

SELECT examPaperID INTO vexampaperID
FROM question
WHERE questionID = NEW.questionID;

 SELECT count(questionID) INTO vcount
FROM question
WHERE examPaperID = vexampaperID
GROUP BY examPaperID;

UPDATE exam_paper
SET totalQuestion= vcount
WHERE examPaperID = vexampaperID;
END;
```

Figure 5. 12: Trigger After Update on Question Table

```
BEGIN
DECLARE vcount INT;
DECLARE vexampaperID INT;

SELECT examPaperID INTO vexampaperID
FROM question
WHERE questionID = OLD.questionID;

SELECT count(questionID) INTO vcount
FROM question
WHERE examPaperID = vexampaperID
GROUP BY examPaperID;

UPDATE exam_paper
SET totalQuestion= vcount-1
WHERE examPaperID = vexampaperID;

END;
```

Figure 5. 13: Trigger Before Delete from Question Table

```
BEGIN
DECLARE vcount INT;
DECLARE vusername INT;



SELECT count(entity_ID) INTO vcount
FROM csvfile
WHERE user_name = NEW.user_name
GROUP BY user_name;

UPDATE user
SET tableCount= vcount
WHERE user_name = NEW.user_name;

END;
```

Figure 5. 14: Trigger After Insert to csvfile Table

iv. Stored Procedure

Stored procedure is used to control the operation of the database that need to call or invoke to run inside the system. Based on the stored procedures list that has been done in Table 4. 11, Figure show the syntax of these procedures that have been done in the system.


```
CREATE OR REPLACE PROCEDURE insert_results (
        IN time VARCHAR(10),IN date VARCHAR(10), IN totalMark
        float(5,2), IN percentage VARCHAR(10), IN user_name
        VARCHAR(20), IN examPaperID INT(5)
)

BEGIN

INSERT INTO results(time, date,totalMark, percentage, user_name,
examPaperID)
VALUES(time, date, totalMark, percentage, user_name, examPaperID);
END insert_results;
```

Figure 5. 15: Stored Procedure to Insert Results Data

```
CREATE OR REPLACE PROCEDURE insert_user (
        IN user_name VARCHAR(20), IN FName VARCHAR(10), IN
        LName VARCHAR(10), IN phoneNo VARCHAR(20), IN email
        VARCHAR(20), IN password  VARCHAR(20), IN position
        VARCHAR(20), IN gender VARCHAR(20), IN country
        VARCHAR(20), IN DoB VARCHAR(20)
)

BEGIN

INSERT INTO user
        (user_name,FName,LName,phoneNo,email,password,position,
        gender, country,DoB)
VALUES
        (user_name, FName, LName, phoneNo , email, password, position,
        gender, country, DoB)";

END insert_user;
```

Figure 5. 16: Stored Procedure to Insert User Data

```
CREATE OR REPLACE PROCEDURE insert_theory (
        IN theoryName VARCHAR(250),IN theoryDescription longtext, IN
        user_name VARCHAR(20)
)

BEGIN

INSERT INTO theory
(theoryName,theoryDescription,user_name) VALUES
(theoryName,theoryDescription,user_name);



END insert_theory;
```

Figure 5. 17: Stored Procedure to Insert Theory Data

```
CREATE OR REPLACE PROCEDURE insert_video(
        IN videoName VARCHAR(250),IN videoLocation longtext,
        IN theory_ID int(10)
)

BEGIN

INSERT INTO videos (videoName,videoLocation,theory_ID) VALUES
                        (videoName,videoLocation,theory_ID);

END insert_video;
```

Figure 5. 18: Stored Procedure to Insert Video Data

```
CREATE OR REPLACE PROCEDURE insert_csvfile(
        IN entityName VARCHAR(50),IN user_name VARCHAR(20),
        IN location VARCHAR(255)
)
BEGIN

INSERT INTO csvfile (entityName,user_name,location) VALUES
        (entityName,user_name,location);

END insert_csvfile;
```


Figure 5. 19: Stored Procedure to Insert CSV File

```
CREATE OR REPLACE PROCEDURE insert_question(

        IN question_statment longtext,IN correctAnswer VARCHAR(200),
        IN examPaperID int(10)
)


INSERT INTO question (question_statment,correctAnswer,examPaperID)
VALUES (question_statment,correctAnswer,examPaperID);

END insert_question;
```

Figure 5. 20: Stored Procedure to Insert Question Data

5.4 Conclusion

In a conclusion, the implementation phase has been explained in this chapter which can be regarded as the process of setting up the system environment and installing the database server that uses to store the data or information for easy searching and retrieval.

The next chapter will discuss the testing phase of ArithmeticNorm system which will test the validation and verify the requirements.

CHAPTER VI

TESTING

6.1 Introduction

In this chapter, a testing phase has been implemented to ensure that users will get well system environment, to discover and eliminate errors that may occur in the system, and to find out whether the system meets all the requirements and objectives stated earlier in Chapter 3. In addition to describing the questionnaire that was used to test the system which is used to collect data and getting opinions from students regarding the system among FTMK students by using Google form.

There are four procedures that include in the testing phase which are test plan, test strategy, test implementation, and test result and analysis. The test plan section consists of three phases that have been discussed which are test organization, test environment, and test schedule.

6.2 Test Plan

The test plan is essential in the development of this system since it explains what needs to be tested to ensure that the system is performing as expected. The test plan helps to track the progress of the project and to achieve successful and productive testing as well. In this section, the test organization, test environment, and test schedule have been explained.

6.2.1 Test Organization

Based on the survey that has been done in Chapter 1, there are some people who volunteer to test the ArithmeticNorm performance system. In addition, there are some people from UTeM university staff and students who have been chosen randomly from different backgrounds to test ArithmeticNorm performance system. The feedback from these testers has been taken and noted for further improvements.

ii. System Developer
Responsible to test all the functionality of the requirements as well as for any errors identified.

iii. Administrator
Responsible for monitoring the system performance and managing the activities of the system.

iv. End User

Responsible for testing the system GUI to find out whether the system is user-friendly, and it is easy to use or not.

6.2.2 Test Environment

The test environment is the setup of the software, hardware, and supporting tools need to conduct a test Arithmetic Norm performance system. Table 6.1 shows the test environment of this system.

Table 6. 1: Test Environment Setup

Environment	Specification
Operating System	Microsoft Windows 10
Hardware	Processor: Intel Core i5 RAM: 4GB and greater
Software	Browser: Google Chrome Database: MYSQL Server: XAMPP server Programming Language: Hypertext Preprocessor (PHP), Cascading Stylesheet (CSS), and JavaScript

6.2.3 Test Schedule

The test schedule is used to confirm that the testing of ArithmeticNorm system has been finished on time which includes the testing tasks, and the target starts and end dates and should also describe each activity. This section clarifies the test schedule that has been made and recorded for the system.

ArithmeticNorm system has been tested for several days to confirm whether the user requirements are fulfilled or not. Table 6.2 shows the details of the testing schedule.

Table 6. 2: Test Schedule for ArithmeticNorm System

Activity	Descriptions	Duration	Start Date	End Date
Unit Testing	Used to test the functionalities of the system module.	8 days	1/8/2021	8/8/2021
Integration Testing	Use to test the module Integrated and verify functionality after integration.	5 days	9/8/2021	13/8/2021
System Testing	Used to evaluate whether the system compliance with its requirements or not.	7 days	14/8/2021	20/8/2021
Acceptance Testing	Used to test the complete system to end user that ensures whether the system meets all customer requirements or not.	3 days	21/8/2021	23/8/2021

6.3 Test Strategy

The test strategy is an outline that shows the testing approach of the system development. The goal of the test strategy is to make sure that the system achieves its objectives. To confirm that all objectives are properly addressed and understood by the end-user, the development and documentation of a test strategy should be done in a systematic manner. As the system improves and develops, it should also be updated frequently to test the new objectives.

Dynamic testing is used in ArithmeticNorm system which complex defects can be tested that may have been skipped in the review processes. Dynamic testing uses two techniques which are black box testing and white box testing. The black box testing method involves testing without regard for the system implementation or internal structure. In the white box testing method, testing will focus on the system implementation and internal structure. Table 6.3 shown the testing methods in detail. ArithmeticNorm system used the black box testing method because it is efficient, and the tester does not need to be familiar with programming languages or the system implementation. Thus, many moderately skilled testers can test the system.

Table 6. 3: Testing Approaches

Approaches	Explanations
Black Box Testing	Black box testing is a method of testing a system through the functional or non-functional based on the system requirements specifications. The tester does not have access to the source code and will often engage with the system user interface by providing inputs and evaluating outputs while executing a black box test without understanding how and where the inputs are processed.
White Box Testing	It is a method of software testing in which the tester must be familiar with the system's internal structure, code, and program to do white box testing on a system. After the detailed design document, the tester must examine the source code to determine which unit of code is acting abnormally.

6.3.1 Classes of Test

In this section, the classifications of test have been discussed which could be carried in the testing process. The classes of test consist of unit testing, integration testing, system testing, and acceptance testing.

v. Unit Testing

This system uses unit testing to find issues early in the development cycle, to ensuring that the module continues to function properly, as well as to validate the correctness and functionality of each module of the system. The practice is to create test cases for all functions and methods so that any changes in the system that cause a problem can be swiftly discovered and corrected.

vi. Integration Testing

Integration Testing is defined to verify the software modules work in unity and to verify all the interfaces which include Interface to the database, and software application. The goal of integration testing is to ensure that data is communicated between system modules.

vii. System Testing

This testing is to check for desired outputs by thoroughly checking each input in the program. In addition to evaluate how components interact with one another and with the system, the whole functional system was tested. This scenario is also known as End-to-End testing.

viii. User Acceptance Testing

The major goal of this testing is to ensure that the system meets the system requirements. The customer is the one who does this validation to satisfied with the system process.

6.4 Test Design

Test description and test data have been explained in detail in this section which is the process of creating test cases for testing the system.

6.4.1 Test Description

In this section, the test case identification, description of testing, testing type, and expected output have been described. The link and list of questions that were asked users who tested the system can be found in Appendix D.

Table 6.4 up to Table 6.8 describes the test case identification, description of testing, testing type, and expected output.

Table 6. 4: Login Module Testing

Test Case ID	Description	Testing Type	Expected Result
TC_01-1	Invalid username or password	Unit Testing	Display error message and the system will not proceed to next page.
TC_01-2	Username is blank or password is blank	Unit Testing	Display error message and the system will not proceed to the next page.
TC_01-3	Valid username and password	Unit Testing	Login successfully and proceed to next page.

Table 6. 5: Add Record Module

Test Case ID	Description	Testing Type	Expected Result
TC_02-1	All or some fields are blank.	Unit Testing	'The field is required' message will display.
TC_02-2	Invalid input for several fields.	Unit Testing	Invalid text will not appear on the screen while invalid media data type or size will not accept, and an error message will display.
TC_02-3	Valid input for all fields.	Unit Testing	'Record added successfully' message will display.

Table 6. 6: Add Media Data Module

Test Case ID	Description	Testing Type	Expected Result
TC_03-1	Fields are blank.	Unit Testing	'The field is required' message will display.
TC_03-2	Invalid media data.	Unit Testing	Invalid media data type or size will not accept, and an error message will display.
TC_03-3	Valid media data.	Unit Testing	'Record added successfully' message will display.

Table 6. 7: Score Student Marks Module

Test Case ID	Description	Testing Type	Expected Result
TC_04-1	All fields are blank.	Unit Testing	The system will not score marks.
TC_04-2	All or some fields have input.	Unit Testing	The system will display the score and the percentage of student answers as well as the correct answers.

Table 6. 8: Normalize Student Table Module

Test Case ID	Description	Testing Type	Expected Result
TC_05-1	There is partial dependency and transitive dependency.	Unit Testing	The system will proceed with 2NF process then proceed with 3NF process.
TC_05-2	There is partial dependency and there is no transitive dependency.	Unit Testing	The system will proceed with 2NF process then 'your table is in 3NF' message will display.
TC_05-3	There is no partial dependency and there is no transitive dependency.	Unit Testing	'Your table is in 2NF' message will display then 'your table is in 3NF' message will display.

6.4.2 Test Data

Test data of ArithmeticNorm system can be described into tables as shown in Table 6.9 up to Table 6.13.

Table 6. 9: Test Data for Login Module

Test Case ID / Column Name	TC_01-1	TC_01-2	TC_01-3
Username	Input black	Aisha1	Aisha111
Password	Input blank	abc	111111
Result Test Data	Login failed. "Please fill out this field" message will display.	Login failed. System will not proceed with next page because username or password do not match on the database.	System access successfully.

Table 6. 10: Test Data for Add Record Module

Test Case ID / Column Name	TC_02-1	TC_02-2	TC_02-3
First Name	Aisha	Aisha	Aisha
Last Name	Input blank	AL-bakri	AL-bakri
Username	Input black	Aisha111	Aisha123
Phone Number	Input blank	011 131118	011 131118
Email	Input blank	Ashe09877@gmail.com	Ashe09877@gmail.com
Password	Input blank	111	111111
Gender	Input blank	Female	Female
Date of Birth	Input blank	25-10-2009	25-10-2009
Country	Input blank	Yemen	Yemen
Result Test Data	Login failed. "Please fill out this field" message will display.	Login failed due to invalid inputs which are username is not available, password most be at least 6 characters.	System access successfully.

Table 6. 11: Test Data for Add Media Data Module

Test Case ID / Column Name	TC_03-1	TC_03-2	TC_03-3
Type	Input blank	3gp/ PNG/ Text	MP4/ JPG/ CSV
Size	Input blank	400 MB	200 MB
Result Test Data	Upload failed. "Please fill out this field" message will display.	Upload failed due to invalid inputs.	Media data upload successfully.

Table 6. 12: Test Data for Score Student Marks Module

Test Case ID / Column Name	TC_04-1	TC_04-2
Answer	Inputs blank	Input some or all answers.
Result Test Data	Score failed.	Score student marks successfully.

Table 6. 13: Test Data for Normalize Student Table Module

Test Case ID / Column Name	TC_05-1	TC_05-2	TC_05-3
Partial dependency	Borrow_ID -> Book_ID Student_ID -> Student_name	Borrow_ID -> Book_ID Student_ID -> Student_name	Student_ID -> Student_name Student_ID -> Student _DoB
Transitive dependency	Book_ID -> Book_name	There is no transitive dependency	There is no transitive dependency.
Result Test Data	Normalize into 2NF and 3NF successfully.	Normalize into 2NF successfully.	Normalize into normal form is failed due to there is no partial and transitive dependency.

6.5 Test Result and Analysis

This section will describe the test result and analysis in various test cases. Table 6.14 up to Table 6.18 show the test result of ArithmeticNorm system.

Table 6. 14: Test Result and Analysis for Login Module

Module: Login		Result	
Test Case ID	**Description**	**Success**	**Fail**
TC_01-1	Invalid username or password	✓	
TC_01-2	Username is blank or password is blank	✓	
TC_01-3	Valid username and password	✓	

Table 6. 15: Test Result and Analysis for Add Record Module

Module: Add Record		**Result**	
Test Case ID	**Description**	**Success**	**Fail**
TC_02-1	All or some fields are blank.	✓	
TC_02-2	Invalid input for several fields.	✓	
TC_02-3	Valid input for all fields.	✓	

Table 6. 16: Test Result and Analysis for Add Media Data Module

Module: Add Media Data		**Result**	
Test Case ID	**Description**	**Success**	**Fail**
TC_03-1	Fields are blank.	✓	
TC_03-2	Invalid media data.	✓	
TC_03-3	Valid media data.	✓	

Table 6. 17: Test Result and Analysis for Score Student Marks Module

Module: Score Student Marks		**Result**	
Test Case ID	**Description**	**Success**	**Fail**
TC_04-1	All fields are blank.	✓	
TC_04-2	All or some fields have input.	✓	

Table 6. 18: Test Result and Analysis for Normalize Student Table Module

Module: Normalize Student Table		**Result**	
Test Case ID	**Description**	**Success**	**Fail**
TC_05-1	There is partial dependency and transitive dependency.	✓	
TC_05-2	There is partial dependency and there is no transitive dependency.	✓	
TC_05-3	There is no partial dependency and there is no transitive dependency.	✓	

We received 32 responses from respected FTMK lectures and students at UTeM university who's chosen to do the testing phase. As show in Figure 6.1 up to Figure 6.4, there are 46% of the respondents are female while 53% of them are male. 62% with a range of age 22 – 24, and 21% with a range of age 19 – 21. 84% of the responses are from people with a bachelor's degree. From a total of 32 responses, there are 40% are learning or using a database for 3 to 6 years while 53% are learning or using a database for less than 3 years.

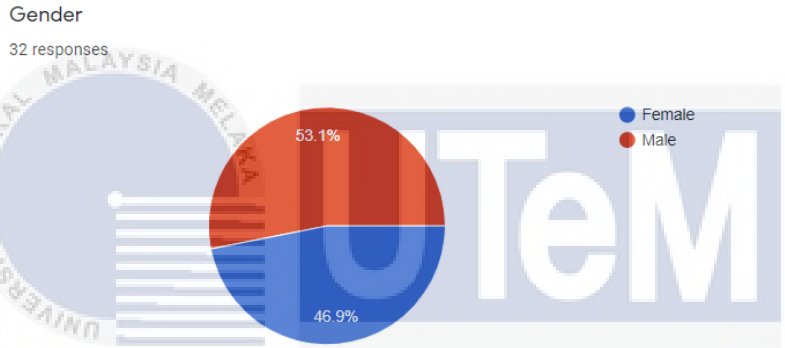


Figure 6. 1: Gender of the Respondents User's

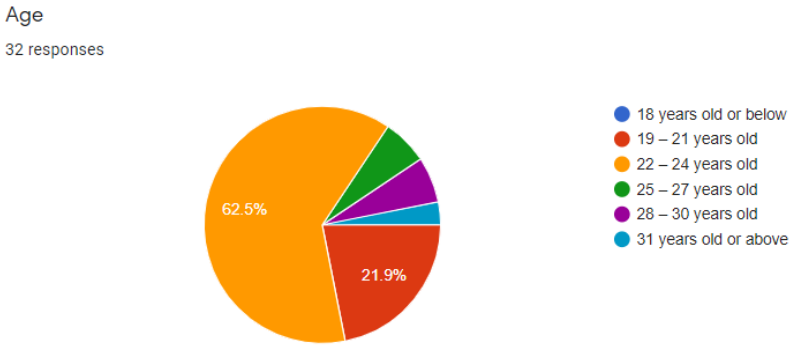


Figure 6. 2: Range Age of the Respondents User's

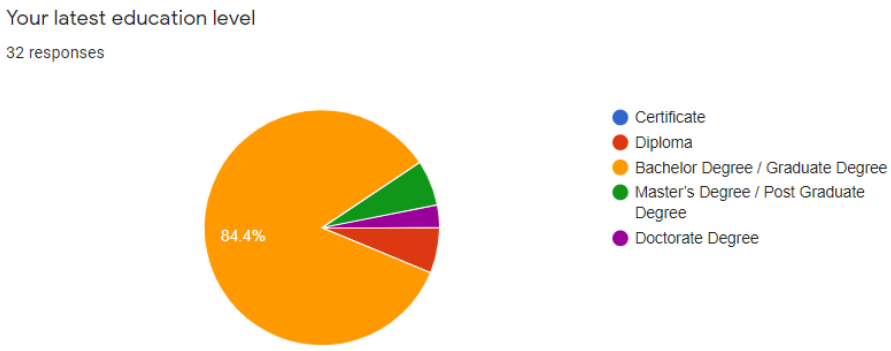


Figure 6. 3: Education Level of the Respondents User's

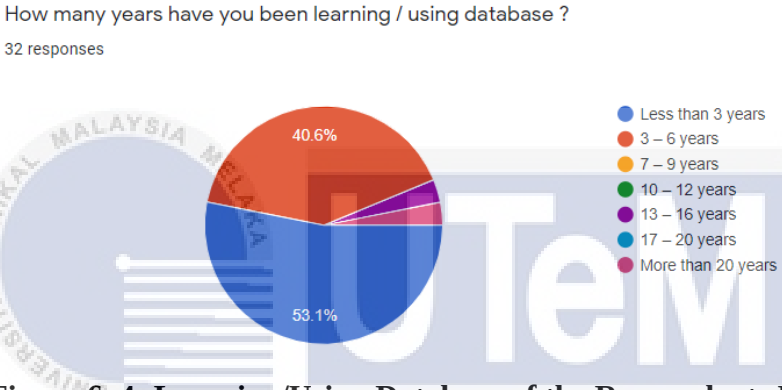


Figure 6. 4: Learning/Using Database of the Respondents User's

The results of the questionnaire have been recorded that used to evaluate the system. All the questionnaire results are gathered and analyzed. The average was evaluated to detect each question's average. Each question has a different level of satisfaction based on the question. Table 6.19 presents the results of agreements of the system usability and users satisfaction using weighted averages.

Table 6. 19: Weighted Averages of Usability and User Satisfaction

Survey Questions	Strongly disagree	Disagree	Neutral	Agree	Strongly agree	Weighted Averages
	1	2	3	4	5	
1. The system appears to me to be more complicated than it needs to be.	15	12	4		1	1.75
2. The system is simple and user friendly.	2		4	16	10	4.00
3. To use this system, I need support.	8	7	10	4	3	2.59
4. The system appears to be in good working order and is nicely integrated.	1	2	6	12	11	3.94
5. I believe that most people can easily pick up on this system.		2	6	13	11	4.03
6. This system takes a long time for me to figure out.	11	11	10			1.97
7. When I use this system, I feel confident.	1		9	12	10	3.94
8. Before I can start using this system, I need to learn a lot of things.	5	12	10	5		2.47
9. The website design was appealing and supportive of my learning abilities.			7	12	13	4.19
10. Each page's information was provided in a clear and understandable way.	2	1	6	12	11	3.91
11. I was satisfied with the overall content provided by this system.	1		3	18	10	4.13
12. After trying this system, I have a greater understanding of the subject and can put what I've learned into practice.			9	11	12	4.09
13. Educational systems, in my opinion, are an efficient technique to learn normalization.			1	11	20	4.59
14. I would return to this website.			9	11	12	4.09
15. The system performs just as I expected.		1	17	4	10	3.72
16. The system assists me in becoming more productive.		1	10	12	9	3.91

17. The system makes normalization topic more interesting.			3	7	22	4.59
18. I am satisfied with the system.			1	16	15	4.44
19. I would recommend the system to a friend.			4	14	14	4.31
20. It is enjoyable to work with.			4	16	12	4.25
21. I need to have it.			4	15	13	4.28
22. The system is user friendly.			4	14	14	4.31
23. To do what I want to do with the system, I need to take the fewest steps feasible.			7	12	13	4.19
24. I do not detect any inconsistencies when I use it.			5	15	12	4.22
25. It would appeal to both students and teachers.			2	9	21	4.59
26. I use it successfully every time.			7	11	14	4.22
27. This is an excellent system for students to use.				10	22	4.69

As a result, based on the weighted averages, the system meets all non-functional requirements that clearly written earlier in Section 2.4.2.

ArithmeticNorm system has become helpful for users and students. As shown in Figure 6.5, 93% of the respondents found that ArithmeticNorm system will increase their productivity while 84% of them confirm that ArithmeticNorm system will be useful in their studies.

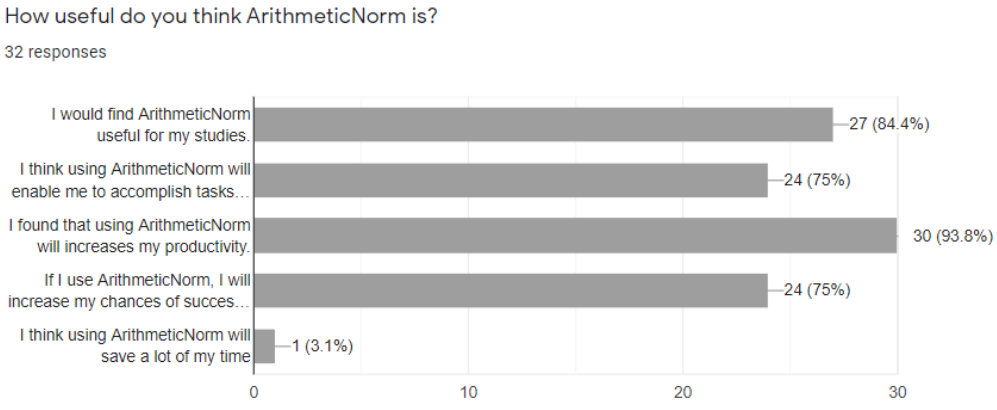


Figure 6. 5: Benefits of ArithmeticNorm system

6.6 Conclusion

In conclusion, this chapter has discussed the testing phase that has been done for testing ArithmeticNorm performance system. The method used to verify and validate the system is to make sure that the system achieves all its requirements and objectives as well as end-user requirements stated earlier. Testing phase mainly focuses on user satisfaction and must be planned wisely and thoroughly due to the costs of fixing the major defects can be very expensive.

The next chapter will explain the weakness and strengths of ArithmeticNorm system as well as discuss future improvements of the system.

CHAPTER VII

CONCLUSION

7.1 Introduction

ArithmeticNorm system is a web-based system that enhances teaching and learning database normalization. Besides that, ArithmeticNorm system will assist students in having a good database design.

This chapter will discuss the weaknesses and strengths of ArithmeticNorm system in addition to some recommendations that have been listed to overcome the issues faced by the end-user to make the system more flexible and beneficial to everyone.

7.2 Observations on Weaknesses and Strengths

Based on the testing results and tester feedbacks, it is found that the system has successfully achieved all the objectives and requirements, on the other hand, it is detected that the system has some weaknesses and strengths comparing with other researchers' tools that have been done in Chapter 2 Table 2.1.

Section 7.2.1 and 7.2.2 discuss the list of weaknesses and strengths of the system.

7.2.1 Weaknesses

i. The system does not provide forget password function to be easier for users to retrieve their passwords.
ii. The system does not provide the steps of normalizing user tables that give students a clear vision of how the process is done on their database.
iii. Normalize user table depend on the functional dependencies that will be input by the user. Thus, the normalization process may get the wrong way if the user has not input clear functional dependencies.

7.2.2 Strengths

i. The normalizing process can give students a clear visual aid using their database. Therefore, students get a clearer understanding by monitoring how their duplicate data can be eliminated.
ii. Students can check the normal forms of their database up to 3NF.
iii. The system can display all the keys of user tables after the normalizing process done.

7.3 Proposition of improvement

ArithmeticNorm system aims to provide a good environment for students to engage them in learning normalization in an effective and efficient way. Thus, to achieve that, based on the weaknesses stated in Section 7.2.1 there are some enhancements that have been considered for future improvements which are:

i. Provide step by step of each normal form process that can give students a better understanding of normalization process.

ii. Develop an algorithm that can check the functional dependencies that are input by the users based on the data. Thus, users make sure that they get the best database.

iii. Expand the process of normal forms up to 6NF.

iv. provide a function that helps users who forget their password to retrieve or change their password.

v. Develop a function that helps users to save and download their database after normalizing.

7.4 Contributions

Based on the normalization importance for computer science students that have been discussed in Chapter 1 Section 1.1, The contribution of this system is for all students who learning databases and who interesting in designing a good database especially for Universiti Teknikal Malaysia Melaka (UTeM) students and lecturers to make full use of this system. ArithmeticNorm system is not only for students but also for teachers and lecturers in the database field to make sure that their students obtain more knowledge about normalization topics.

7.4 Conclusion

In a conclusion, ArithmeticNorm system expects to help students and lecturers in the database normalization topics and provide an efficient environment that can be accessible and user-friendly for the end users. This system has successfully achieved the objectives, and functional and non-functional requirements that have been stated earlier in Chapter 1, and Chapter 3. However, this system still has some weaknesses that need to be improving in the future.

REFERENCES

[1] (DBLC) Database Life Cycle. (n.d.). Retrieved from https://www.relationaldbdesign.com/database-design/module3/dblc-design-stages.php

[2] 3 Reasons To Normalize Your Data. (2019, October 23).

[3] Amin, M., Romney, G., Dey, P., & Sinha, B. (2019, October). Teaching relational database normalization in an innovative way. *Journal of Computing Sciences in Colleges*, *35*, 48–56.

[4] Bahmani, A. H., Naghibzadeh, M., & Bahmani, B. (2008, May 4). Automatic database normalization and primary key generation. *Canadian Conference on Electrical and Computer Engineering*, p. 6. doi:10.1109/ccece.2008.4564486

[5] Brumm, B. (2017, Sep 30). Database Normalization.

[6] Chapple, M. (2020, April 12). Database Normalization Basics. Retrieved from https://www.lifewire.com/database-normalization-basics-1019735

[7] Codd. (1970).

[8] Cooper, T. (2021, Apr 2). Why is Data Modeling So Difficult?

[9] Data Anomalies. (n.d.). Retrieved from https://databasemanagement.fandom.com/wiki/Data_Anomalies

[10] Database Normalization. (n.d.). Retrieved from https://www.w3schools.in/dbms/database-normalization/

[11] Database Normalization. (n.d.). Retrieved from https://www.tutorialride.com/dbms/database-normalization.htm

[12] Database normalization. (2018, March). Retrieved from https://en.wikipedia.org/wiki/Database_normalization

[13] Diederich, J., & Milton, J. (1988, September). New methods and fast algorithms for database normalization. *ACM Transactions on Database Systems*, p. 27. doi:10.1145/44498.44499

[14] Georgiev, N. (2008, September 1). A Web-Based Environment for Learning Normalization of Relational Database Schemata. *Umea University Department of Computing Science*, p. 63. doi:10.1.1.160.4514

[15] Introduction to Data Normalization. (n.d.).

[16] Ketkar, P., Barbadekar, B. V., & Dhabe, P. S. (2011, Mar). Possible Algorithms of 2NF and 3NF for DBNorma- A tool for Relational Database Normalization. *Dnyanganga College of Engineering and research*, p. 7. doi:01.IJIT01.01.86

[17] Li, L. (2019). Database Normalization Explained.

[18] Lucchesi, C. L. (1978, February 21). Candidate Keys for Relations. *Journal of Computer and System Sciences*, p. 10. doi:10.1016/0022-0000(78)90009-0

[19] Mendjoge, N. J. (2016). Intelligent tutoring system for Database Normalization.

[20] Mendjoge, N., R. Joshi , A., & Narvekar, M. (2016, Aug 13). *Intelligent Tutoring System for Database Normalization.* Mumbai, J. Sanghvi College of Engineering. Pune, India: IEEE. doi:10.1109/ICCUBEA.2016.7860013

[21] MUÑOZ, A. (2019, NOVEMBER 11). Why is database normalization so important? Retrieved from https://blog.saleslayer.com/why-is-database-normalization-so-important

[22] Narvekar, M., Mendjoge, N., & Joshi, A. (2016). Intelligent tutoring system for Database Normalization.

[23] Normalization of Database. (n.d.). Retrieved from https://www.studytonight.com/dbms/database-normalization.php

[24] Rapid Application Development (RAD). (2021, March 31). Retrieved from https://kissflow.com/low-code/rad/rapid-application-development/

[25] SHANARDI, A. (2018). DRAWBACKS OF NORMALIZATION.

[26] Stellnberger, E., Pfaff, E., Diaz-Sanchez, D., & Ward-Caviness, C. (2018, 10 19). FDTool. *a Python application to mine for functional dependencies and candidate keys in tabular data*, p. 16. Retrieved from https://f1000research.com/articles/7-1667

[27] Sunitha, G., & Jaya, A. (2013, May). A knowledge based approach for automatic database normalization. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 2*, p. 4. Retrieved from http://ijarcet.org/wp-content/uploads/2013/06/1816-1819.pdf

[28] Testing Phase in SDLC. (n.d.). Retrieved from https://study.com/academy/lesson/testing-phase-in-sdlc.html

[29] Three Information Systems Development Methods Information Technology Essay. (2015, Jan 1). p. 7. Retrieved from https://www.ukessays.com/essays/information-technology/three-information-systems-development-methods-information-technology-essay.php

[30] Wang, J., Stantic , B., & Ren, X. (2015). *Normalzation Tool*. (N. Tool, Producer, & Griffith University) Retrieved from ict.griffith.edu.au: http://www.ict.griffith.edu.au/normalization_tools/normalization/ind.php

[31] Wang, T. (., Du, H., & Lehmann, C. M. (2010). Accounting For The Benefits Of Database Normalization.

[32] Welzer, T., Rozman, I., & G, J. (1989, January). Automated normalization tool. *University of haribor, faculty of technical sciences, Smetanova 17*, p. 6. Retrieved from https://doi.org/10.1016/0165-6074(89)90224-X

[33] What is Data Normalization and Why Is It Important. (2019, May 7). Retrieved from https://www.import.io/post/what-is-data-normalization-and-why-is-it-important/

[34] What Is Performance Tuning? Why Should You Do It? (n.d.). Retrieved from https://technologypartners.net/blog/2016/06/performance-tuning-why-do-it-st-louis-information-technology-recruiters/

[35] Winters, K. (2021). The Importance of Data Normalization. Retrieved from https://tipsmake.com/the-importance-of-data-normalization

[36] Yazici, A., & Karakaya, Z. (2007). *JMathNorm A Database Normalization Tool Using Mathematica.* Atilim University, TOBB University of Economics & Technology, Computer Engineering, Ankara - Turkey. Retrieved from https://link.springer.com/content/pdf/10.1007%2F978-3-540-72586-2_27.pdf

[37] Yee, S. (2016, Apr 02). Importance of Normalization. Retrieved from https://www.slideshare.net/Uniqueangel1/importance-of-normalization

APPENDIX A: Normal Form Algorithm

i. Check and find 2NF pseudo-code

```
Start
Open file
Fetch data into allRows array

FDleftArray = Array;
FDrightArray = Array;
aleft = Array;
aright = Array;
newtable = Array;
newtable2 = Array;
allleft = Array;


Foreach ( FDleft as a => b)
        push FDleft [a] into FDleftArray,
        push FDright [a] into FDrightArray;
End foreach

countFD = count FDleftArray;
FD = Array;
AllFD = Array;

for (c =0; c <countFD; c++)
      push FDleftArray [c] into FD;
      push FDrightArray [c] into FD;
      push FD into AllFD;
      FD = Array;
End for

for (c =0; c <countFD; c++)
      If  c ∉ allleft  Then
            for (a =0; a <countFD; a++)
              If FDleftArray [c] == FDleftArray [a] Then
```

```
                    if (FDleftArray [c] ∉ newtable Then
                        push FDleftArray [c] into newtable;
                    End If
                    push c into aleft;
                    push a into aleft;
                    push c into allleft;
                    push a into allleft;
                End If
            End for

            aleftuniq = unique of aleft;
            countnewtable = count unique of aleft;

            for (i =0; i <countFD; i++)
                if i ∈ aleftuniq Then
                    push FDrightArray [i] into newtable;

                End If
                else
                    continue;

            End for
            push newtable into newtable2;
            aleft = Array;
            newtable = Array;
        End If
End for

temallcolumn = Array;
for (c =0; c < count(newtable2); c++)
    for (i =0; i < count(newtable2 [c]); i++)
        if newtable2 [c][i] ∉ temallcolumn Then
            push newtable2 [c][i] into temallcolumn;
        End If
    End for
End for

missedcolumn = Array;
for (c =0; c < count(allRows); c++)
```

```
        if allRows [c] ∉ temallcolumn Then
                push allRows [c] into missedcolumn;
        End If
End for

If count primary > 1 Then
    If missedcolumn not empty Then
            p = Array;
            for (c =0; c <count(primary); c++)
                push primary[c] into p;
            End for

            for (c =0; c <count(missedcolumn); c++){
                push missedcolumn[c] into p;
            End for
            push p into newtable2;
    End If
End If

newtable2NFtem = newtable2;
FK = Array;
FK2NF = Array;
tr1;
tr2;

for (i =0; i < count(newtable2NFtem); i++)
    push newtable2NFtem [i][0] into FK;
End for

for (c =0; c < count(FK); c++)
        if FK[c] ∉ FDleftArray Then
            unset FK[c];
        End If
End for

FK2NF = FK;
PKFK = Array;

for (i =0; i < countFD; i++)
```

```
        if FDrightArray[i] ∈  FK Then
            tr1 = FDleftArray[i];
            for (c =0; c < count(newtable2NFtem); c++)
                if newtable2NFtem[c][0] == FDrightArray[i] Then
                        tr2 = newtable2NFtem[c][1];
                        unset newtable2NFtem[c];
                        for (c =0; c < count(FK2NF); c++)
                                if FK2NF[c] == FDrightArray[i] Then
                                        if FK2NF[c] ∈ primary Then
                                                if PKFK ∉ FK2NF[c] Then
                                                        push FK2NF[c] into PKFK;
                                                End If
                                        End If
                                        unset FK2NF[c];

                                End If
                        End for

                        for (c =0; c < count(newtable2NFtem); c++)
                                if newtable2NFtem[c][0] == tr1Then
                                        if tr2 ∉ newtable2NFtem[c] Then
                                                push tr2 into newtable2NFtem[c];
                                        End If
                                End If
                        End for
                End If

                End for
    End If
End for

for (c =0; c < count(FK2NF); c++)
  if FK2NF[c] ∈ primary Then
        if FK2NF[c] ∉ PKFK Then
                push FK2NF[c] into PKFK;
        End If
        unset FK2NF[c];
  End If
End for
```

```
for (c =0; c < count(FK2NF); c++)
        if FK2NF[c] ∈ primary Then
              if FK2NF[c] ∉ PKFK Then
                  push FK2NF[c] into PKFK;
         End If
              unset FK2NF[c];
        End If

 End for


for (c =0; c < count(newtable2NFtem); c++)
     if newtable2NFtem[c][0] ∈ primary Then
       for (i =0; i < count(FK2NF); i++)
             push FK2NF[i] into newtable2NFtem[c]
        if missedcolumn not empty Then
              if count primary == 1 Then
                    for (m =0; m <count(missedcolumn); m++)
                          push missedcolumn[m] into newtable2NFtem[c]
                    End for
              End if
        End if
        End for
     End if
     else
       if count primary == 1 Then
             if missedcolumn not empty Then
               for (c =0; c < count(newtable2NFtem); c++)
                     tem = Array;
                     for (p =0; p <count(primary); p++)
                           push primary[p] into tem;
                     End for

                     for (m =0; m <count(missedcolumn); m++)
                           if missedcolumn[m] ∉ tem Then

                                 push missedcolumn[m] into tem;
                           End If
                     End for
```

```
                    for (i =0; i < count(FK2NF); i++){
                        push FK2NF[i] into tem;
                    End for
                End for
                Push tem into newtable2NFtem;

            End If
        End If

    End else
End for

for (i =0; i < count(newtable); i++)
  for (c =0; c < count(allRows[0]); c++)
      if allRows[0][c] ∈ newtable[i] Then
          push c into n;
      End If
  End for

  alldata = Array;
  for (c =0; c < rowcount +1; c++)
      for (a =0; a < num; a++)
          if a ∈ n Then
              push allRows[c][a] into data;
          End If
          else
              continue;
          End else
      End for
      If data ∉ alldata
          push data into alldata;
      End If
      data = Array;
  End for

  If count newtable2NFtem <= 1 Then

      for (i =0; i < count(newtable2NFtem); i++)
          for (c =0; c < count(allRows[0]); c++)
              if allRows[0][c] ∈ newtable2NFtem [i] then
```

```
                        push c into n;
                End if
        End for
        alldata = Array;
        for (c =0; c < rowcount; c++){
                for (a =0; a < num; a++){
                        if a ∈ n then
                                push allRows[c][a] into data;
                        End if
                        Else
                                continue;
                        End else
                End for
                If data ∉ alldata then
                        push data into alldata;
                End if
                data = Array;
        End for

        for (c =0; c < count(alldata); c++)
                for (a=0; a < count(n); a++)
                        Print alldata[c][a];
                End for
        End for
    End for
End If

Else
    Print “table is already in 2NF”;
End else

Close file
End
```


ii. Check and find 3NF pseudo-code

```
Start
```

```
Open file
Fetch data into allRows array

FDleftArray = Array;
FDrightArray = Array;
aleft = Array;
aright = Array;
newtable = Array;
newtable2 = Array;
allleft = Array;


Foreach( FDleft as a => b)
        push FDleft [a] into FDleftArray,
        push FDright [a] into FDrightArray;
End foreach

countFD = count FDleftArray;
FD = Array;
AllFD = Array;

for (c =0; c <countFD; c++)
    push FDleftArray [c] into FD;
    push FDrightArray [c] into FD;
    push FD into AllFD;
    FD = Array;
End for

for (c =0; c <countFD; c++)
    If  c ∉ allleft  Then
            for (a =0; a <countFD; a++)
              If FDleftArray [c] == FDleftArray [a] Then
                    if (FDleftArray [c] ∉ newtable Then
                            push FDleftArray [c] into newtable;
                    End If
                    push c into aleft;
                    push a into aleft;
                    push c into allleft;
                    push a into allleft;
```

```
            End If
        End for

        aleftuniq = unique of aleft;
        countnewtable = count unique of aleft;

        for (i =0; i <countFD; i++)
            if i ∈ aleftuniq Then
                push FDrightArray [i] into newtable;

            End If
            else
                continue;

        End for
        push newtable into newtable2;
        aleft = Array;
        newtable = Array;
    End If
End for

temallcolumn = Array;
for (c =0; c < count(newtable2); c++)
    for (i =0; i < count(newtable2 [c]); i++)
        if newtable2 [c][i] ∉ temallcolumn Then
            push newtable2 [c][i] into temallcolumn;
        End If
    End for
End for

missedcolumn = Array;
for (c =0; c < count(allRows); c++)
    if allRows [c] ∉ temallcolumn Then
        push allRows [c] into missedcolumn;
    End If
End for

If count primary > 1 Then
    If missedcolumn not empty Then
```

```
                p = Array;
                for (c =0; c <count(primary); c++)
                        push primary[c] into p;
                End for

                for (c =0; c <count(missedcolumn); c++){
                        push missedcolumn[c] into p;
                End for
                push p into newtable2;
        End If
End If

newtable2NFtem = newtable2;
FK = Array;
FK2NF = Array;
tr1;
tr2;

for (i =0; i < count(newtable2NFtem); i++)
        push newtable2NFtem [i][0] into FK;
End for

for (c =0; c < count(FK); c++)
        if FK[c] ∉ FDleftArray Then
                unset FK[c];
          End If
End for

FK2NF = FK;
PKFK = Array;

for (i =0; i < countFD; i++)
        if FDrightArray[i] ∈  FK Then
              tr1 = FDleftArray[i];
              for (c =0; c < count(newtable2NFtem); c++)
                  if newtable2NFtem[c][0] == FDrightArray[i] Then
                              tr2 = newtable2NFtem[c][1];
                              unset newtable2NFtem[c];
                              for (c =0; c < count(FK2NF); c++)
```

```
                    if FK2NF[c] == FDrightArray[i] Then
                        if FK2NF[c] ∈ primary Then
                            if PKFK ∉ FK2NF[c] Then
                                push FK2NF[c] into PKFK;
                            End If
                        End If
                        unset FK2NF[c];

                    End If
                End for

                for (c =0; c < count(newtable2NFtem); c++)
                    if newtable2NFtem[c][0] == tr1Then
                        if tr2 ∉ newtable2NFtem[c] Then
                            push tr2 into newtable2NFtem[c];
                        End If
                    End If
                End for
            End If

        End for
    End If
End for
```



```
for (i =0; i < countFD; i++)
  if FDrightArray[i] ∈ FK then
        tr1 = FDrightArray [i];
        trtem = FDleftArray [i];

        for (x =0; x < countFD; x++)
            if FDleftArray[x] == tr1 then
                targettr =  FDrightArray[x];
            End if
        End for

        for (x =0; x < count(newtable2); x++)
            if newtable2 [x][0] == trtem then
                for (t =0; t < count(newtable2[x]); t++)
                    if newtable2 [x][t] == targettr then
```

```
                                        Delete newtable2[x][t];
                                    End if
                                End for
                            End if
                        End for
                    End if
                End for


If count primary == 1 then
    for (c =0; c < count(FK2NF); c++)
        if FK2NF[c] ∈ primary then
            if FK2NF[c] ∉ PKFK then
                push FK2NF[c] into PKFK;
            End if
            Delete FK2NF[c];
        End if
    End for
End if


    for (c =0; c < count(FK2NF); c++)
        if FK2NF[c] ∈ primary then
            if FK2NF[c] ∉ PKFK then
                push FK2NF[c] into PKFK;
            End if
            Delete FK2NF[c];
        End if
    End for

    for (c =0; c < count(FK); c++)
        if FK[c] ∈ primary then
            if FK[c] ∉ PKFK3NF then
                push FK[c] into PKFK3NF;
            End if
            Delete FK[c];
        End if
    End for
```

```
for (c =0; c < count(newtable2NFtem); c++)
        if count primary > 1 then
           if primary[0] ∈ newtable2NFtem[c] & primary[1] ∈ newtable2NFtem[c] then
                  for (i =0; i < count(FK2NF); i++)
                        if FK[i] ∉ newtable2NFtem[c] then
                              push FK2NF[i] into newtable2NFtem[c];
                        End if
                        If missedcolumn not empty then
                              If count primary == 1 then
                                    for (m =0; m <count(missedcolumn); m++)
                                       If newtable2NFtem[c] ∉ primary then
                                             Push missedcolumn[m] into
                                    newtable2NFtem[c];
                                       End if
                                    End for
                              End if
                        End if
                  End for
           End if
        End if
        else
                  if count primary == 1 then
                        if missedcolumn not empty then
                              for (c =0; c < count(newtable2NFtem); c++)
                                    tem = Array;
                                    for (p =0; p <count(primary); p++)
                                          push primary[p] into tem;
                                    End for

                                    for (m =0; m <count(missedcolumn); m++)
                                          if missedcolumn[m] ∉ tem then
                                                push missedcolumn[m] into tem;
                                          End if
                                    End for
                                    for (i =0; i < count(FK2NF); i++)
                                          push FK2NF[i] into tem;
                                    End for
                              End for
                              Push tem into newtable2NFtem;
```

```
                    End if
                End if
            End else
End for
    transitive = Array;
    transitive_table = Array;

    for (t =0; t < countFD; t++)
        if FDrightArray[t] ∈ FK then
            tr = FDleftArray[t];
            if tr ∈ FK or tr ∈ primary then
                push FDrightArray[t] into transitive;
            End if
        End if
    End for

    If count(primary) > 1 then
        for (t =0; t < countFD; t++)
            if FDrightArray[t] ∈ FK then
                tr = FDleftArray[t];
                if tr ∉ transitive_table then
                    push tr into transitive_table;
                End if
            End if
        End for
    End if

    ELSE
        for (t =0; t < countFD; t++)
            if FDrightArray[t] ∈ FK then
                tr = FDleftArray[t];
                if tr ∉ primary then
                    if tr ∉ transitive_table then
                        push tr into transitive_table;
                    End if
                End if
            End if
        End for
    End else
```

```
for (c =0; c < count(newtable2); c++){
    if newtable2[c][0] ∈ primary then
        for (i =0; i < count(FK); i++)
            if count primary > 1 then
                for (x =0; x < count(newtable2); x++){
                  if primary[0] ∈ newtable2[x] & primary[1] ∈ newtable2[x]
                then
                    if FK[i] ∉ newtable2[x]) & FK[i] ∉ transitive then
                        push FK[i] into newtable2[c];
                    End if
                  End if
                End for
            End if

            if missedcolumn not empty then
                if count primary == 1 then
                    for (m =0; m <count(missedcolumn); m++)
                        if newtable2[c] ∉ primary then
                            push missedcolumn[m] into newtable2[c];
                        End if
                    End for
                End if
            End if
        End for
    End if
    else
        if count primary == 1 then
            if missedcolumn not empty then
                for (c =0; c < count(newtable2); c++)
                    tem = Array;
                    for (p =0; p <count(primary); p++)
                        push primary[p] into tem;
                    End for

                    for (m =0; m <count(missedcolumn); m++)
                        if missedcolumn[m] ∉ tem then
                            if missedcolumn[m] ∉ FK then
```

```
                                        push missedcolumn[m] into tem;
                                    End if
                                End if
                            End for

                            for (i =0; i < count(FK); i++)
                                for (t =0; t < countFD; t++)
                                    if FDrightArray[t] ∈ FK then
                                        tr = FDleftArray[t];
                                        if tr ∉ primary then
                                            push tr into transitive_table;
                                        End if

                                        If FK[i] ∉ transitive then
                                            push FK[i] into tem;
                                        End if
                                    End if
                                End for
                            End for
                        End for

                        push tem into newtable2;
                    End if
                End if
            End else
        End for

        If count newtable == count newtable2NFtem then
            print “table already in 3NF”;
        End if

        Else
            for (i =0; i < count(newtable2); i++)
                for (c =0; c < count(allRows[0]); c++)
                    if allRows[0][c] ∈ newtable2[i] then
                        push c into n;
                    End if
                End for
                alldata = Array;
```

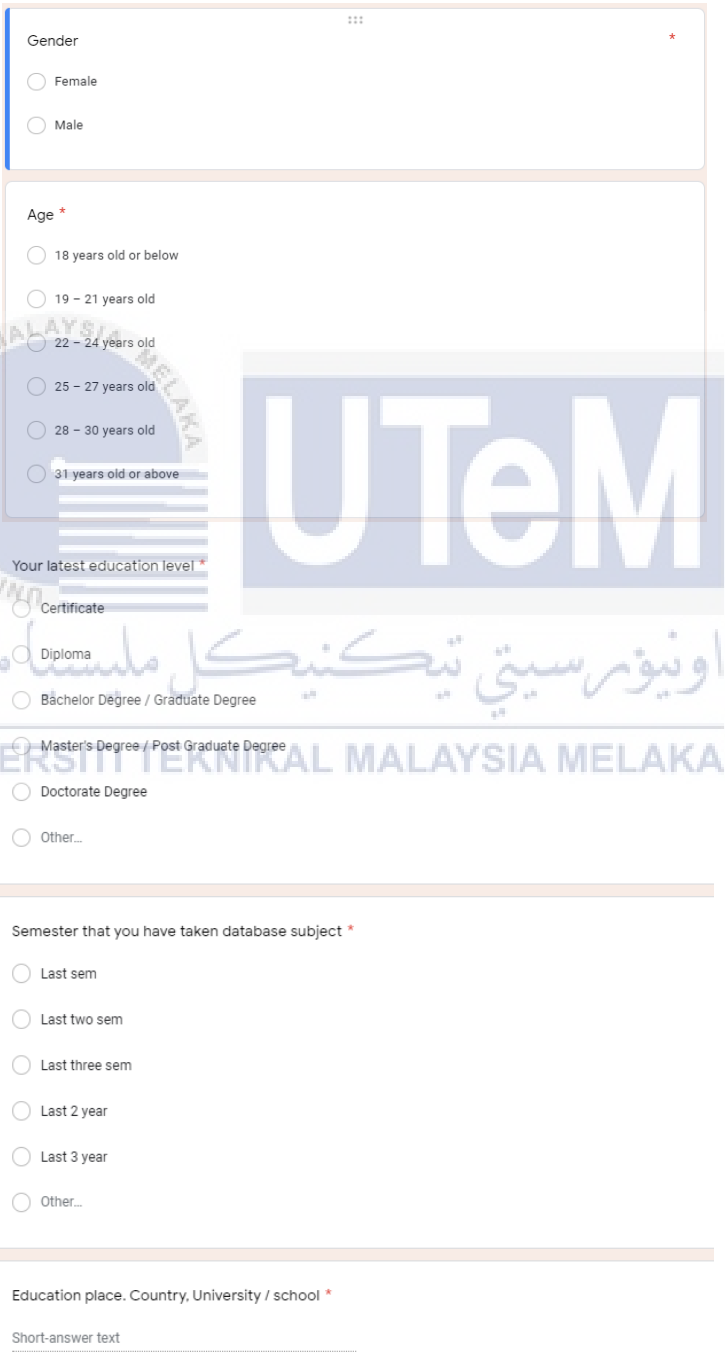
```
for (c =0; c < rowcount; c++){
        for (a =0; a < num; a++){
                if a ∈ n then
                        push allRows[c][a] into data;
                End if
                Else
                        continue;
                End else
        End for
        If data ∉ alldata then
                push data into alldata;
        End if
        data = Array;
End for

for (c =0; c < count(alldata); c++)
        for (a=0; a < count(n); a++)
                Print alldata[c][a];
        End for
End for
End for
End else
Close file
End
```

APPENDIX B: Questions of Students' Questionnaire

The link of students' questionnaire can be found in

https://docs.google.com/forms/d/1Jz5fWkUclcFgR7qvRgYFXTZawyDpIER3u8ojot8p4to/edit



Database Subjects - Scale of Happiness

Please rate your experience dealing with this subject. Following the Likert scale, the numbering below represent your scale of experience when learning the listed topics.
1 - Depressed.
2 - Sad.
3 - Content.
4 - Happy.
5 - Excited.
6 - Not applicable (Not being taught).

Through your experience, please rate the below topics according to the scale of happiness experience. *

	1	2	3	4	5	6
Database Int...	○	○	○	○	○	○
Database Lif...	○	○	○	○	○	○
Entity Relatio...	○	○	○	○	○	○
Normalization	○	○	○	○	○	○
Data Definiti...	○	○	○	○	○	○
Data Manipu...	○	○	○	○	○	○
RA and SQL -...	○	○	○	○	○	○
RA and SQL -...	○	○	○	○	○	○
RA and SQL -...	○	○	○	○	○	○
RA and SQL -...	○	○	○	○	○	○
RA and SQL -...	○	○	○	○	○	○
RA and SQL -...	○	○	○	○	○	○

Normalization is: *

- ○ Putting fields from different tables into one big database.
- ○ Organizing a database to remove repeated entries and increase the accuracy of the data.
- ○ Removing all necessary data from a database.
- ○ All of the above.

The following figure shows example of: *

Students

IDSt	LastName	IDProf	Prof	Grade
1	Mueller	3	Schmid	5
2	Meier	2	Borner	4
3	Tobler	1	Bernasconi	6

Startsituation

Result after normalisation

Students

ID	LastName
1	Mueller
2	Meier
3	Tobler

Professors

IDProf	Professor
1	Bernasconi
2	Borner
3	Schmid

Grades

IDStIDProf	Grade	
1	3	5
2	2	4
3	1	6

- 1NF
- 2NF
- 3NF

The following figure shows example of: *

Students

FirstName	LastName	Knowledge
Thomas	Mueller	Java, C++, PHP
Ursula	Meier	PHP, Java
Igor	Mueller	C++, Java

Startsituation

Result after Normalisation

Students

FirstName	LastName	Knowledge
Thomas	Mueller	C++
Thomas	Mueller	PHP
Thomas	Mueller	Java
Ursula	Meier	Java
Ursula	Meier	PHP
Igor	Mueller	Java
Igor	Mueller	C++

- 1NF
- 2NF
- 3NF

Is normalization difficult topic? *

- Yes
- No

Kindly justify your answer *

Long-answer text

In your opinion, what could be the challenges faced by the students in understanding and implementing database normalization? *

- [ ] Dry subject
- [ ] Theoretical subject
- [ ] Tricky
- [ ] Difficult to remember
- [ ] Difficult to understand
- [ ] Complex
- [ ] Do not like to observe pattern of data
- [ ] Did not refer to a proper data
- [ ] Other...

In your opinion, what motivates the students to learn database normalization? *

- [ ] Free in asking a lot of questions to the lecturer
- [ ] Completing various assignments
- [ ] Extracurricular activities related to the topics
- [ ] Lecturer gives relevant examples related to the topics
- [ ] Lecturer gives support in teaching and learning
- [ ] Lecturer is friendly and can ask anything
- [ ] Lots of videos related to topics been given
- [ ] Other...

Among the topics given below, which is the easiest topic? *

- Database Life Cycle
- Entity Relationship Diagram (ERD)
- Normalization
- Data Definition Language
- Data Manipulation Language
- RA and SQL - single query
- RA and SQL - set operation
- RA and SQL - aggregate function
- RA and SQL - inner Join
- RA and SQL - outer Join
- RA and SQL - subquery

Among the topics given below, which is the most difficult topic? *

- Database Life Cycle
- Entity Relationship Diagram (ERD)
- Normalization
- Data Definition Language
- Data Manipulation Language
- RA and SQL - single query
- RA and SQL - set operation
- RA and SQL - aggregate function
- RA and SQL - inner Join
- RA and SQL - outer Join
- RA and SQL - subquery

Do you think normalization is a tough topic? *

- Strongly disagree
- Disagree
- Neutral
- Agree
- Strongly agree

Kindly justify your answer above *

Short-answer text

Would you like to try a tool that may help you in checking if your database is normalize or not? *

- Yes, of course!
- No, thanks.
- Yes, if i'm avaiable.

Do you agree that a normalization tool will help students in learning normalization concepts? *

- Yes
- Maybe
- No

Kindly justify your answer above

Long-answer text

Would you like to test our proposed tool? *

- Yes
- No

APPENDIX C: Questions of Teachers' Questionnaire

The link of teachers' questionnaire can be found in

https://docs.google.com/forms/d/1tdNmjnTCH_TGMbwrY7a1A6bl0GsNB8Elg6XKNHb8TTU/edit

Gender

- Female
- Male

Age *

- 28 – 31 years old or below
- 32 – 35 years old
- 36 – 39 years old
- 40 – 43 years old
- 44 years old or above

Education Level *

- Certificate
- Diploma
- Bachelor Degree / Graduate Degree
- Master's Degree / Post Graduate Degree
- Doctorate Degree
- Other...

How many years have you been teaching database subject? *

- Less than 3 years
- 3 – 6 years
- 7 – 9 years
- 10 – 12 years
- 13 – 16 years
- 17 – 20 years
- More than 20 years

Where have you been teaching? or where did you teach? *

Long-answer text

Is normalization a database topic that you teach? *

- Yes
- No

Do you think that students have difficulties in learning normalization topics? *

- Yes
- No

Kindly justify your reason. *

Long-answer text

Database Subjects - Scale of Experience

Please rate your experience teaching this subject. Following the Likert scale, the numbering below represents your scale of experience when teaching the listed topics.
1 - Very difficult.
2 - Difficult.
3 - Neutral.
4 - Easy.
5 - Very easy.
6 - Not applicable (Never teach).

List of Database Topics *

	1	2	3	4	5	6
Database Int...	○	○	○	○	○	○
Database Lif...	○	○	○	○	○	○
Entity Relatio...	○	○	○	○	○	○
Normalization	○	○	○	○	○	○
Data Definiti...	○	○	○	○	○	○
Data Manipu...	○	○	○	○	○	○
RA and SQL -...	○	○	○	○	○	○
RA and SQL -...	○	○	○	○	○	○
RA and SQL -...	○	○	○	○	○	○
RA and SQL -...	○	○	○	○	○	○
RA and SQL -...	○	○	○	○	○	○
RA and SQL -...	○	○	○	○	○	○

Among the topics given below, which is the easiest topic? *

- Database Life Cycle
- Entity Relationship Diagram (ERD)
- Normalization
- Data Definition Language
- Data Manipulation Language
- RA and SQL - single query
- RA and SQL - set operation
- RA and SQL - aggregate function
- RA and SQL - inner Join
- RA and SQL - outer Join
- RA and SQL - subquery

Among the topics given below, which is the most difficult topic? *

- Database Life Cycle
- Entity Relationship Diagram (ERD)
- Normalization
- Data Definition Language
- Data Manipulation Language
- RA and SQL - single query
- RA and SQL - set operation
- RA and SQL - aggregate function
- RA and SQL - inner Join
- RA and SQL - outer Join
- RA and SQL - subquery

Kindly justify your answer above *

Long-answer text

Do you have experience using any normalization tool? *

- Yes
- No

If yes, please name the tool or perhaps a link that of that tool

Long-answer text

Do you agree that the normalization tool will reduce the time needed to teach normalization concepts? *

- Yes
- Maybe
- No

Kindly justify your answer above *

Long-answer text

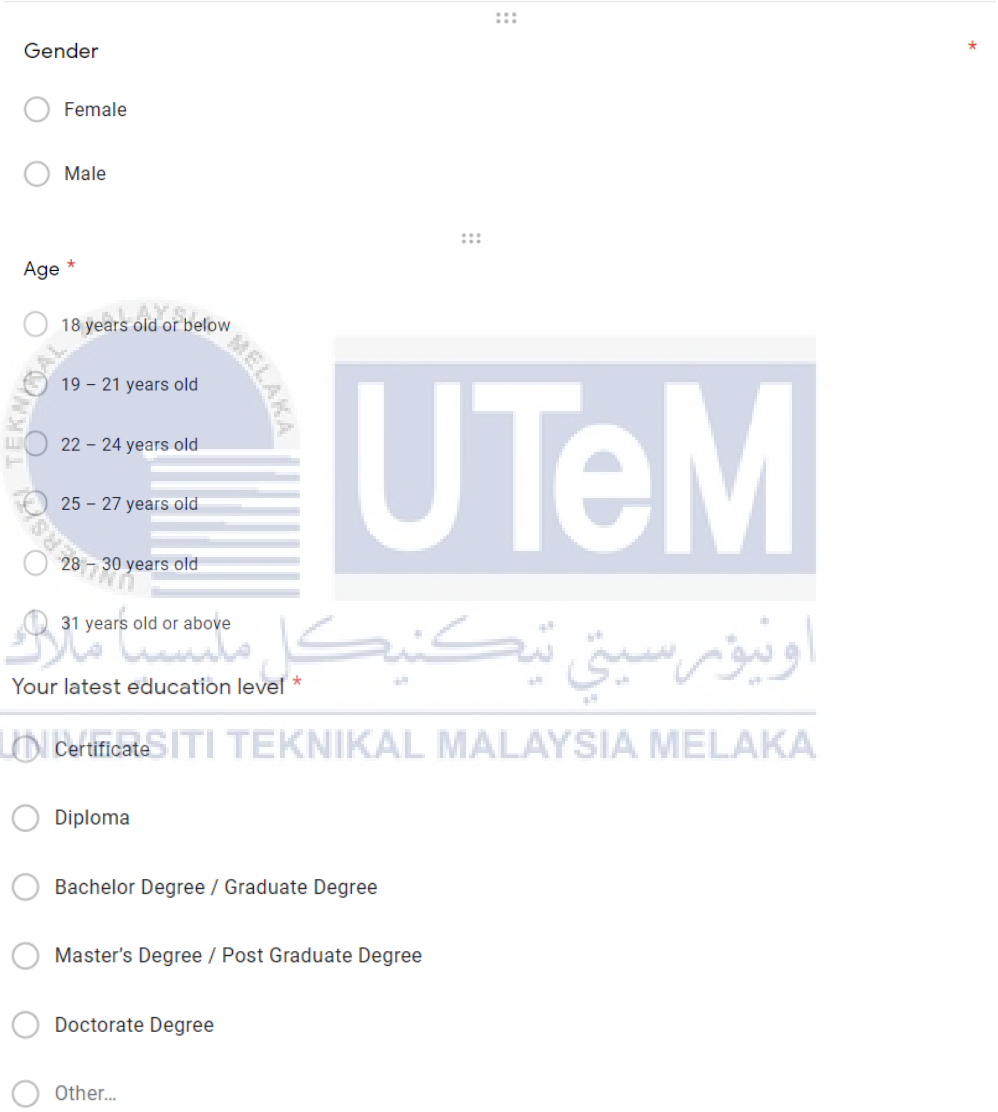
Would you like to test our proposed tool? *

- Yes
- No

APPENDIX D: Questions of Users' Questionnaire

The link of users' questionnaire can be found in

https://docs.google.com/forms/d/1239IHVxSOGp7ktJ4dpPDstPNXDtnQSGyS50t3AzbFx4/edit



How many years have you been learning / using database ?

- ○ Less than 3 years
- ○ 3 – 6 years
- ○ 7 – 9 years
- ○ 10 – 12 years
- ○ 13 – 16 years
- ○ 17 – 20 years
- ○ More than 20 years

ArithmeticNorm system - Scale of agreements

Please rate your experience dealing with the system. Following the Likert scale, the numbering below represent your scale of experience with the system.
1 - Strongly disagree
2 - Disagree
3 - Neutral
4 - Agree
5 - Strongly agree

Please state your level of agreement or disagreement for the following statements based on your experience with the system. *

	1	2	3	4	5
I find this system to be more complicated than it should be	○	○	○	○	○
I think the system is simple and easy to use.	○	○	○	○	○
I need technical support to use this system.	○	○	○	○	○
I find the system functioning smoothly and is well integrated	○	○	○	○	○
I think most people can learn this system quickly.	○	○	○	○	○
I find this system to be time-consuming.	○	○	○	○	○
I feel confident while using this system.	○	○	○	○	○

I think there are a lot of things to learn before I can start using this system.	○	○	○	○	○
The website design was pleasing and supported my ability to learn.	○	○	○	○	○
Information on each page was written in a way that was easy to understand.	○	○	○	○	○
I was satisfied with the overall content contained within this system.	○	○	○	○	○
After reviewing this system, I better understand the subject, and can apply what I have learned.	○	○	○	○	○
I personally find educational system are an effective way to learn normalization.	○	○	○	○	○
I would visit this website again.	○	○	○	○	○
The system does everything I would expect it to do.	○	○	○	○	○
The system helps me be more effective.	○	○	○	○	○
The system makes normalization more interesting.	○	○	○	○	○
I am satisfied with it.	○	○	○	○	○
I would recommend it to a friend.	○	○	○	○	○
It is fun to use.	○	○	○	○	○
I feel I need to have it.	○	○	○	○	○
The system is user friendly.	○	○	○	○	○
The system requires the fewest steps possible to accomplish what I want to do with it.	○	○	○	○	○

I don't notice any inconsistencies as I use it.	○	○	○	○	○
Both student and teachers users would like it.	○	○	○	○	○
I can use it successfully every time.	○	○	○	○	○
I think that students should use this system.	○	○	○	○	○

How useful do you think ArithmeticNorm is? *

- [ ] I would find ArithmeticNorm useful for my studies.
- [ ] I think using ArithmeticNorm will enable me to accomplish tasks more quickly.
- [ ] I found that using ArithmeticNorm will increases my productivity.
- [ ] If I use ArithmeticNorm, I will increase my chances of successfully completing normalization course
- [ ] Other...

Would you like to test ArithmeticNorm system? *

- ○ Yes
- ○ No

To improve upon the website design, I would suggest the following:

Long-answer text

Is there are any issues that you want us to know related to the system? *

Long-answer text