Branch Credit Information System

(BCIS)

CHEONG KOK YAN

WEK 990056

FACULTY OF COMPUTER SCIENCE AND

INFORMATION TECHNOLOGY

UNIVERSITY OF MALAYA

SESSION 2001/2002

TABLE OF CONTENTS

3.0 System Analysis

4.0 System Design

5.0 System Implementation

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Branch Credit Information System or *BCIS* is a client-server system using *Wide Area Network (WAN)*, privately owned and provided by banks themselves, or outsourcing to other vendors, to connect to the database located at the HQ's server. *BCIS* will be use by branches' user to entry as well as add new entry for new borrowers and loans, update information of their existing borrowers, checking for the borrowers status at other branches, checking status of the loans whether they are performance loan or non-performance loan, and query for all personal information regarding to specify borrower.

The project will be developed systematically using *Waterfall With Incremental Model*. The purpose of this combination model is to take the advantages from one model and eliminate or lessen the disadvantages from the other. Besides, the system built to achieve error proof, good overall performance, reliability, availability, ease of use, and ensure data integrity.

ACKNOWLEDGEMENT

Development of this project involved a lot of efforts from many individuals, whom I like to appreciate my greatest appreciation. First of all, I would like to give my highest "salute" to my thesis supervisor, Pn. Maizatul Akmar Ismail, who wiling to spare her precious time to guide, and provide continuous support to me throughout the project development. Thank you, Pn. Maizatul !!!

I would like to express my gratitude to my thesis modulator, Encik Mustaffa Kamal, who wiling to spare his invaluable time and knowledge during the viva presentation.

In addition, I would like to thank Mr. N. Sivanasvaran, Executive Officer of Bumiputra-Commerce Bank Berhad (Section 14). He is very helpful, friendly, and kindly provided all the information I need during my interview with him.

Lastly, I would like to express my appreciation to all my fellow friends, and Faculty of Computer Science & Information Technology's staffs, for their continuous support, and helping me to complete this project successfully.

CHAPTER 1

INTRODUCTION

1.0 Introduction

1.1 Definition of *Branch Credit Information System*

Branch Credit Information System (BCIS) is a client-server application, using *Wide Area Network (WAN)*, privately owned and provided by banks themselves, or outsourcing to other vendors, to connect to the database located at the headquarter (HQ) server. Besides connecting between branches and headquarter, *BCIS* also need *Local Area Network (LAN)* to connect between *WAN* and the computers reside the *BCIS* application.

BCIS will be used by branches users to entry as well as add new entry for new borrowers profile at any branches, update information of their existing borrowers regarding their recent payment and status of the loan, checking for the borrowers status at the same branch and other branches, checking status of the loans whether they are performance loan or non-performance loan, query for all personal information and loans regarding to specify borrower.

BCIS will try to lift the burden of analysing the loans from the shoulder of the HQ's loan manager to the branches manager. With the help from *BCIS*, branch managers manage to create a full report accompanying the loan application to headquarter for approving process. Indirectly, the approval process will be shorter, reliable, and accurate, due to the manager or panel of officers do not need to waste, or divided their previous time on the unnecessary tasks.

1.2 Why The Existing System Unsuitable?

Financial market becomes more competitive everyday, especially when the implementation of globalisation is around the corner. Do our financial institution in Malaysia can stand the pressures, and globalisation power? This is among the questions that always ask by citizens in Malaysia, including the CEOs of the banks. The answer is surely "NO", if we keep our "antic" systems, and don't upgrade our local banks financial systems to become more advance, and fulfil all the needs at present, as well as in the future. In order to competitive with international banks, local banks have to always step one step ahead in front of them, and not just purchasing their "out-of-date" system.

1.2.1 Existing System Operation Too Complex

The existing loan system is too complex and not user-friendly enough for the User to adapt to the system and start doing their routine work right after the delivering without special supervision. Continuous trainings and supervisions need to supply to the Users in order to have the system operate without any problem.

As a result, the Users will feel afraid to use the system and may be creating unnecessary errors, like clicking on the wrong button and performing the incorrect procedures.

1.2.2 Late Approval Of Loan

Nowadays, fast loan approval will automatically created a best weapon for enhancing the business competitive. The existing systems are only performing as an interface for the branches Users to input the borrower information after the loan approved by Financial Institution headquarter. At branches, Users do not have permission to make any analysing on the loan because they normally do not have the priority or permission to view all the data regarding the borrower from the existing system, which classify as *private and confidential*. In additional, Users at branches do not have any power to do any approval of the loans.

Branch manager need to attach all the information and documents provided by the borrower and send them to headquarter for approval. At headquarter, the information and documents will be analysed by loan manager or panel of officers at headquarter. As a Financial Institution commonly has more than ten branches, the responsibilities of the loan managers at headquarter are very tiring and slowing the process. This problem will not be problem any more when these tiring tasks reallocated to the shoulders of the loan managers at all the branches.

This late approval of loan will create hoax and will make the Financial Institution reputation decrease. The Financial Institution needs a good system and changing on the business requirement to create a best system, which promise an environment of fast approval. We can see the essential for immediate or fast decision from the example *case** in the following page.

*(*All characters below not related to anyone, which are alive or death. However, if any similarity, is only coincidences)*

Mr. X is a normal man with a happy family. He wants to buy a house at Happy Valley Garden worth RM200,000. He went to HappyLoan Bank to get a loan of RM130,000. The bank's officer fills several forms with Mr. X's personal information, and his financial status. Then the officer using the existing loan system to check the status of Mr. X at the centre database. After about an hour, the officer gives a "sweet" promise for loan approval in one week's time.

Mr. X must be very happy. He will not hesitate to pay the house down payment of RM70,000 for his dream house. Do you know what happened after one week later?

At the same branch, one week later...

The bank's officer said "Sorry Sir. We do not approve your loan because you do not meet our loan approval requirements."

Alas! Mr. X lost all his deposit. He felt faint as all his saving for 10 years vanished...

- The End -

The slow decisions from the loan managers at headquarter, and not reliable current system will make people lost their confidence to the local banks. According to one of the advertisement by a foreign bank, it only needs 24 hours to approve a loan. Isn't it very attractive?

The weakness of the current system will be eliminate, or at least improved in the new version of branch loan system, *Branch Credit Information System (BCIS)*.

1.2.3 Incomplete Testing

System users usually come from non-IT educational background. They may be unconsciously entering incorrect data into the database if the system does not have a complete set of validation and restrictions. Nowadays, there are still many systems not tested by all categories of user, including those with IT background, and non-IT background. Current system only tested by the developers themselves that come from IT background.

As a result, some system will encounter "disaster" after a few days launched. This problem caused by the insensitivity of system developers to the needs of the users. Developers seldom included "normal" checking for errors, which always done by normal user (non-IT background). The unreliability system will cause the user lost their confidence to the system, and afraid to use the system in the future.

1.3 Objectives of *Branch Credit Information System*

Branch Credit Information System (BCIS) has objectives of development below: -

- Enhance the existing banks branches loan system to be more effective and promising, which provided accurate, and immediate information for the user;
- Provide user-friendly GUI for more error proof, and less training for first time user, as most of the users are not come from IT educational background;
- Full validation for data input to ensure data integrity;
- Decisions make by the loan manager can be done after undergo systematic and complete analysis on all information provided;
- Ease to use, as well as user-friendly because almost all the user at banks are not come from Information Technology (IT) educational background;
- Ensure data validation, and have full-cycle links (tables in database) by setting all necessary limitations to the user;
- System can perform operation of collecting required information and essential full set of analyzing on the data, before the branch manager come into any decisions of loan approval.

1.4 Why *Branch Credit Information System*???

1.4.1 More Reliable and Promising

Branch Credit Information System (BCIS) is more reliable and promising, and may be always looking forward by the Users. *BCIS* will provided what the Users need badly, ease of use. Normally system like credit system is very hard to use and need a lot of training before the Users do the right procedures. This is will badly decrease the confident of the User to continue using the system. *BCIS* is coming to rescue by providing all the necessary limitations in the system and instructions to the Users.

Besides, *BCIS* also providing features for verifications and validations to ensure the data integrity. If the Users enter an invalid input like invalid date, the system will automatically prompt out to tell the Users to make modifications to the inputs.

BCIS also provides maintenance options for the Users to cover the Users from the complexity of the backend database. Users can backup and restore the database from the same system. This feature avoids the Users accidentally corrupting the database and violates the data integrity.

1.4.2 More User-Friendly

BCIS offer a user-friendly interface, complete instructions, professional help, and ease-of-use features to the user, even if the user does not have any IT knowledge. Due to the target user, which does not know how to use the system, *BCIS* offered a good set of limitations, as well as validation, in order to filter out the error that always done by the user. Due to this reason, different users with both IT background, and non-IT background will invite to test *BCIS* continuously in different situations.

The familiarity of the system's features by the user will directly maintain a good condition of the database, and ensure data integrity. The target of *BCIS* is to be an error-proof system, and easy to use as "ABC ".

1.5 Scope of Development

1.5.1 Scope of *Branch Credit Information System*

BCIS is a client-server system installed at client (workstation), and linked with database located at server (server). This system only installed at client. The scope of development included: -

- Friendly user-interface to input data from bank branches to the database located at Headquarter;
- Easy and fast query for any information stored at database;
- Delete existing customers after validating the deleting meet all the procedures and requirements;
- Checking for data integrity, and full-cycle link of specific record before updating to the database;
- Easy database maintenance, such as backup and restore database;
- Different level of system access to increase the security;
- Encryption for user password to increase security;
- Administrator could set the system setting to satisfy the users need;
- Clear instructions for every features and functions that available;
- Professional Help for the users; and
- User-friendly interface and error proof (data integrity).

1.5.2 Limitations

BCIS has a few limitations. The limitations create intended, or need continuously enhancements. The limitations/weakness are: -

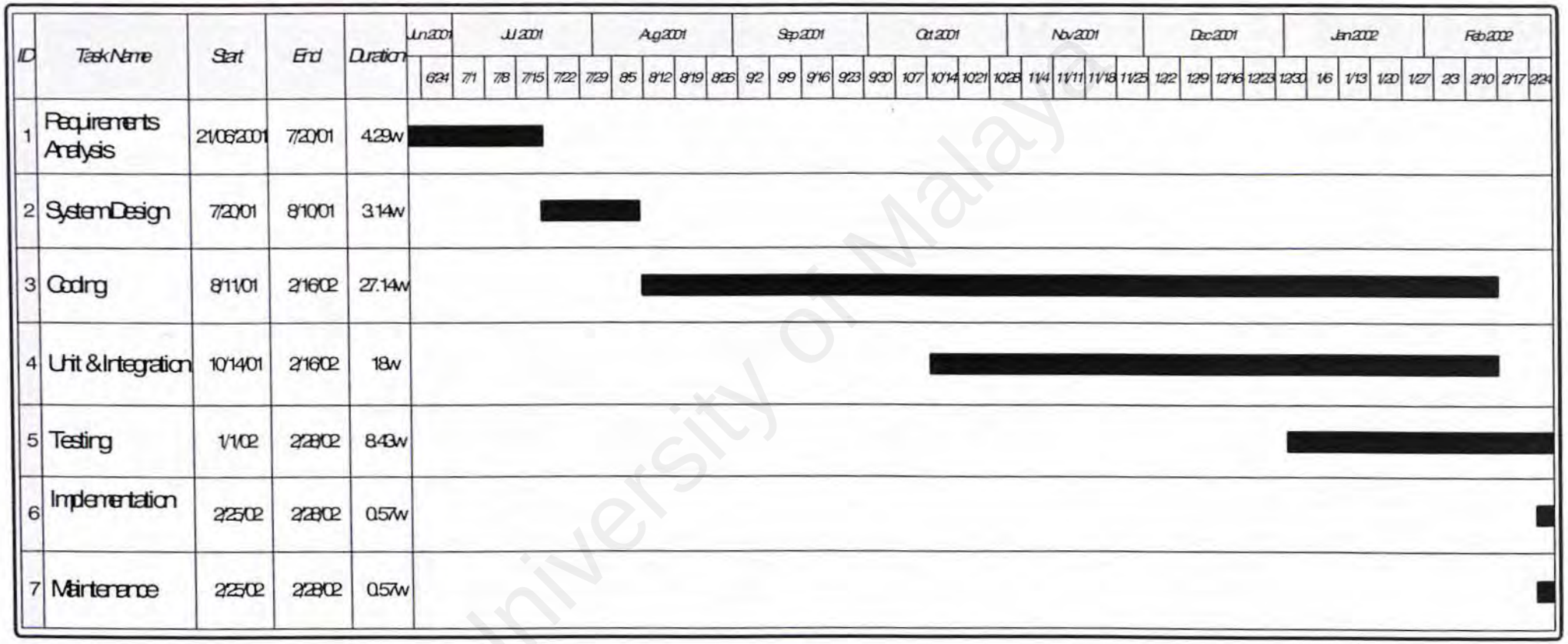
- Not platform compatible. *Branch Credit Information System* only suitable for Windows based platform;

- *Branch Credit Information System* needed to be enhance/update/review in new versions from time to time following the fast changes/update in business requirements.

1.6 Project Schedule

Development of *BCIS* is following the schedule at *Figure 1.0*.

(Please refer to the following page for the diagram of the project schedule)



1.7 Methodology Used

Branch Credit Information System (BCIS) developed based on the *Waterfall Combining with Incremental Model.* This methodology used because *BCIS* is a system that need active communication with the future users, in order to develop a system that really suit the need of the users. The advantages of using this model highlighted in *Chapter 3.*

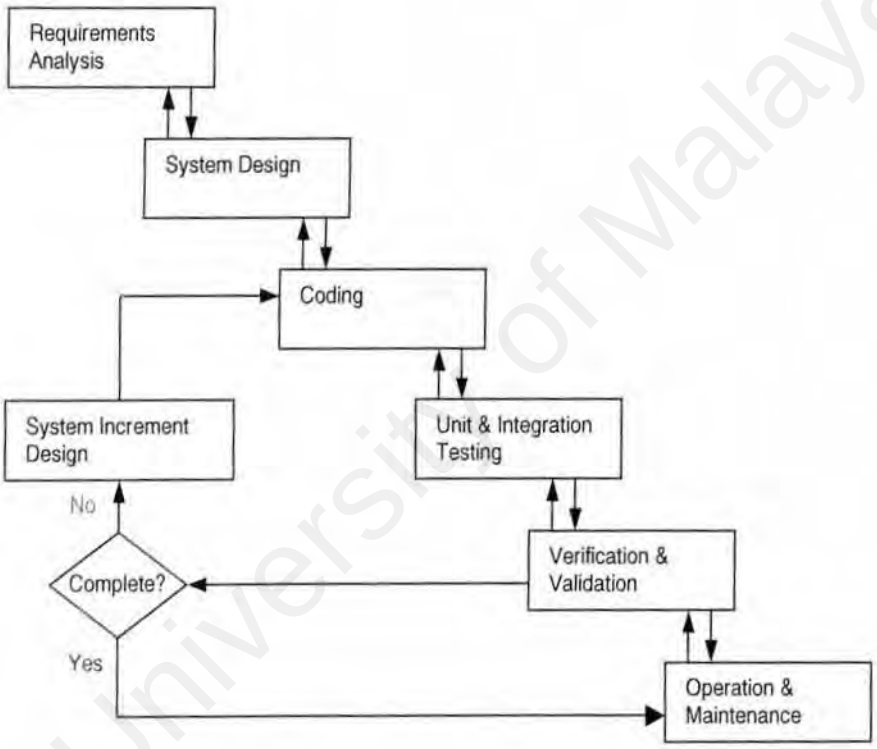


Figure 1.1: Methodology Used

CHAPTER 2

LITERATURE REVIEW

2.0 Literature Review

2.1 Why Literature Review Important?

Literature Review is the most important part and also the richest information part, as well as details about this project. Literature Review even considered by many researchers as the most important part of the project. The benefits that we can gain from this part highlighted below: -

- Gives the developers a precious opportunity to study about the additional knowledge and information gained from various resources to develop this project;
- Places the development project in the context of the other existing systems that have the similar characteristics;
- Helps the system developers to know the existing features and functions offered by similar systems;
- Opportunity for developers to do comparison on the past developed projects to study strength and weakness of the existing systems; and
- Overview for the developers to improve weakness and fulfill requirements needed.

2.2 Data Gathering

As this system is mostly based on the user requirements and business constraints, the methods suitable used are interviews, white papers, observation and analysis on the existing system.

For interviews, users or potential users are interviewed to collect their ideas and information, especially about the loan approval procedures, and the existing system.

Analysis done on the loan procedures and requirements with the help from white papers obtained from Internet. These white papers highlight standard loan approval procedures and its requirements. However, different banks have different loan approval procedures, based on their strategic planning.

Observation of the existing system is done at the banks, with the help from the banks' staffs. However, observation is strictly limited to just a few features, as the *Private & Confidence (P&C)* data cannot be exposed to third party.

2.3 Overview Of Loans

2.3.1 Introduction

Security is often regarded as most important qualification for a loan. This is not entirely true. While security is normally required for a proposed loan, it is only one of the factors the banks takes into consideration. A bank, in determining whether a proposed loan is desirable or not, irrespective of whether security is offered, also takes into account other factors [16].

2.3.2 Purpose of the loan

The purpose of a loan must fulfils banking policy, and not breach any of the directive and controls imposed by the Bank Negara of Malaysia, and the Government.

Banks invariably provide short-term capital to manufacturers for the purpose of raw materials, to businessmen as working capital, to traders to enable them to increase their stock and long-term loans to purchasers of houses, often repayable over a period of a number of years [9].

All banking activities in the country are subject to directives and controls issued by Bank Negara and Malaysian Government. The Bank Negara from time to time will issue certain guidelines to all the commercial banks [16].

2.3.3 Amount of Loan

The amount of loan invariably depends on the type and size of business of the borrower. It is necessary to obtain from the borrower a statement of his asset and liabilities [9].

In cases where the borrower has his own business it is always prudent to restrict the amount of the loan to below a reasonably acceptable level of his own resources where the bank can feel comfortable.

In cases where the borrowers are large trading concerns, such as partnerships or limited companies, it is necessary to obtain from them copies of their balance sheets and trading and profit and loss accounts for a least the previous three years of business [8].

2.3.4 Duration of the Loan

As a general rule the bank retains the right to demand the repayment of a loan at ant time. In other words a loan to a borrower is repayable on demand. In practice, however, the bank very rarely recalls a loan on demand, except in very exceptional circumstances. Even so, this often brings anxiety and uneasiness to the borrower [9].

The provision of short-term capital and medium-term finance for trade, commerce and industry reflects the primary function of banks in this country. Occasionally there are some special categories of borrowers who may be allowed to enjoy credit facilities well beyond the usual period [8].

Banks had been actively providing long-term housing loans for the purchase of dwelling houses which are repayable over a period of a number of years, some as long as thirty years [16].

2.3.5 Source of Repayment

Another important matter for consideration is the ability of the borrower to repay his loan. Temporary self-liquidating loans pose little or no trouble at all to the banks. Examples of these self-liquidating loans include: -

- A loan to finance a borrower's second house on the promise that the loan will be satisfied within the next six months from the proceeds of sale of his first house;
- A loan against the security of a life policy which is due for maturity soon; and
- A loan to a company as working capital pending the raising of its permanent working capital.

Considerable care has to be exercised when a borrower requests for a loan as additional working capital and the loan is to be repaid out of the profits extended over a period of years [16].

2.3.6 Types Of Loans

Malaysia's local banks offer variety of loans to their customers, classify according to their maturity, type of collateral, and other special features. Banks offer four main loans to meet the customers' requirements and needs – short-term loans, bridge loans, revolving credit loans, and term loans [16].

Over than a half of bank commercial loans are made for a short term, that is, period of less than a year. Most of these loans are for financing increases in inventory for seasonal borrowers. The loan is repaid when the borrower's inventory is sold and its receivables are collected. Short-term loans also finance borrowers with short-lived and projected-oriented needs for funds – for example, service businesses such as accounting or engineering design firms.

Bridge loan is another type of short-term loan. Bridge loan can be thought of as project-type loans that bridge a period of time up to a specific event that generates sufficient funds for repayment of the loan.

Revolving credit loans finance the expansion of current assets or the retirement of current liabilities. This type of credit often is called asset-based lending because the amounts borrowed are tied to a borrowing base formula that limits the outstanding amount to a margin percentage on the borrower's receivables, inventory or extractive industries, reserves owned. This credit is the long-term equivalent of short-term line-of-credit loans entailing a commitment to advance funds up to a maximum line for longer terms, that is, for as much as five years. The need for funds arises most often because the borrower cannot fund its increasing sales from internal sources such as retained earnings.

Term loans are loans with maturities over one year. Often they finance the purchase of fixed assets or the broad expansion of production capacity, but they may also be made to finance a change in company control or an acquisition or to take out a revolving credit loan. Term loans have advantage over bonds because term loans can be executed quickly, flexibly, and with low issuance costs. Most term loans have immediate maturities of less than 10 years [8].

2.3.7 Loan Securities

Since banks are theoretically in the business of taking risk, most of the banks do not want to be concern of any more unnecessary risk and feel of insecure. Banks always require a borrower to furnish security for his loan. This is necessary even though the borrower's financial position may appear to be sound for there is no

certainty that his position will not change. The fact is a borrower's position can change, and sometimes, quite quickly. Security, hence, constitutes a very important part in a bank's deliberation of a borrower's request for a loan [8].

Loans can be categorized as secure/clean and insecure, depending to the reputation of the borrowers, borrowers' financial history, borrowers' financial status, and value of properties owned by the borrowers.

For the insecure loans, banks require borrowers to prove the security of the loan by attach the permitted types of securities. The examples of securities that normally required by banks are: -

- land or landed property
- letter of guarantee
- letter of credit
- stocks and shares
- cash and fixed deposit receipt
- debentures
- inventory machinery
- contractual rights
- insurance policy
- ships, aircrafts and other carriages
- miscellaneous

Normally, banks only approve maximum of 90% loan with securities to lessen the risk [9].

2.3.8 Loan Classifications

Loans can be classified as different types of loans, depending to the borrowers' reputation, payment history, securities attached, financial status, stage of loans review, and other requirements.

Loans review divided into two stages – beginning of the loan application, and loan payments term. At the beginning of the loan application, loan will be classified as secure loan, or insecure loan. This kind of classification is more depending to the borrower's reputation, financial status, and whether the borrower is currently under any legal action.

The second category of loan classification is whether the loan is a performance loan or non-performance loan. This kind of category is making totally depending to the history payments, and end-financing year review. Loan that classified as non-performance loan is the loan that has payments not following the stated term. Normally, banks will classified a loan as a non-performance loan when the borrower does not make any payments for three months/term or more [9].

2.3.9 Loan Approval Procedure

Not all the banks in Malaysia have the same loan approval procedures. Different banks have different requirements, times, and procedures to approve loans. The banks with the guidelines from Bank Negara Malaysia, modifying and updating the loan approval requirements and procedures to competitive among themselves for attracting their current, as well as the future customers.

A loan can be approved in three days to two weeks time depending to the procedures, the clearance of information provided by the borrowers, the burden of works on the loan manager, power to approve loans, and other legal procedures.

Normally, a borrower must fill in a application form, for example his/her name, identification card numbers, occupation, his/her spouse information, properties owned, amount of loan, tenure of the loan applied, and others information for the reference of branch's loan manger and the headquarter. Branch loan manager will do some analysis regarding to the information provided by the borrower about his/her background, prospect, and risk factors involved. After this, the loan application will be hand out to the bank headquarter to get approval. The staffs or officer-in-charge at the headquarter will make a complete analysis on the borrower with the help from the earlier analysis done by the branch loan manager. The analysis will include all the details about the background of the borrower, the history of loans from any branches at the same bank, or from another banks,

history of payments, financial status, and whether the borrower is under any legal actions, civil or criminal offences.

After all the related analysis done, the documents will be pass to a panel of officers, or managers to approve the loan. A loan will be put under several condition, including on hold, not approve, delay approve, and approve. Any confusing or unclear information that provided by the borrower will make the whole process restart from the very beginning.

The results will be notified to the customer by the branch. The branch will be given the result and for the unsuccessful loan there are also included the reasons behind the failure. According to the past cases, these entirely loan approval procedures will needed at least three days or may be longer than two weeks [9]. Refer to *Figure 2.1* for the summary of the standard local commercial banks' loan approval procedures.

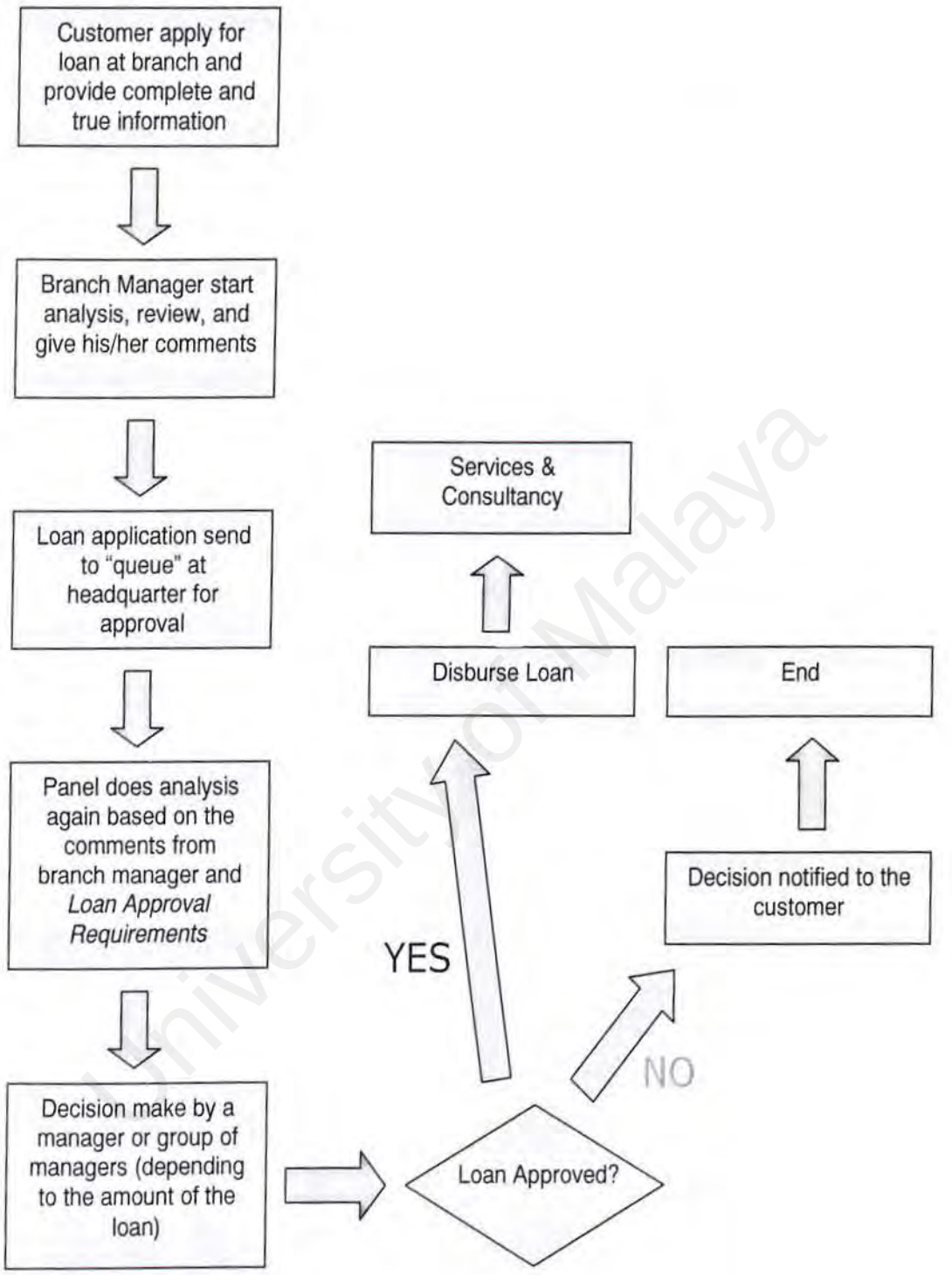


Figure 2.1: Summary of Standard Loan Approval Procedures

2.4 Review Of Existing System

2.4.1 Overview of Existing System

Banks have existing systems that use for supporting the daily process of loan application from the future customers. However, the existing system does not really involved directly in the approval process. The systems normally only consists of functions that providing static information. Banks do not have any loan approval *decision support systems* to help the decision maker to make mature and accurate decision.

Existing systems performed as an interface to input data into headquarter database, query information regarding with the specify borrowers from the same branch or branches, generate monthly payments reports, and connect the branches' system with the database at headquarter for checking purpose and updating. The existing systems also use to check for the amount of money borrowed from any branches, and the history of payments [9].

2.5 Software Evaluation

2.5.1 Database Management System (DBMS)

Microsoft SQL Server 7.0

Microsoft SQL Server 7.0 offers a solution that's aimed at rapid development and low-cost implementation. Microsoft SQL Server 7.0 run limited on Windows based platform. Built-in Internet integration allows users to conduct business on the Internet and build intranet sites. Microsoft SQL Sever 7.0 has increased many fronts depending to the previous versions, and terabyte database are supported. SQL Server 7.0 leads in price-performance by reducing administrative overhead and lowering total cost of ownership with a design that is geared toward dynamic configuration. Many of the day-to-day manual tasks of the database administrator have been automated. Dynamic locking, dynamic memory allocation, auto-grow tempdb, auto-grow log files, auto-grow database files, automatic creation of the database upon a restore, automatic index creation across an entire database, and many other new features help eliminate administrative tasks. Microsoft SQL Server 7.0 minimizing complexity for users, administrators, and developers allows application to be deployed at a lower cost and in less time. Microsoft SQL Server 7.0 offers a cheap implementation, which will required about US$1,999 for 10 clients or US$3,999 for 25 clients [17].

Oracle

Oracle platform is available for multiple operating systems and research proved that Oracle runs great on Unix. Oracle is more standards-based as well with a set of neat features. Oracle databases are as powerful as the users want them to be. Oracle also is able to efficiently utilize hardware platform that and manage multiple high-speed processors, clustered servers, high bandwidth connectivity and fault tolerant storage technology. Java application can run perfectly with combination of Oracle database. Oracle also provided the users with more power and flexibility with the database to meet the user requirements. Oracle able handles a rapidly expanding amount of users and/or data gracefully [17].

One of the disadvantages is Oracle has weird concepts and names as well. As a result, users have to be undergoing specialized training/knowledge to be more familiar with Oracle database management; even the experts of other DBMS, like Microsoft SQL Server and Microsoft Access. Besides, Oracle needs a costly start-up solution of database management. For example, Oracle has a package solution that starts at $6,767 running on Unix platform [18].

Microsoft Access

Microsoft Access provides users with one of the simplest and most flexible DBMS solutions on the market today. Regular users of Microsoft products will enjoy the familiar Windows "look and feel" as well as the tight integration with other Microsoft Office family products. An abundance of wizards lessen the complexity of administrative tasks and the ever-present Microsoft Office Helper is available for those who care to use it. As Access is a state-of-the-art application for Microsoft Windows, user can use the facilities of Dynamic Data Exchange (DDE), ActiveX objects, and ActiveX custom controls. DDE can be used as connections to other applications using macros or Microsoft Visual Basic. ActiveX is an advance Windows capability that, in part, allows users to link objects to or embed objects in Access database. Access 2000 can also act as an ActiveX server, allowing users to open and manipulate Access database objects, such as tables, queries, and forms, from other Windows-based application. Microsoft Access has added benefits of ease of use reports manager, web integration, and SQL Server integration [13].

2.5.2 Programming Language

Microsoft Visual Basic 6.0

Visual Basic 6.0 is one of the products in Microsoft Visual Studio package. Visual Basic 6.0 allows the users to develop Windows application quickly and easily without being expert in other programming languages.

Visual Basic provides a graphical environment in which user can visually design the forms and controls that building blocks of applications. Visual Basic supports many tools, especially to build 3-D Windows "look and feel" interface that will help users be more productive, as the users do not have to waste unnecessarily time on the interface. The tools are not limited to, projects, forms, class objects, templates, custom controls, add-ins, and database managers, users also can use the tools together to create complete applications in months, weeks, or even days; producing an application using another language can take much longer [12].

Visual Basic continues to sport the Explorer-style development environment, modeled after Windows Explorer. This makes it easy for a user to jump right into creating applications with Visual Basic. Almost all the objects and tools on the screen can be manipulate through a right-click. Users can set properties, add controls, and even view context-sensitive help with this single action. Visual Basic 6.0 has special features – ISAPI Application and Dynamic HTML project templates. These templates provide you with a framework to develop server-side components as well as "smart" Web pages and applications [7].

Java 2

Java is a powerful computer programming language that is fun to use for novices while simultaneously being appropriate for experienced programmers building substantial information systems. Java programmer can take advantage of rich collections of existing classes in Java class libraries to form a complete Java program. The class libraries are known as the Java APIs (Applications Programming Interfaces). Normally, class libraries are available in large quantities from the Internet and World Wide Web as freeware or shareware.

Java applications can use the Internet to interact with other applications and with databases. These capabilities allow Java programmers to develop the enterprise-level distributed applications used in industry today. Java application can be written to execute in any computer platform, yielding major savings in system development time and cost for corporations. The disadvantage is the time-consuming and complex effort that goes into designing and developing new classes and methods. Java programs execute interpretively on the clients machine. Interpreters execute slowly compared to fully complied machine code [6].

2.6 Client-Server Architecture

As the term suggests, a client-server environment is populated by clients and servers. The client machines are generally single-user PCs or workstations that provide a highly user-friendly interface to the end user. The client-based station generally presents the type of graphical interface that is most comfortable to users, including the use of windows and mouse.

Each server in the client-server environment provides a set of shared user services to the clients. The most common type of server currently is the database server, usually controlling a relational database. The server enables many clients to share access to the same database and enables to use of a high-performance computer system to manage the database. In addition to clients and servers, the third essential ingredient of the client-server environment is the network. Normally, clients and servers linked by a single LAN or WAN. The main purposes of implementation client-server are: -

- There is heavy reliance on bringing user-friendly applications to the user on his or her own PC;
- Filter the user from the complexity of the database, and the server's operating system, which is sometimes hard to manage by an ordinary user;
- Local banks emphasis on centralizing loan database, so the bank management enable to maintain control of the loans borrowed by customers at different branches in Malaysia; and

- Growing availability and affordability of microcomputers and networks [20].

Implementation of client-server is easy and cheap to maintain. Please refer to the *Figure 2.2* for the diagram that shown how the client-server architecture implemented successfully.

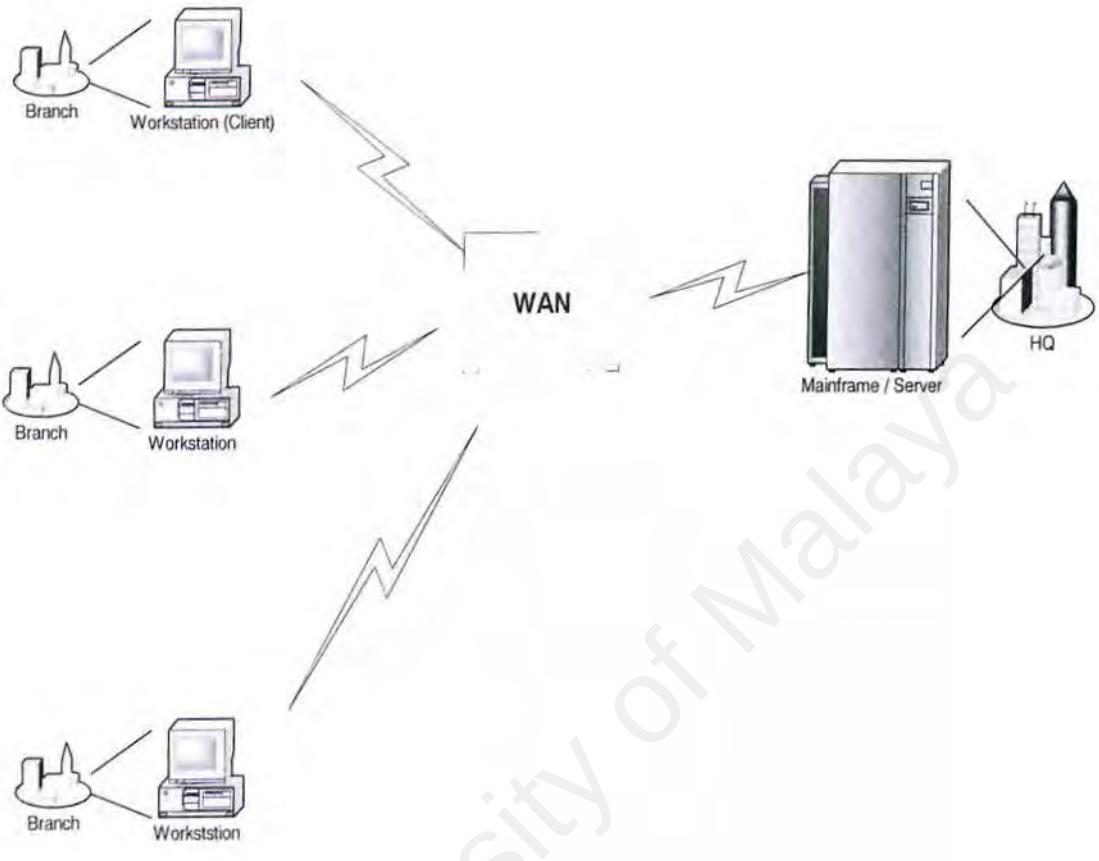


Figure 2.2 Client-Server Architecture

CHAPTER 3

SYSTEM ANALYSIS

3.0 System Analysis

3.1 System Analysis

System analysis is the study of a business problem domain to recommend improvements and specify the business requirements for the solution of new problems. System analysis is a term that collectively describes the early phases of systems development.

System analysis also included the method of studying requirements, methodology used, and the possible improvements of the system.

Method is an orderly procedure for accomplishing a specific task. Methodology is a collection of methods, tools, standards, practices, and always used in software development projects. There are many types of methodology that can be used based on project characteristic – size, reliability and safety requirements, team expertise, budget, and time [19].

3.2 Methodology Used

Methodology used to develop *BCIS (Branch Credit Information System)* is *Waterfall Model* combining with the *Incremental Model*. The purpose of this combination is to take the advantages from one model and eliminate or lessen the disadvantages from the other. For more information about this modified model illustrated in *Figure 3.3*.

3.2.1 Waterfall Model

Royce brought the first waterfall model into the life of software development in 1970. Waterfall model is the model where the stages of development are depicted as cascading from one to another. One development stage should be complete before the next begins. The waterfall model has been used to prescribe software development activities in a variety of contexts [19].

When all the requirements and specification elicited from the customer, analyzed for completeness and consistency, and documented in requirements documents, then the development process can continues to the following stage – system design. The waterfall model presents a very high-level view of what goes on during development, and it suggests to developers the sequence of events that should be expect to encounter. Its simplicity makes it easy to explain to customers who are not familiar with software development; it makes explicit which intermediate products are necessary in order to begin the next stage of development.

Original waterfall model is modified into a better model, which frequently review of the previous stage based on the feedback from customers and review by the developers. This model is widely used by many developers because it is more reliability than the original model [19]. The modified version of waterfall model is shown in *Figure 3.1*.

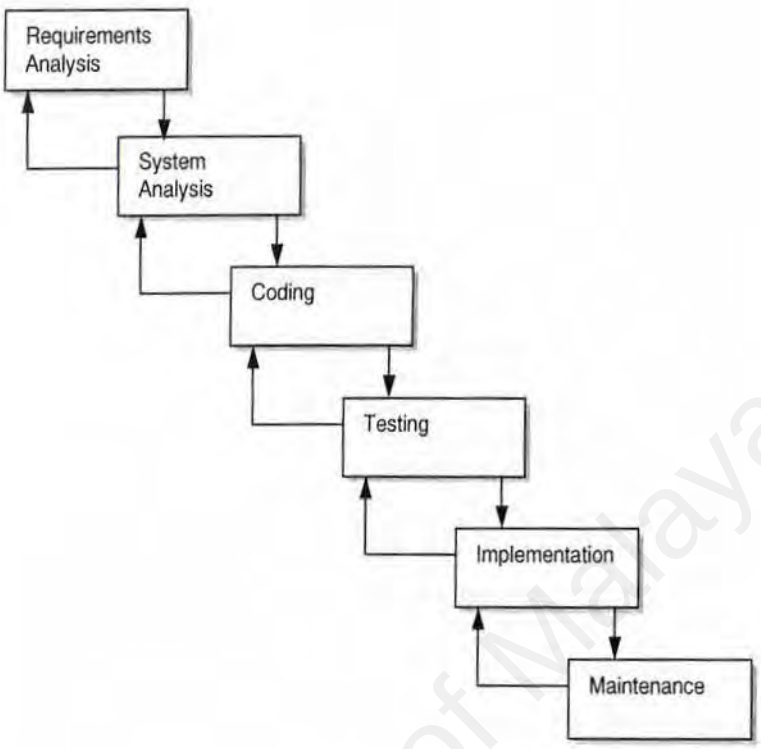


Figure 3.1: Modified Waterfall Model

3.2.2 Incremental Model

Incremental Model development avoids the problem of constant change, which characterize evolutionary prototyping. Normally, an earlier process of system architecture is established to act as a framework. Systems component are incrementally developed and delivered within this framework.

The framework and the components are not changed after validated and delivered, unless errors are discovered. However, even after the framework together with the

newly finished components delivered to the users, the design and additional requirements can be influence by the users' feedback.

Although incremental model used continuing increment system components, developer and users will first focus on the essential, or main features. Additional functionality, like more security measures needed, included only if and when it is needed by the users after they get first contact with the newly built system. This model will give a good impression to the developer or the vendor and gain confidence of the users, as the system developed tend to be more reliability, and provided sufficient support to the users after delivered [19]. *Figure 3.2* illustrates the *Incremental Model.*

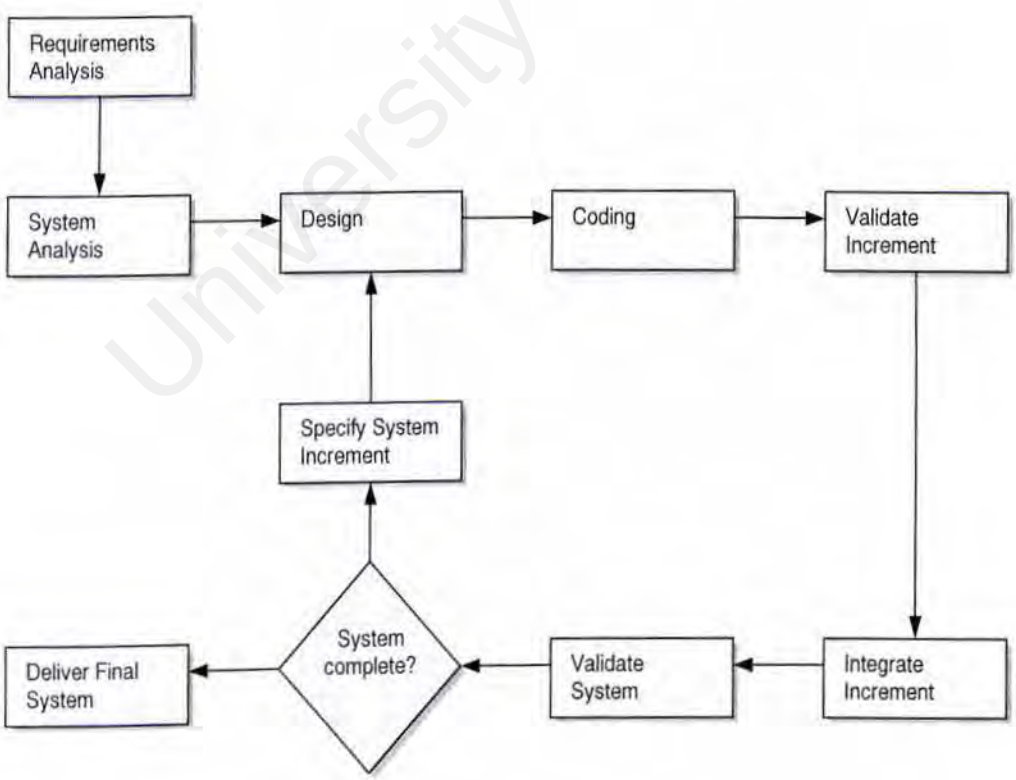


Figure 3.2: Incremental Model

3.3 Why Waterfall Model Combining With Incremental Model?

Branch Credit Information System (*BCIS*) is a system designed mainly targeted on user-friendly purpose, and continuous increment features and functions, like additional field for data entry.

Advantages gained from this combination of two models – Waterfall Model and Incremental Model (Figure 3.3) in developing of this system are: -

- The simplicity of this model makes it easy to explain to customers, or future users who are not familiar with software development, about the operation, and functioning of the system;
- Presents a very high level view of what goes on during development, and the possible events, which the developers will encounter;
- More control on the time as this model enabled project managers to gauge how close the project is to completion at a given point of time;
- Avoids the problem of constant change, or "big bang effect" because the system will grows with additional system until the system meet all the requirements, and the users are closely involved in the system development;
- System is encouraged to develop in less constrained by the fix architecture for easier to modify its design and increment additional functions and features in the future without developed the system again from the very beginning;

- The system can be put to functionality early in special case, for example to gain market competitiveness, even before all the features and function specify integrated into the system;
- Training can begin on an early release of the framework, or consider finished system, even if some functions are missing. The training process allows developers to observe the obstacles encountered by the users, as well as how certain functions are executed in a non-technical environment; and
- Validation stage allows developers to fix unanticipated problems globally and quickly, as well as improving system performance.

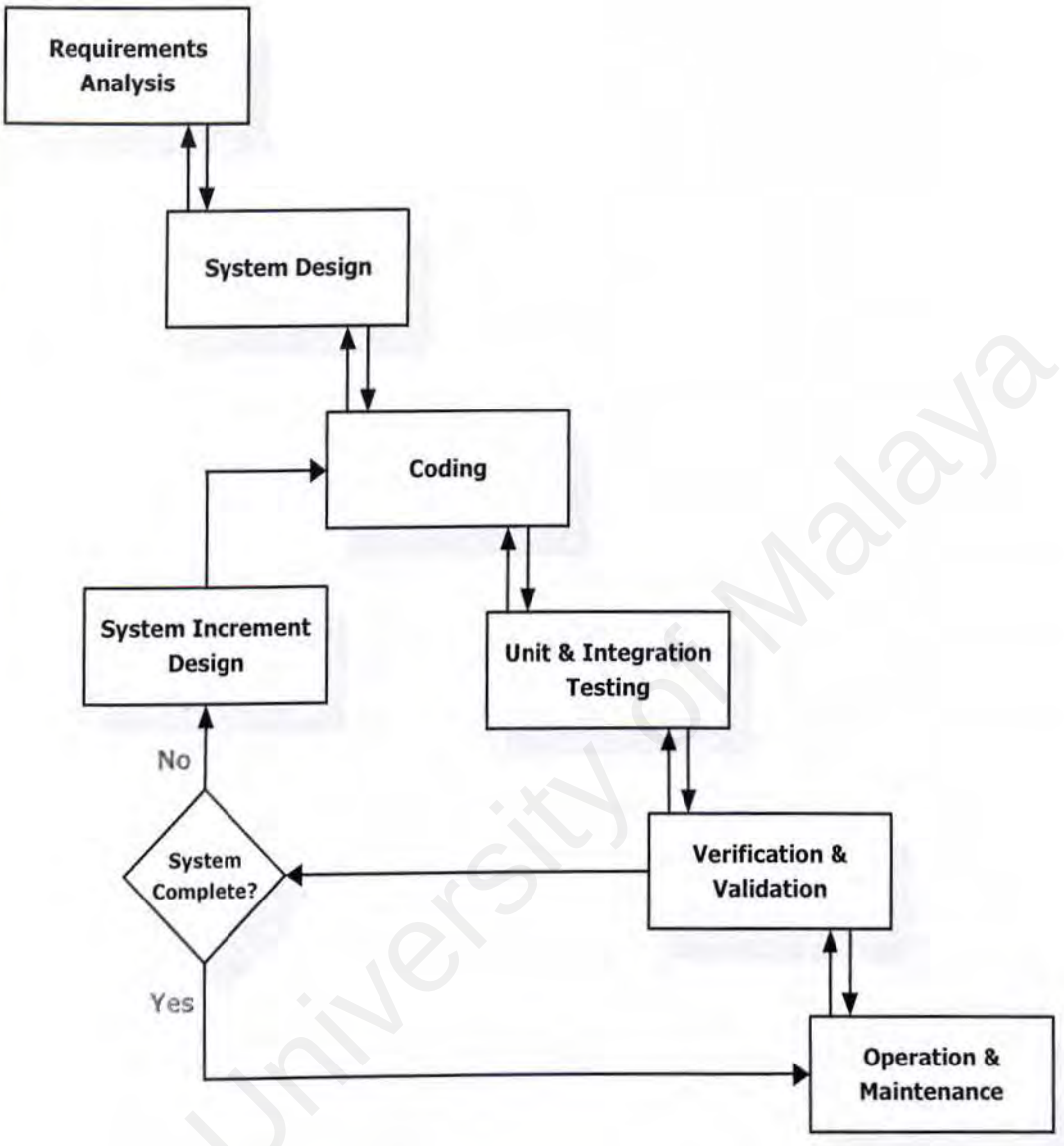


Figure 3.3 Waterfall Model With Incremental Model

3.4 Choice of Development Technologies

After analysis on the strengths, advantages, constraints, and limitations of the various technologies – software and hardware, this system will be built using programming language Microsoft Visual Basic 6.0, DBMS Microsoft SQL Server 7.0, client-server architecture, platform Windows 98 in client side and platform Windows NT/2000 in server side.

For the development of the *BCIS*, Microsoft Visual Basic 6.0 is chosen for its ease to use, thus more focus on the designing and developing the functions of the system. Visual Basic integrated well with Windows based platform, and run faster than its competitive programming language – Java. System developed with Visual Basic has a Windows based graphical user interface, thus giving the users more familiar with functions available, less training, and user-friendly.

Backend system consists of a *Database Management System (DBMS)* installed at Server side. The DBMS chosen is Microsoft SQL Server 7. SQL Server is chosen comparing with Microsoft Access 2000 because SQL Server is more reliable; support a large number of concurrent users, good security features, and good database management, as well as support a *Very Large Database (VLDB)* that may be in terabytes of data.

Client-server architecture of network is chosen because it is more reliability, ensuring good security measures, learner to add additional clients, and easy to

implemented. Client-server is more reliability than other network, like Internet, because client-server is normally using *Local Area Network (LAN)* to connect between them. *LAN* can guarantee safe and sound transportations of data without concerns about the data secrecy and integrity.

Windows platform is chosen for the client and server side. The reason for this choice is Windows platform is easier to use and many users already use Windows before, thus they need less training and can maintain the system well. Besides, Windows platform designed to work perfectly with Visual Basic program and SQL Server.

ODBC or *Open Database Connectivity* is used to connect the client side with the server side. *ODBC* is chosen because it is fast, easy to manage, and allows access by every users at the same client side.

3.5 Hardware & Software Requirements

3.5.1 Server Hardware Requirements

- Server with not less than 350MHz processor
- 64MB RAM memory or more (recommend 128MB)
- 2GB of free hard disk space or more (depending to the size of data)
- Network Interface Card (NIC) and network connection with recommended bandwidth at 10Mbps or more
- Other support peripheral devices such as *Uninterruptible Power Supply (UPS)*

3.5.2 Client Hardware Requirements

- Processor not less than 300Mhz
- 64MB memory or more (recommend 64MB)
- Network Interface Card (NIC)
- Other peripheral device for input and output (I/O) operation, such as keyboard and mouse

3.5.3 Server Software Requirements

- Operating System Platform – Windows NT/2000
- Database Management System – Microsoft Server 7.0

3.5.4 Client Software Requirements

- Operating System Platform – Windows 95/98/NT/2000
- SQL Server 7.0 ODBC Driver

3.6 Requirements Analysis

3.6.1 Functional Requirements

Functional Requirement is a description of activities and services a system must provided.

Add New Entry

The system provided users to add any new entry of customers at any time. User can add new data for new borrowers, or new loans for the same borrower. All entry will be validated for the completeness of link between data, and the system limitation of the data types (ensure data integrity). The system automatically check on the data entered by the users, to make sure if the customers already exist in database or not, to avoid duplicate entries.

Edit Existing Borrowers Information

User can modify the specific borrowers information, except some sensitive field, such as the principal outstanding. System administrator authorization and login required for gaining full access to all the editing features of the system.

Delete Existing Customer

User does not have permission to delete any borrowers from the database. Only system administrator has permission to delete the borrowers. Deleting situation need to satisfy all the requirements in loan requirements before the system

proceed. Deleting will cleared all data in the HQ database, that directly linked to the borrower and him/her loan, to ensure no "incomplete cycle" to avoid lost link.

Search Specify Customer

User can search for borrowers from the same branch or other branches. The search engine is using wild card to perform searching through the database with borrower number (assigned by the system), I/C number, and name. The customers will be search for the same branch and also for other branches, unless it specified by the user. All required data will be display and additional information also can be obtained if the users need it for special case.

User Authentication And Authorization

Users have to login with the username and password provided by the system administrator. However, User has to change the password periodically, or whenever he/she like after granted access to the system. Failed logins more than the permitted times with automatically lock the User. Users have limited access to the features of the system. Only system administrator has full access to all the features and setting of the system.

Add New Users

System administrator has the authority to add new users by provided the future users their username and default password. The system will prompt out to direct

the User to change their password immediately after granted access to the system for the first time.

Delete Existing Users

System administrator can delete the existing system after he/she login into the system. The users deleted from the system login list no longer can enter the system.

Block/Unblock Users

System administrator able to block users that suspected will harm the system or the data itself, or for system maintenance. System administrator also can unblock the users, depending on initiative of the administrator himself/herself. Besides, system administrator can change setting to block the User for specific period of time, or unlimited time.

Database Maintenance

System administrator can does the basic database maintenances, such as backup and restore database. This system helps to hide the complexity of database from the system administrator

3.6.2 Non-Functional Requirements

Non-functional requirement is a description of other features, characteristics, and constraints that define a satisfactory system.

User-Friendly Interface

The system has a user-friendly interface with the "feel and look" of normal Windows based applications to reduce learning curves. Meaningful and descriptive button and icon is used to add the user-friendly environment to the system.

Reliability And Availability

The system will be tested by two categories of users – technical and non-technical. This is to avoid errors caused by the unfamiliarity of the limitations of the system by different background users. The reliability and availability is very important to protect the rights of the customers, and as well as gain market competitiveness for the banks.

Meet User Need

With the combination of *Waterfall Model* and *Incremental Model*, the system will continuously increased and try to meet all the User's need in the incremental process. In special case, such as, a bank need badly to implement the system earlier than the schedule, components or half-finished system with essential features, can be delivered first and followed by continuously update.

CHAPTER 4

SYSTEM DESIGN

4.0 System Design

4.1 What is System Design?

System design is defined as those tasks that focus on the specification of a detailed computer-based solution, on the technical or implementation. It is also called physical design.

4.2 Architectural Design

Large system is decomposed into sub-system that provides almost all related set of services. This is the initial design of identifying sub-systems, establishing a framework for sub-system control and communication, and also relationships of the sub-systems. Below are

- Structure charts showing sub-systems that available in the system (*Figure 4.1)*;
- Flow chart to describe the actual processes of the system (*Figure 4.2*);
- *Data Flow Diagrams* (*Figure 4.3.0 – Figure 4.3.4*) to provide the of the processes and the data flow; and
- Diagram (*Figure 4.4*) to describe the operations of the system to add understanding of the processes, components, and location of the distributed application.

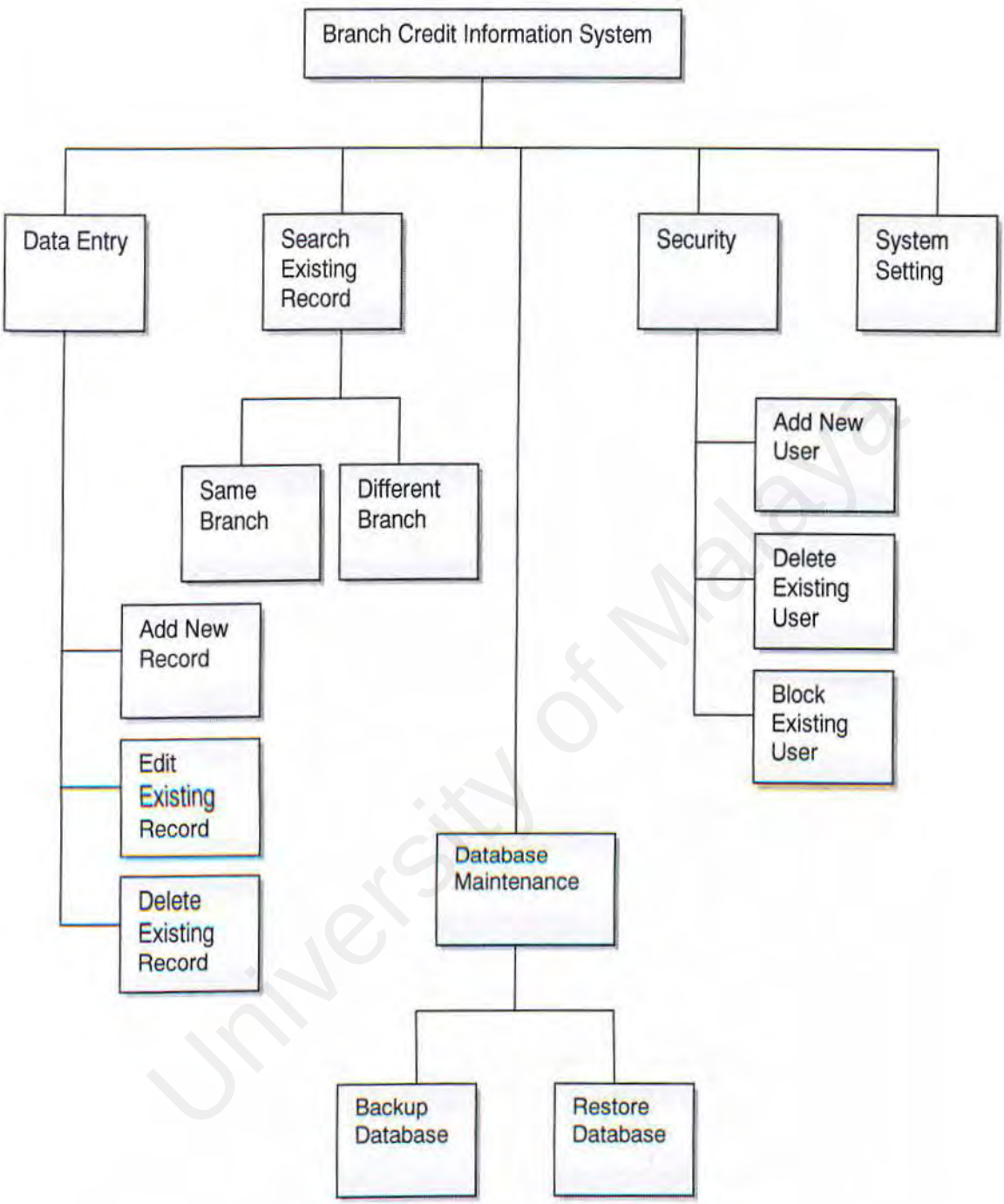


Figure 4.1 Structure of the BCIS

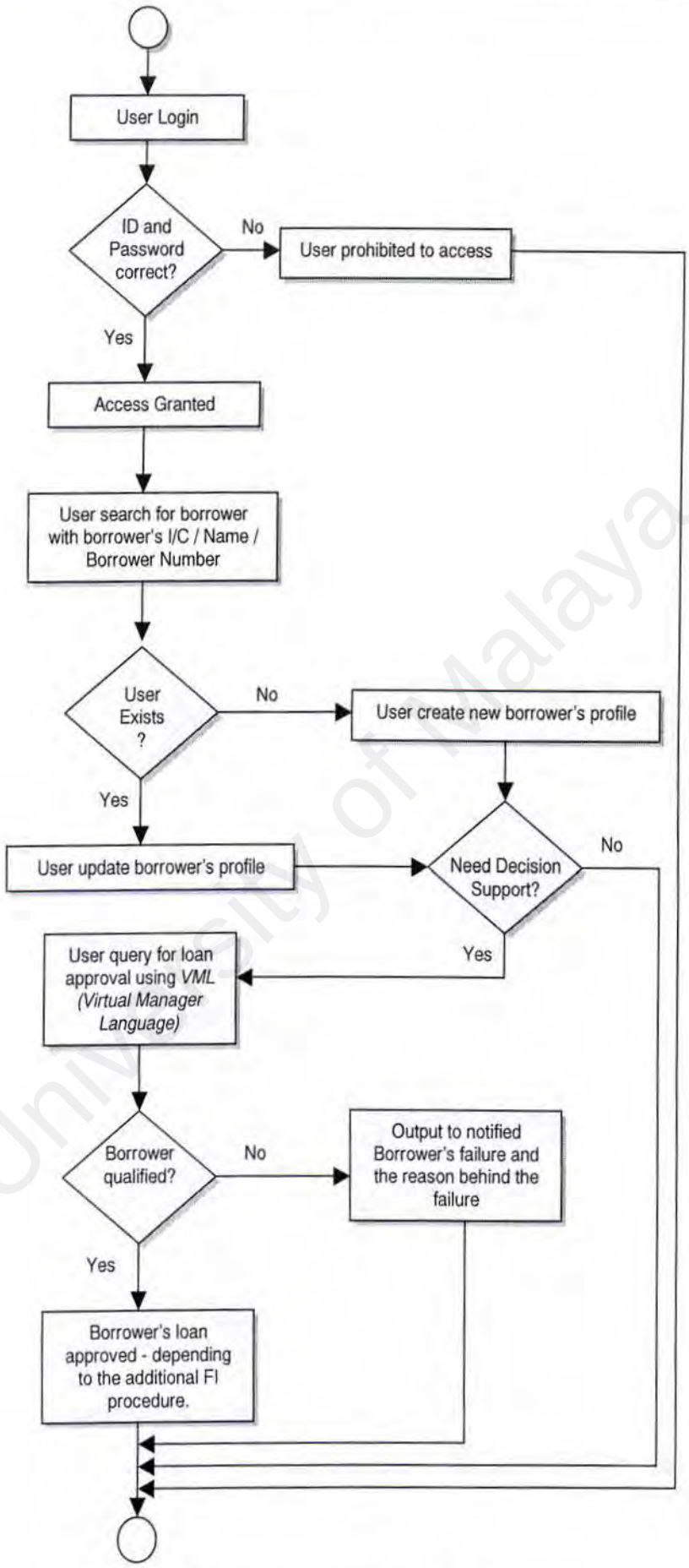


Figure 4.2 Flow Chart of the Procedures

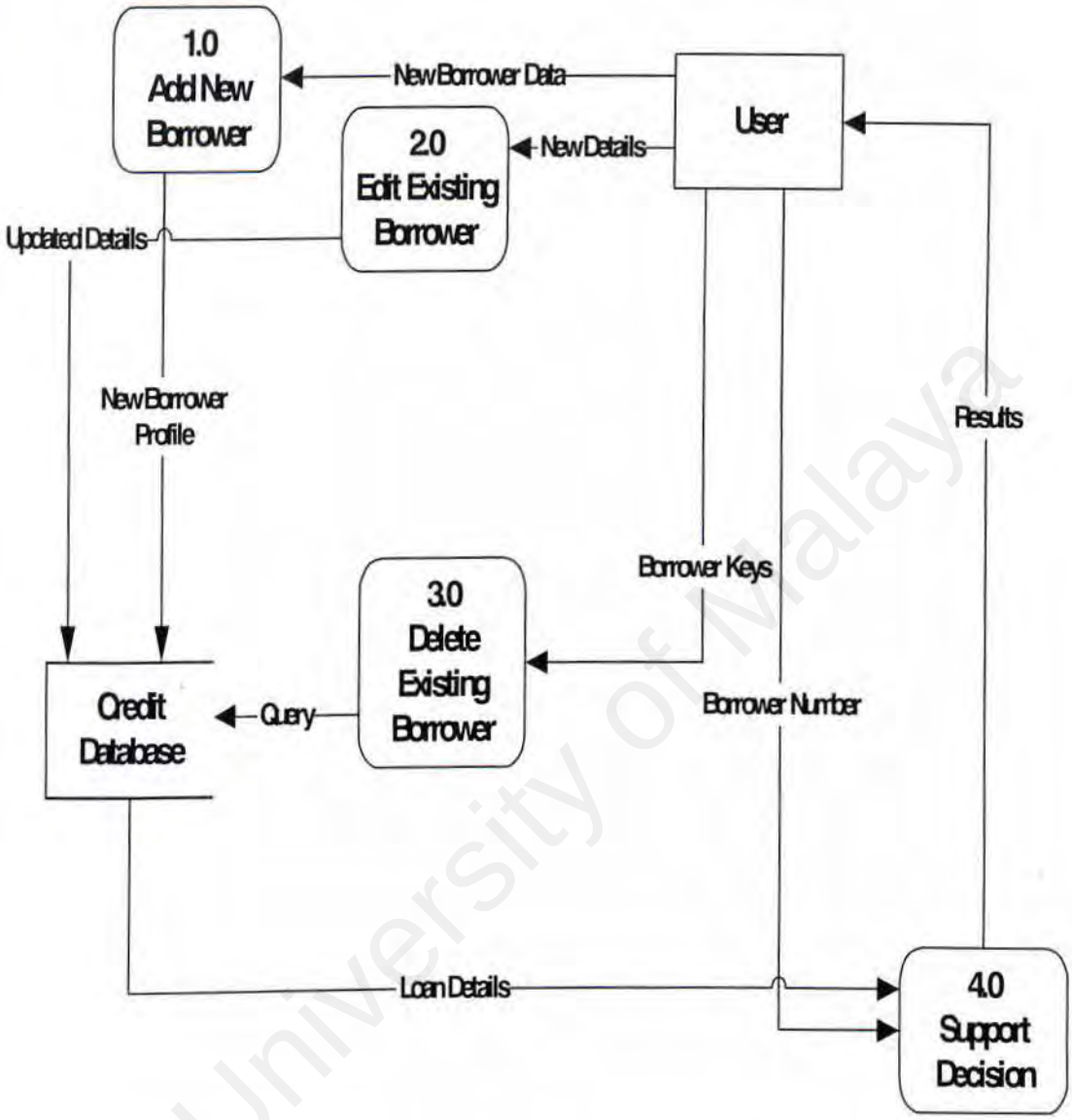


Figure 4.3.0 Highest Level of Data Flow Diagram

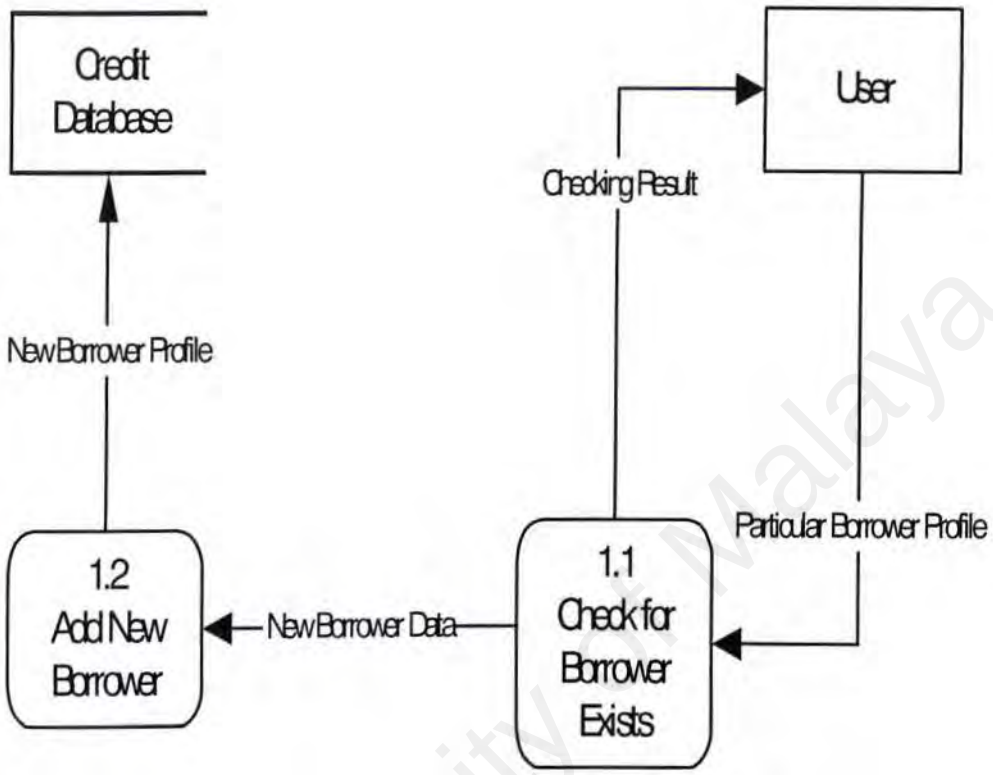


Figure 4.3.1 DFD – Details of 1.0

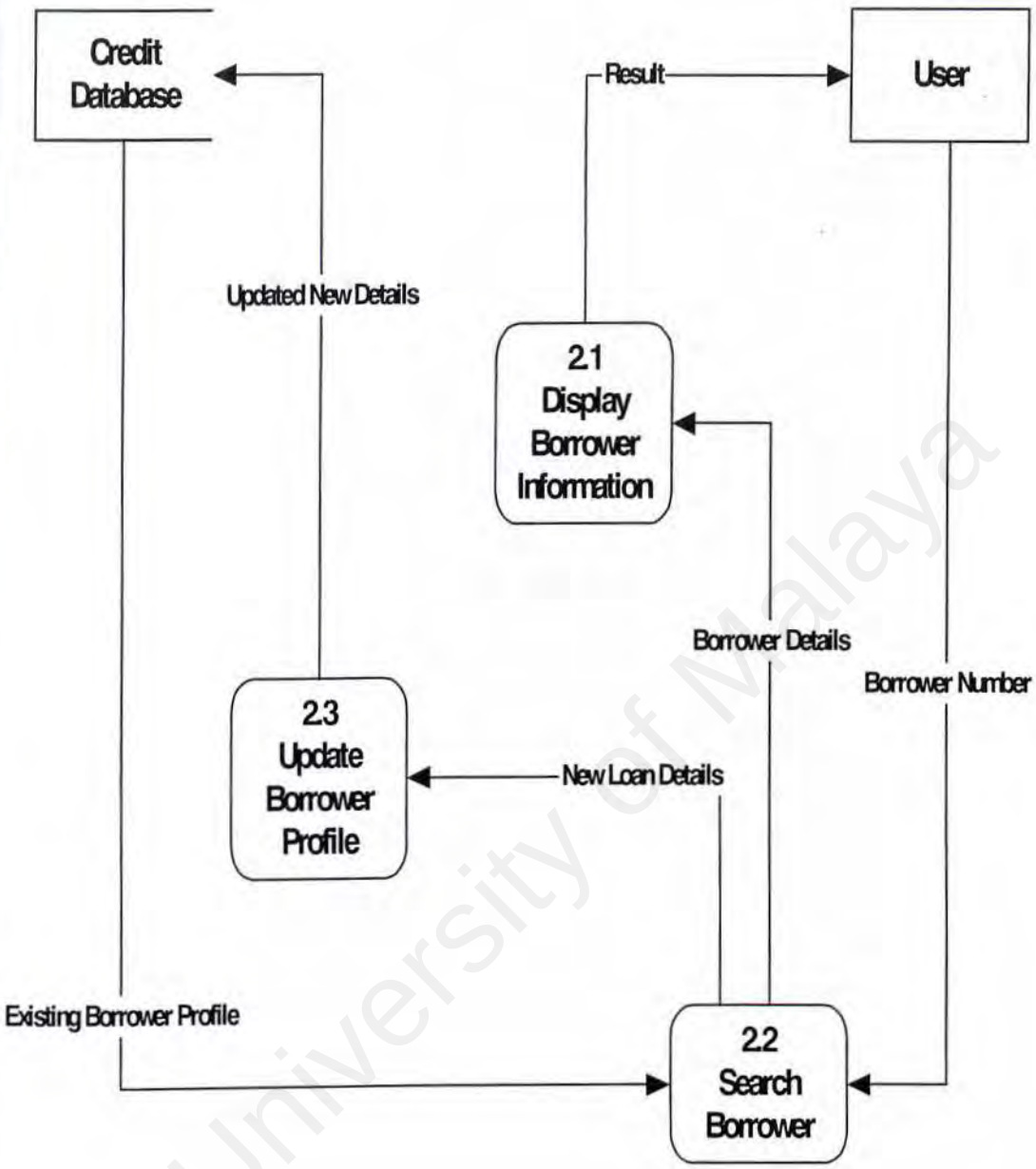


Figure 4.3.2 DFD – Details of 2.0

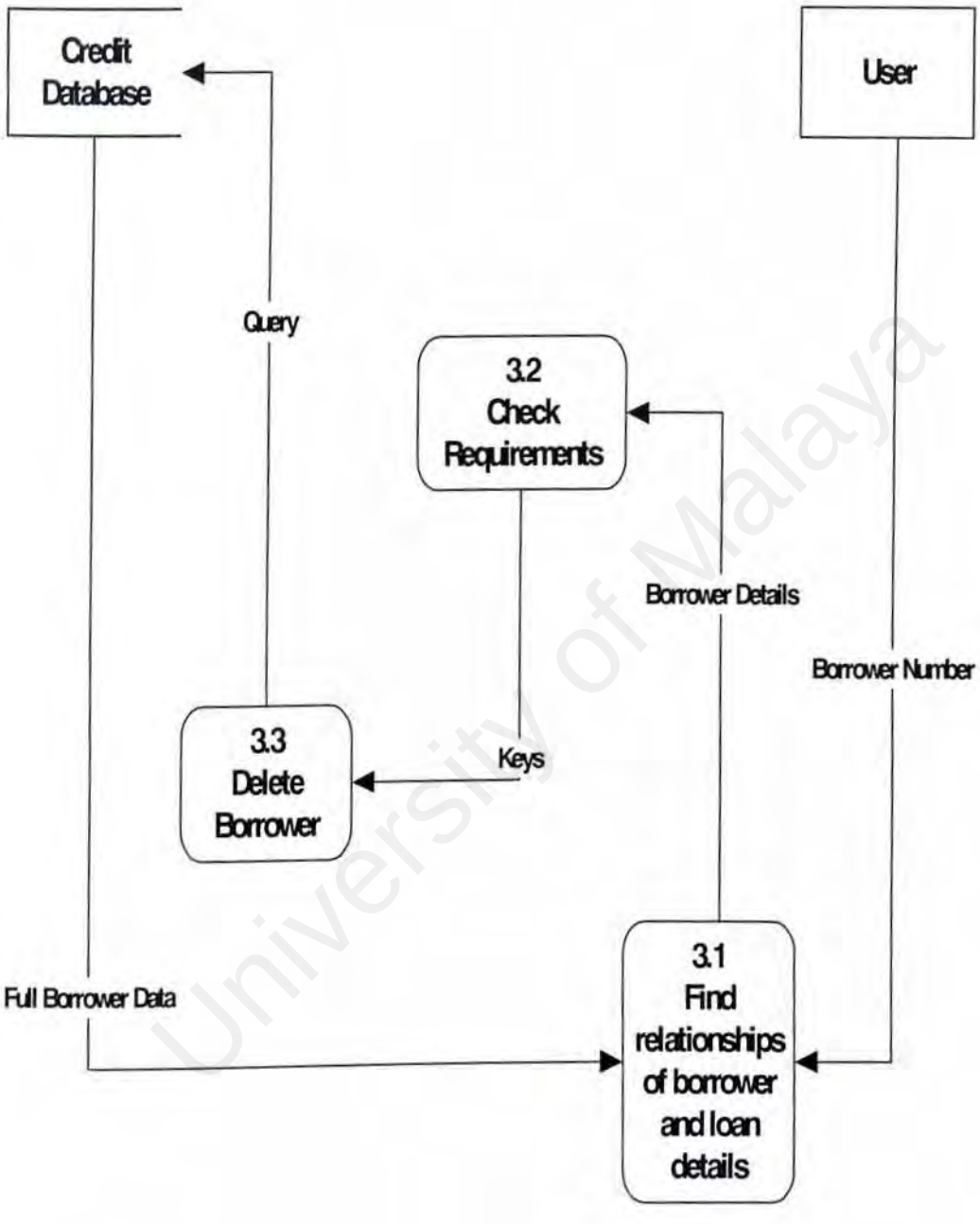


Figure 4.3.3 DFD – Details of 3.0

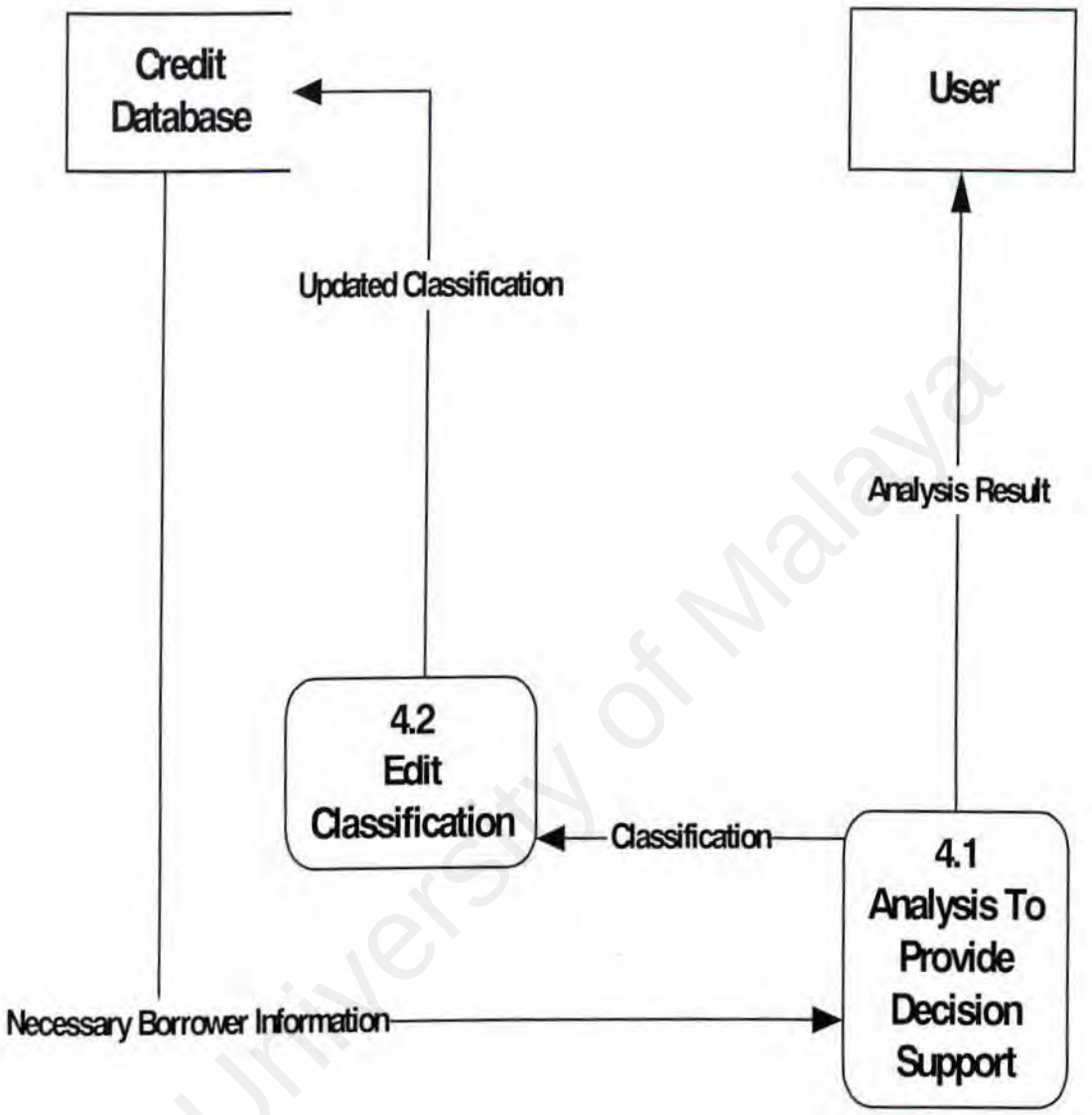


Figure 4.3.4 DFD – Details of 4.0

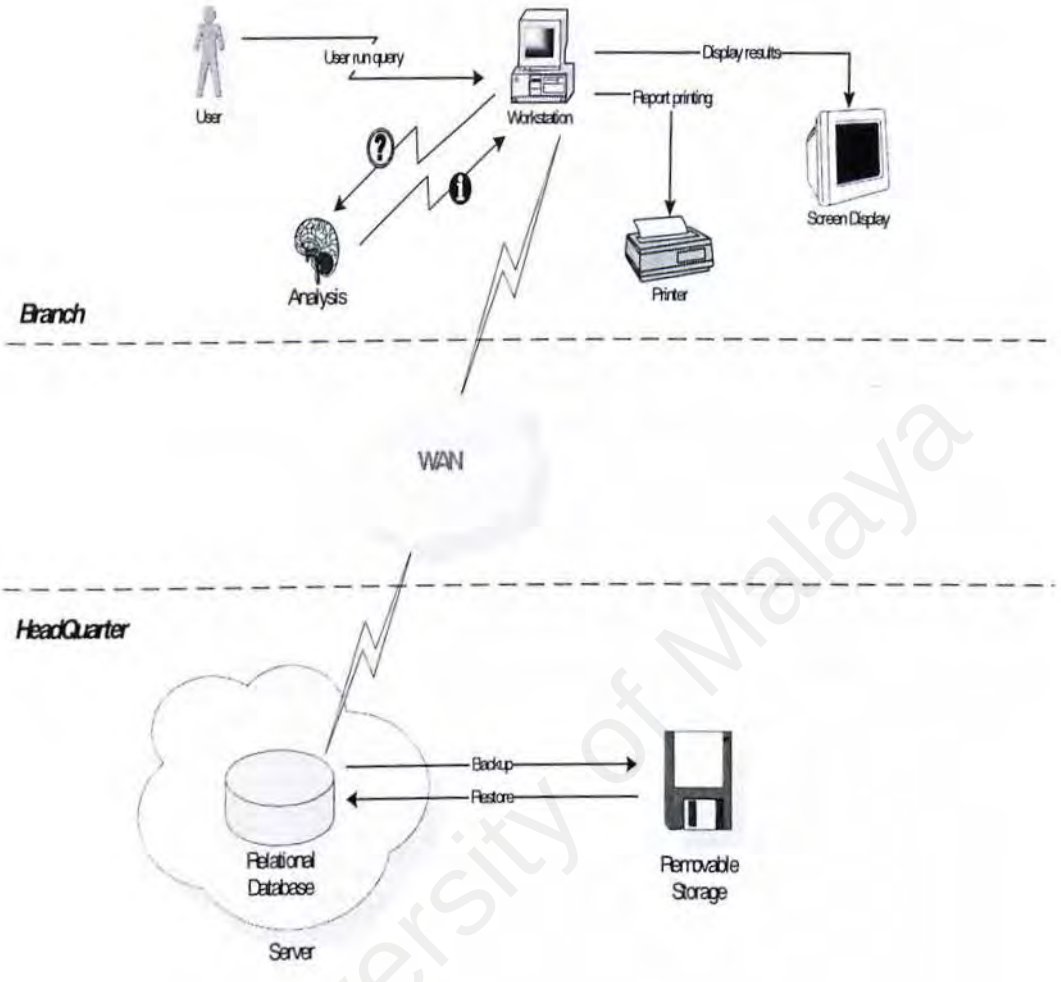


Figure 4.4 Operations of BCIS

4.3 Database Design

Database organization decisions have a great impact to the overall on performance, and flexibility. The purpose of this part is to prepare technical design specifications for a database that will be adaptable to future requirements and expansion.

Refer to *Figure 4.5* for more information on *BCIS* database design, which ensured good performance, data reliability, and data availability.

The database structure of the relationships in the *BCIS'* loans database is listed in the following sections. Refer to the tables *(Table 4.1 – Table 4.16)* for details about data type, and the descriptions.

Field	Data Type	Descriptions
fld_borrowerNo	Varchar	Primary key. "B" and 12 numbers beginning with the first borrower – B000000000001. Automatically provided by SYSTEM; increment by 1 for new record. Cannot edit by the USER
fld_IC	Varchar	Primary key for searching; ensure no duplicate entry of the same borrower
fld_name	Varchar	Enter by USER (as in IC); no short form permitted
fld_DOB	Char	Enter by USER (as in IC); format in DD/MM/YYYY
fld_perAddr1	Varchar	Enter by USER; permanent address for contact
fld_perAddr2	Varchar	Additional information about the permanent address (longer than 60 characters)
fld_curAddr1	Varchar	Enter by USER; current address for contact
fld_curAddr2	Varchar	Additional information about the current address (longer than 60 characters)
fld_contactNo	Varchar	Current contact number (handphone or fixedline)
fld_custDesc	Varchar	Description of the borrower; refer to *tbl_custDesc*
fld_dateOpen	Char	Date of the creation of the new borrower. Default is Today.
fld_dateClose	Char	Date of the deletion of the existing borrower. Default is the date of the deletion.
fld_status	Char	Status of the borrower; 'N' (New), 'A' (Amendment), 'D' (Deleted/Cancelled)
fld_curLoanNo	Varchar	Current Loan Number under the specific borrower. Used to record the total of loans.

Table 4.1 Database Table fld_borrowerProfile

Field	Data Type	Descriptions
fld_borrowerNo	Varchar	Primary key. Refer to the primary key at *tbl_borrowerProfile*. Cannot edit by the USER
fld_IC	Varchar	Primary key for searching; ensure no duplicate entry of the same owner/director
fld_name	Varchar	Enter by USER (as in IC); no short form permitted
fld_DOB	Char	Enter by USER (as in IC); format in DD/MM/YYYY
fld_perAddr1	Varchar	Enter by USER; permanent address for contact
fld_perAddr2	Varchar	Additional information about the permanent address (longer than 60 characters)
fld_curAddr1	Varchar	Enter by USER; current address for contact
fld_curAddr2	Varchar	Additional information about the current address (longer than 60 characters)
fld_shareHold	Char	Percentage of shares that hold by the specific owner/director (must be more than but not equal to 0%, and below but not equal to 100%)
fld_borrowerType	Char	Type of the borrower whether "DIRECTOR" or "OWNER"
fld_status	Char	Status of the owner/director; 'N' (New), 'A' (Amendment), 'D' (Deleted/Cancelled)

Table 4.2 Database Table tbl_ownerDirector

Field	Data Type	Descriptions
fld_borrowerNo	Varchar	Primary key. Refer to the primary key at *tbl_borrowerProfile*. Cannot edit by the USER
fld_loanNo	Varchar	Primary key. Automatically increments 1 for new record by system; "L" following by two numbers of customer description code (taken from tbl_borrowerProfile) and then following by another 5 numbers to indicate the total of loans under the same borrower number
fld_branch	Char	Provided by ADMIN on the first time of the activation of the system. Set once only.
fld_apprvLimit	Char	Enter by USER or provided by VLM (Virtual Loan Manager) after the USER run the VLM sub-system
fld_dateOpen	Char	Creation date of the new loan. Default is Today.
fld_dateClose	Char	Closing date of the existing loan. Default is Today.
fld_lending	Char	Status of the loan whether "Clean" or "Secure"
fld_secValue	Char	Reliazable value of the securities attached. Must be in Ringgit Malaysia.
fld_classification	Char	Classification of the loan (refer to the standard set by Bank Negara)
fld_repaidTerm	Char	Term agreed between the FI and borrower for the repayment back of the loan
fld_originalTenure	Char	Original Tenure agreed between the FI and borrower.
fld_legalActionStatus	Char	Status of the loan whether the loan under any legal action; e.g. Order of Frozen by High Court
fld_loanStatus	Char	Status of the loan whether the loan been rescheduled.
fld_shareSec	Char	Status of the loan whether the loan has any securities attached
fld_loanSectoral	Char	Sector of investment using the loan

Table 4.3 Database Table tbl_loanProfile

Field	Data Type	Descriptions
fld_borrowerNo	Varchar	Primary key. Refer to the primary key at *tbl_borrowerProfile*. Cannot edit by the USER
fld_loanNo	Varchar	Primary key. Automatically increments 1 for new record by system; "L" following by two numbers of customer description code (taken from *tbl_borrowerProfile*) and then following by another 5 numbers to indicate the total of loans under the same borrower number
fld_branch	Char	Provided by ADMIN on the first time of the activation of the system. Set once only.
fld_principal	Char	Original Approved Principal is the basic principal, which borrowed by the Borrower. Value in RM.
fld_interest	Char	Last Annual Interest is the interest rate in percentage fixed by the FI for that year of loan. Value in RM.
fld_totalOS	Char	Total Outstanding is the total of the money borrowed from the FI by adding Last Annual Interest to Original Approved Principal. Value in RM.
fld_totalPaid	Char	Total Paid Until Now is the money that had been paid by the Borrower to the FI from the date of the approval of the loan until Today. Value in RM.
fld_lastPayment	Char	Date Borrower does the last payment. Default date is Today.
fld_recentPayment	Char	Total of recent payment done by the Borrower in order to paid the loan, depends on the Repayment Term.

Table 4.4 Database Table tbl_currency

Field	Data Type	Descriptions
fld_borrowerNo	Varchar	Primary key. Refer to the primary key at *tbl_borrowerProfile*. Cannot edit by the USER
fld_loanNo	Varchar	Primary key. Automatically increments 1 for new record by system; "L" following by two numbers of customer description code (taken from tbl_borrowerProfile) and then following by another 5 numbers to indicate the total of loans under the same borrower number
fld_branch	Char	Provided by ADMIN on the first time of the activation of the system. Set once only.
fld_landedProperty	Char	"Y" or "N". Refer to the Land or landed property
fld_letterOfGuarantee	Char	"Y" or "N". Refer to the Letter of guarantee
fld_letterOfCredit	Char	"Y" or "N". Refer to the Letter of credit
fld_stocksAndShares	Char	"Y" or "N". Refer to the Stocks and shares
fld_cashAndFD	Char	"Y" or "N". Refer to the Cash and fixed deposit receipt
fld_debentures	Char	"Y" or "N". Refer to the Debentures
fld_inventoryMachinery	Char	"Y" or "N". Refer to the Inventory, Machinery, Equipment and other assets
fld_contractualRights	Char	"Y" or "N". Refer to the Contractual Rights, Contractual Moneys and Book Debts
fld_insurancePolicy	Char	"Y" or "N". Refer to the Insurance policy
fld_carriages	Char	"Y" or "N". Refer to the Ships, Aircraft and other carriages
fld_miscellaneous	Char	"Y" or "N". Refer to the Miscellaneous

Table 4.5 Database Table tbl_secAll

Field	Data Type	Descriptions
fld_borrowerNo	Varchar	Primary key. Refer to the primary key at *tbl_borrowerProfile*. Cannot edit by the USER
fld_loanNo	Varchar	Primary key. Automatically increments 1 for new record by system; "L" following by two numbers of customer description code (taken from tbl_borrowerProfile) and then following by another 5 numbers to indicate the total of loans under the same borrower number
fld_branch	Char	Provided by ADMIN on the first time of the activation of the system. Set once only.
fld_secCode	Char	Code indicates the type of land or landed property pledged by the borrower
fld_marketValue	Char	The current market value of the property in Malaysian Ringgit
fld_forcedSaleValue	Char	Refers to the forced sale value of the property in Malaysian Ringgit
fld_nameValuer	Varchar	The full name of the professional appraisers or the person making the valuation on the property
fld_dateValuation	Char	Date of which the valuation is made
fld_reservedValue	Char	Reserved value of the property in Malaysia Ringgit
fld_desc	Varchar	Information of where the land is situated and acres

Table 4.6 Database Table tbl_landedProperty

Field	Data Type	Descriptions
fld_borrowerNo	Varchar	Primary key. Refer to the primary key at *tbl_borrowerProfile*. Cannot edit by the USER
fld_loanNo	Varchar	Primary key. Automatically increments 1 for new record by system; "L" following by two numbers of customer description code (taken from *tbl_borrowerProfile*) and then following by another 5 numbers to indicate the total of loans under the same borrower number
fld_branch	Char	Provided by ADMIN on the first time of the activation of the system. Set once only.
fld_secCode	Char	The type of guaranteed letter pledged by the borrower.
fld_amtGuarantee	Char	The total amount guaranteed by the guarantor in Malaysian Ringgit for the borrower.
fld_nameGuarantee	Char	Name of the person guaranteeing the borrower, as appeared on the letter.
fld_IC	Char	The guarantor's Identification Card (IC) and Passport (PP) No. for an individual or Business Registration No. for Company.

Table 4.7 Database Table tbl_letterOfGuarantee

Field	Data Type	Descriptions
fld_borrowerNo	Varchar	Primary key. Refer to the primary key at *tbl_borrowerProfile*. Cannot edit by the USER
fld_loanNo	Varchar	Primary key. Automatically increments 1 for new record by system; "L" following by two numbers of customer description code (taken from *tbl_borrowerProfile*) and then following by another 5 numbers to indicate the total of loans under the same borrower number
fld_branch	Char	Provided by ADMIN on the first time of the activation of the system. Set once only.
fld_secCode	Char	The type of credit letter pledged by the Borrower.
fld_amount	Char	The value of the letter of credit in Malaysian Ringgit
fld_dateExpired	Char	Date of which the letter of credit is terminated.
fld_nameIssuer	Varchar	Name of the person issuing the letter of credit.

Table 4.8 Database Table tbl_letterOfCredit

Field	Data Type	Descriptions
fld_borrowerNo	Varchar	Primary key. Refer to the primary key at *tbl_borrowerProfile*. Cannot edit by the USER
fld_loanNo	Varchar	Primary key. Automatically increments 1 for new record by system; "L" following by two numbers of customer description code (taken from *tbl_borrowerProfile*) and then following by another 5 numbers to indicate the total of loans under the same borrower number
fld_branch	Char	Provided by ADMIN on the first time of the activation of the system. Set once only.
fld_secCode	Char	The type of stocks and shares pledged by the Borrower.
fld_marketValue	Char	The total market value of quoted shares to the nearest Malaysian Ringgit or net tangible asset backing per share of unquoted shares.
fld_dateValuation	Char	Date of which the valuation is made.

Table 4.9 Database Table tbl_stocksAndShares

Field	Data Type	Descriptions
fld_borrowerNo	Varchar	Primary key. Refer to the primary key at *tbl_borrowerProfile*. Cannot edit by the USER
fld_loanNo	Varchar	Primary key. Automatically increments 1 for new record by system; "L" following by two numbers of customer description code (taken from *tbl_borrowerProfile*) and then following by another 5 numbers to indicate the total of loans under the same borrower number
fld_branch	Char	Provided by ADMIN on the first time of the activation of the system. Set once only.
fld_secCode	Char	The type of cash and fixed deposit receipt pledged by the Borrower.
fld_amount	Char	The total amount of security pledged by the borrower to the nearest Malaysian Ringgit.

Table 4.10 Database Table tbl_cashAndFD

Field	Data Type	Descriptions
fld_borrowerNo	Varchar	Primary key. Refer to the primary key at *tbl_borrowerProfile*. Cannot edit by the USER
fld_loanNo	Varchar	Primary key. Automatically increments 1 for new record by system; "L" following by two numbers of customer description code (taken from *tbl_borrowerProfile*) and then following by another 5 numbers to indicate the total of loans under the same borrower number
fld_branch	Char	Provided by ADMIN on the first time of the activation of the system. Set once only.
fld_secCode	Char	The type of debentures pledged by the Borrower.
fld_amount	Char	The value of the debentures in Malaysian Ringgit.
fld_dateValuation	Char	Date of which the valuation is made.

Table 4.11 Database Table tbl_debentures

Field	Data Type	Descriptions
fld_borrowerNo	Varchar	Primary key. Refer to the primary key at *tbl_borrowerProfile*. Cannot edit by the USER
fld_loanNo	Varchar	Primary key. Automatically increments 1 for new record by system; "L" following by two numbers of customer description code (taken from *tbl_borrowerProfile*) and then following by another 5 numbers to indicate the total of loans under the same borrower number
fld_branch	Char	Provided by ADMIN on the first time of the activation of the system. Set once only.
fld_secCode	Char	The type of inventory, machinery, equipment or other assets pledged by the Borrower.
fld_amount	Char	The value of the inventory, machinery and equipment in the nearest Malaysian Ringgit.
fld_basis	Char	Refers to the basis of valuation such as current market value, depreciated value, auction value, tender price or ohers.

Table 4.12 Database Table tbl_inventoryMachinery

Field	Data Type	Descriptions
fld_borrowerNo	Varchar	Primary key. Refer to the primary key at *tbl_borrowerProfile*. Cannot edit by the USER
fld_loanNo	Varchar	Primary key. Automatically increments 1 for new record by system; "L" following by two numbers of customer description code (taken from *tbl_borrowerProfile*) and then following by another 5 numbers to indicate the total of loans under the same borrower number
fld_branch	Char	Provided by ADMIN on the first time of the activation of the system. Set once only.
fld_secCode	Char	The type of contractual rights, contractual money and book debts pledged by the borrower.
fld_amount	Char	Refers to the assignment value pledged to the nearest Malaysian Ringgit.

Table 4.13 Database Table tbl_contractualRights

Field	Data Type	Descriptions
fld_borrowerNo	Varchar	Primary key. Refer to the primary key at *tbl_borrowerProfile*. Cannot edit by the USER
fld_loanNo	Varchar	Primary key. Automatically increments 1 for new record by system; "L" following by two numbers of customer description code (taken from *tbl_borrowerProfile*) and then following by another 5 numbers to indicate the total of loans under the same borrower number
fld_branch	Char	Provided by ADMIN on the first time of the activation of the system. Set once only.
fld_secCode	Char	The type of insurance policy pledged by the Borrower.
fld_amount	Char	Refers to the cash surrender value for the insurance policy in the nearest Malaysian Ringgit.

Table 4.14 Database Table tbl_insurancePolicy

Field	Data Type	Descriptions
fld_borrowerNo	Varchar	Primary key. Refer to the primary key at *tbl_borrowerProfile*. Cannot edit by the USER
fld_loanNo	Varchar	Primary key. Automatically increments 1 for new record by system; "L" following by two numbers of customer description code (taken from *tbl_borrowerProfile*) and then following by another 5 numbers to indicate the total of loans under the same borrower number
fld_branch	Char	Provided by ADMIN on the first time of the activation of the system. Set once only.
fld_secCode	Char	The type of ships, aircraft or other carriages pledged by the Borrower.
fld_amount	Char	Refers to the value of the security pledged by the Borrower.
fld_basic	Char	Refers to the basis of valuation such as current market value, depreciated value, auction value, tender price or others.

Table 4.15 Database Table tbl_carriages

Field	Data Type	Descriptions
fld_borrowerNo	Varchar	Primary key. Refer to the primary key at *tbl_borrowerProfile*. Cannot edit by the USER
fld_loanNo	Varchar	Primary key. Automatically increments 1 for new record by system; "L" following by two numbers of customer description code (taken from *tbl_borrowerProfile*) and then following by another 5 numbers to indicate the total of loans under the same borrower number
fld_branch	Char	Provided by ADMIN on the first time of the activation of the system. Set once only.
fld_secCode	Char	The type of contractual rights, contractual money and book debts pledged by the Borrower.
fld_amount	Char	The total amount of the security pledged by the Borrower.

Table 4.16 Database Table tbl_miscellaneous

4.4 User Interface Design

User interface design is very important to offer a user-friendly, reliability, intuitive, minimize the need for users to memorize the process and events, and at the same time give a good impression to the users. Below are the few screenshots of user interface of *BCIS* system.

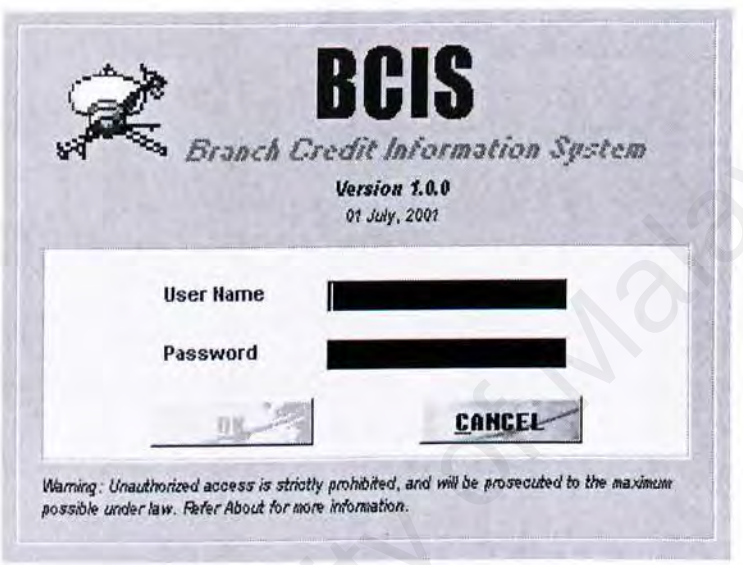


Figure 4.5 Login Screen

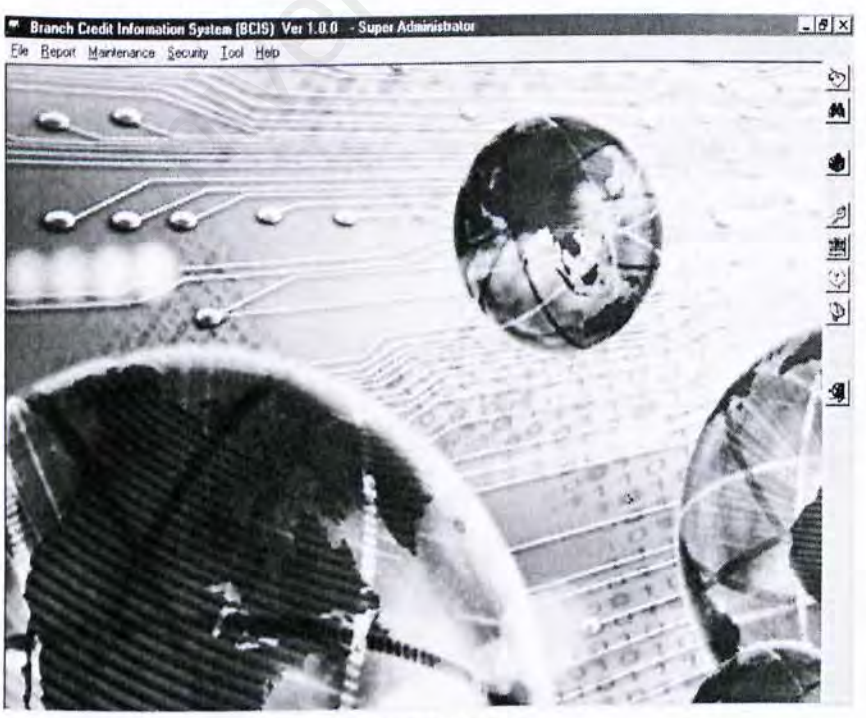


Figure 4.6 Main Menu

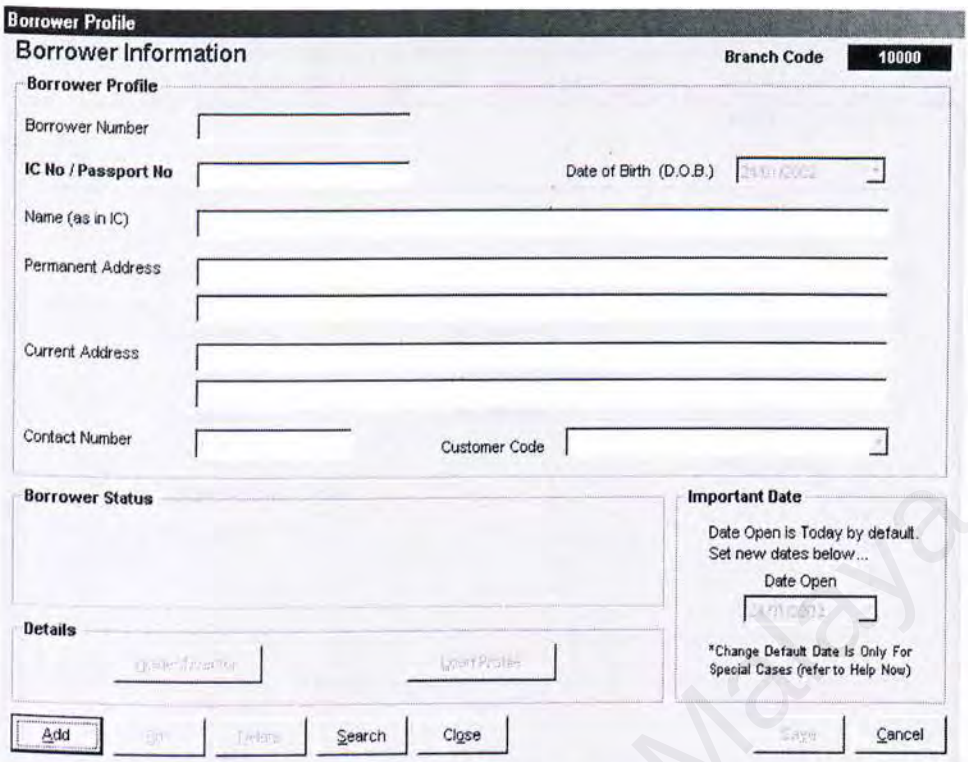


Figure 4.7 File -> Data Entry

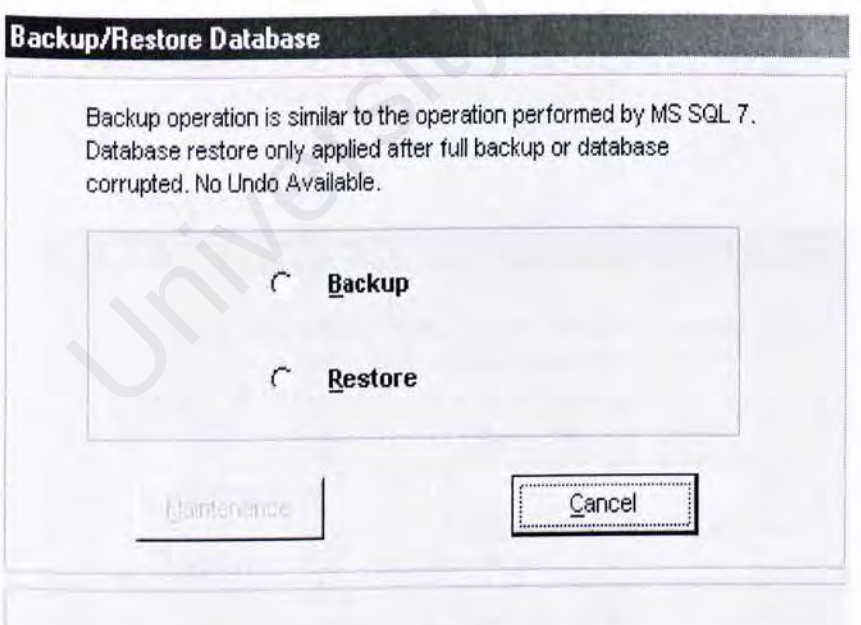


Figure 4.8 Maintenance -> Backup/Restore

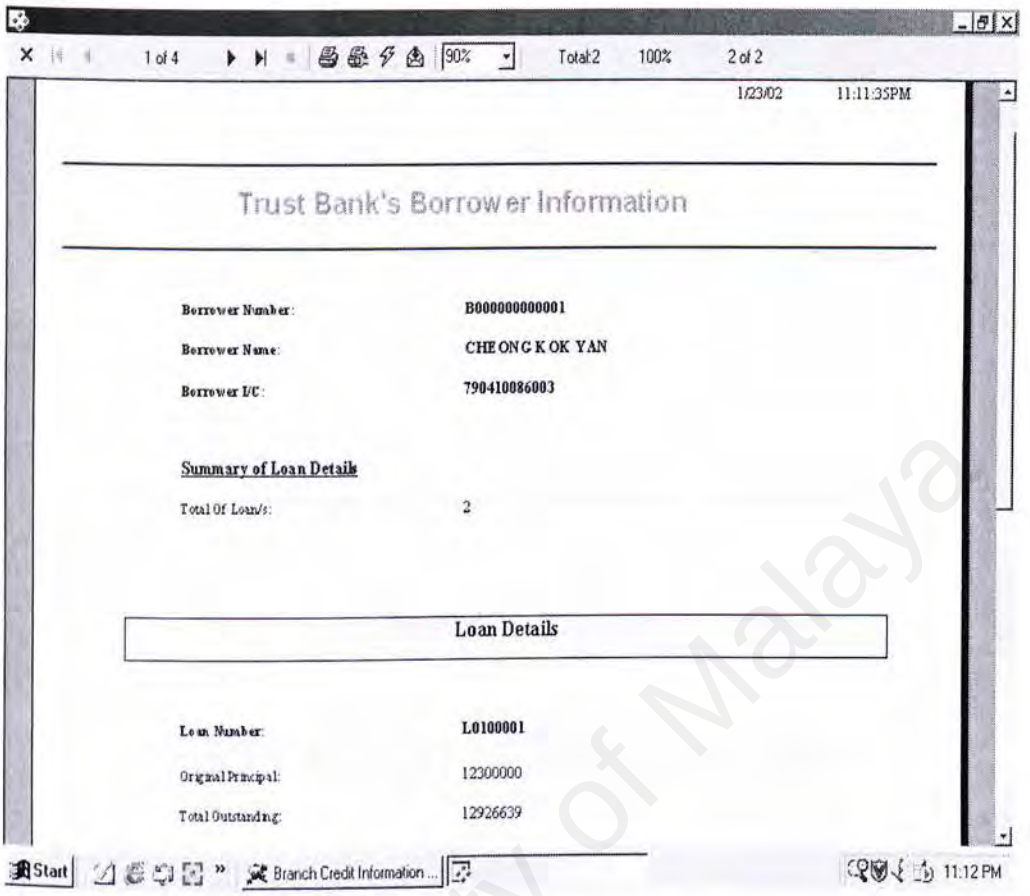


Figure 4.9 Report -> Report Manager

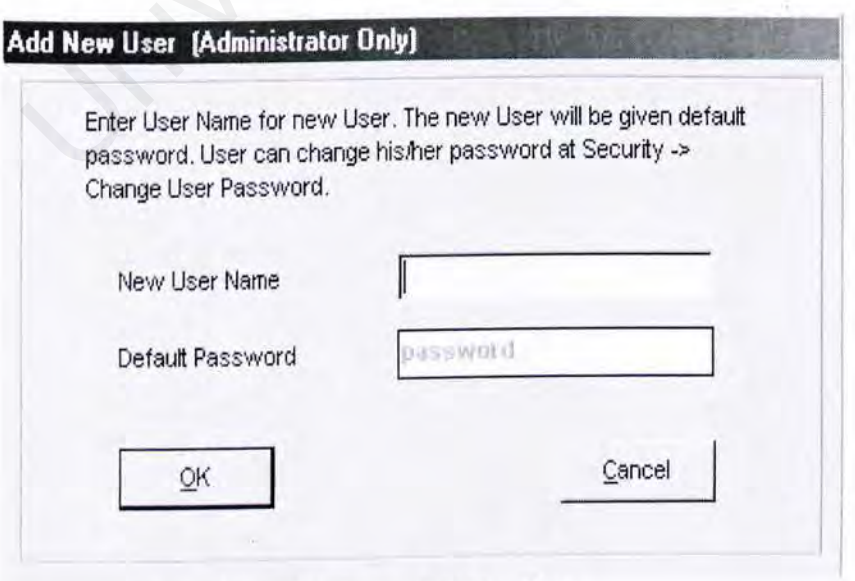


Figure 4.10 Security -> Add New User

CHAPTER 5

SYSTEM IMPLEMENTATION

5.0 System Implementation

Implementation is the stage of translates and implements the details of the earlier design representing the system into programming realization.

5.1 Development Environment

Development environment has a certain impact on the development of a system. The choice of hardware and software combinations could make a clear different and also may be leave impact in the project schedule and the success of the project. The hardware and software tools used for developing and running with full performance are as below.

5.1.1 Hardware Used

- 500 MHz processor;
- 128MB RAM memory;
- 5 GB of hard disk space
- Network Interface Card (NIC);
- Network cable connection (LAN or WAN) with recommended bandwidth of 10Mbps or more; and
- Other support peripheral devices such as Uninterruptible Power Supply (UPS).

5.1.2 Software Used

- Operating system platform – Windows 2000;
- Database Management System – Microsoft SQL Server 7.0;
- Main Programming Language – Visual Basic 6.0;
- Third Party Support Software – Seagate Crystal Report 7.0;
- Programming Tool – Macromedia Dreamweaver 4.0, Adobe Photoshop 6.0; and
- Antivirus software

5.2 System Development

The development of the *Branch Credit Information System* is done using Microsoft Visual Basic 6.0.

5.2.1 System Coding

During the coding process of the system, good programming practices are including in all over the project. One of the examples is turning on the *Option Explicit* option in the Declaration section. With this feature turning on, Visual Basic will helps to identify any undefined variables by prompting out an error message. Undeclared local variables are as slow as twice the speed of declared local variable. It is clear that turning on this option will definitely enhance a lot the overall speed of the system.

Besides, meaningful and consistent variable names help the next programmer who wants to enhance, or maintaining the system in the future. It is bad programming practices to name the variables with the name that only can understand by the programmer himself, like naming the variables with the name of his pet or wife. This action will definitely create headache to other programmer that need to make changes to the system in the future. Example of the good declaration of variable name:

```
Dim TotalBorrower as Integer
Dim BorrowerName as String
Dim BorrowerExist as Boolean
```

A programmer also has to make sure the functions, subs, declarations, and modules simple and readability. For example, no more than one statement should appear in a same line, and separate methods should be separated by blank lines. Besides, the nested control structures must be wrote in proper manner to lessen mistakes and more readability, as shown below:

```
For countTotalBorrower  = 0 to 10
        For countTotalLoan = 1 to 10
                If BorrowerExist = True then
                        <statement>
                End if
        Next
Next
```

Updating to the database need to be handled with care to preserve data consistency in the database. Any errors when updating the database have

possibility corrupting the data, or lost updates. Here, transaction is used to avoid any lost updates and links. For example:

```
Dim con as rdoConnection
.....
con.BeginTrans
con.Execute "INSERT INTO tbl_borrowerProfile VALUES (" & _
txtBorrowerName.Text  & "," & _
txtBorrowerAddress.Text  & "," .....

If con.Errors = 0 then
        con.CommitTrans
Else
        con.RollBackTrans
End if
```

5.2.2 Debugging

During the coding of the system, there may be possibility that the system will create unexpected results, for example counting incorrect total, and load forms without following the sequences.

One the good debugging methods is using *Debug.Print*. This method of debugging use the *Immediate* window to track a running procedure by printing messages or expression values to it while the code is running. Placing the *Debug.Print* statements at strategic points in the code can greatly aid in debugging problems. Some suggested strategic points to place the *Debug.Print* statements are at the head or foot of a procedure, while iterating objects or values

in a collection or loop, after a complex calculation, after accepting user input, and before deleting files or objects.

Another debugging method is using a breakpoint. Breakpoint causes the executing code to stop at a particular line but still keep the system state in memory. This make all the variables in the current scope are available for inspection in the *Immediate* window. By using *Step Into* and *Step Over* functionality to move through the code statement-by-statement enabled the code execute in slow motion.

As an alternative to setting breakpoints, *MsgBox* function used to indicate the program's state. With this strategy, any part of the program can be monitored, and *MsgBox* function can be called whenever it needed to return the information.

CHAPTER 6

TESTING

6.0 Testing

Testing is a critical step in assuring the quality of the developed system and is representing the overall review of specification, design, and coding. Testing is performed to ensure that the programs are executed correctly.

6.1 Testing Techniques

There are few testing techniques applied in the testing stage of the system. The techniques used are white box testing, and black box testing.

6.1.1 White box testing

White box testing is a test that involved directly with the structure of the code within a module or code segment. The test should adequately cover all the code in a given module, but this is hard to accomplish. Code coverage is define in terms of seven types:

- Segment coverage;
- Branch coverage or node testing;
- Compound condition coverage;
- Basis path testing;
- Data flow testing;
- Path testing; and
- Loop testing

6.1.2 Black Box Testing

In addition to white box testing there is also black box testing. Black box testing assuming that the logic structure of the code is unknown. This is the point at which the function of a module is tested. There are several tests can made at this point:

- Error guessing;
- Equivalent class partitioning;
- Boundary value analysis;
- Cause-and-Effect graphing;
- Domain testing;
- Module interface testing; and
- Command-line testing

6.2 Testing Strategies

Testing a large system usually involving several stages. The stages are unit testing, integration testing, and system testing.

6.2.1 Unit Testing

Unit testing is a set of basic tests at the module level. Each module is treated as a independent component and is tested individually to ensure the component operate well and in an expected operations. The unit testing involving:

- Testing the interfaces to ensure information flows properly into and out of each program unit;
- Testing boundary conditions to ensures each component is operating correctly at the boundary values;
- Ensure all independent paths in a control structure are tested at least once; and
- Testing all error handling paths

6.2.2 Integration Testing

Integration testing is one of the frustrating parts of software development and testing. Purpose of integration testing is to adequately test whether or not the system actually runs as one program as expected. Incremental integration testing approach is used in testing the system. By using this approach, the program is

constructed and tested in small segments where error is easier to isolate and corrected.

6.2.3 System Testing

System testing is actually a series of different test designed to fully exercise the system to uncover its limitations and measure its capabilities. The objective is to test an integrated system and verify that it meets specified requirements.

CHAPTER 7

SYSTEM EVALUATION

7.0 System Evaluation

This chapter will review the strengths and limitations of the system as well as the problems encountered and their solutions.

7.1 Problems Encountered and Solutions

There are many problems encountered at the beginning, during, and as well as after the delivery of the system.

7.1.1 Difficulties In Determining The Appropriate Development Tools

There are many development tools that are available in the market, especially the fourth programming language, like Visual Basic, Java, and C++. Choosing a suitable tool is a critical process as all tools available have their strengths and weaknesses.

Besides, the availability of a technology, hardware and supporting software to support, its learning curve, compatibility with the existence operating system and technologies. A great deal of reading and research from many sources, like books and Internet regarding the problems helped to solve the problem, and choose the suitable tools.

7.1.2 Difficulties In Determining The Scope Of The Project

Due to the interdependence nature of credit systems and inexperience in developing the similar systems related to the banking fields, made outlining the scope of the project difficult. The inability to distinguish which of the system features are essential, or optional has made defining the project scope complicated. In order to define the project scope, discussions with project supervisor and advices from my training's ex-supervisor, Mr.Thong.

7.1.3 Difficulties In Information Gathering

Gathering information is one of the most important phases of the project. Problems are arising when dealing with the business requirements, as most of the users and bank managers reluctant to disclose the information because the information required is considered as *Private and Confidential*. This has made useful data gathering techniques such as interviewing and observation impossible.

Therefore, in order to gather the needed information, other method of acquisition information required, such as informal interview with the users, research papers, Internet, and books.

7.1.4 Difficulties In Defining System Requirements

The difficulties in finding information have led to difficulties in defining the requirements for the system. Unlike some other systems where either the domain experts or requirements can be found easily from the basic sources, such as Internet, and plenty of real life workable systems available, such as online shopping. System requirements of the project are following the information gathered from the users from several banks with informal interviews.

7.1.5 Difficulties In Defining The Flow Logic Of The System

The system is only based on the information gathered from interviews with the users; as a result, the flow logic of the system is very hard to define. The system is only following the flow logic based on my understanding of the business requirements and the important of ease of use.

7.2 System Strengths

7.2.1 Targeted for Popular Windows Platform

Branch Credit Information System is targeted on the Microsoft Windows platform, which is the most popular and common operating system in the business environment today. Most of the computers in the banks are installed with Windows platform, therefore no incompatibility problems arisen.

7.2.2 Graphical User Interface (GUI)

The design of the interface of the system is based on graphical user interface (GUI). It is designed to be as user-friendly as the system can with the available technologies in market. Therefore, the system is relatively easy to learn and use.

7.2.3 Consistent User Interface

The user interfaces maintains its consistency throughout the system. The choice of icons, text formatting and color are standardized across all sections.

7.2.4 Search Existing Borrower

The system will automatically check for existing borrower when the user input the borrower profile, such as Identification Card number. This feature will eliminate any duplicate entry and mistakes from the user.

7.2.5 Good Security Features

Different level of users is created for different level of permissions on the system. The user needs to have the right password and login name to grant access to the system. The users will remind automatically by the system to change the password every one week. Besides, all the passwords are encrypted with special algorithm to ensure system safe and block any unauthorized users to access the system.

7.2.6 Reliable And Ensure Data Integrity

The programming of the system using transaction features to enable the ability of rollback when the system encounters any errors that have possibility to violate data integrity. This is implemented to ensure no lost updates or lost links occurred in the database.

7.3 System Limitations

Although the system implemented to adapt many different environments, the system still has several limitations.

7.3.1 Not Platform Independent

This system is built with Microsoft Visual Basic 6.0 and using Microsoft SQL Server 7.0 as backend. As a result, this system needs to be install and run on Windows platform. Customization needs to be done if the user needs to change the database management system (DBMS) from MS SQL 7.0 to Oracle or other DBMS.

7.3.2 No Real Time Information

As this system is built with static features because the server does not post any real time information from when there are any modifications or updates to the data in the database. This limitation caused by the no real time support by

Microsoft Visual Basic, comparing to the unique feature owned by other web based tools, such as ASP.

7.3.3 Lack of Functional Modules

Currently, the functionality provided by the system is still very limited, and no fully support the fast growing of the business environment. Functions such as real *Decision Support System* is still under research and not included in the system, due to less information gathered, and absence of domain experts.

7.4 Future Enhancement

The system is still under continuous developing as this system is built under incremental model. New updates and upgrades will be integrated in the system from time to time.

7.4.1 Decision Support System

Decision Support System (*DSS*) will be integrated into the system as soon as possible because this feature is very useful and suitable to be one of the competitive weapons in the competitive market now. This feature will provide fast approval system for the use of the branch manager to approve the loan without any interrupt from headquarter. As a result, the loan approval procedures will definitely faster and more reliable.

7.4.2 Integration To Other Existing Financial System

The usefulness of an isolated or independent system is very limited. This is because the important and increasing of functionality of the system is depending of the level of interdependency. The output of the sub-systems will be automatically flow into the system without the interruption from human will definitely save costs in labor, and save a lot of time.

CHAPTER 8

CONCLUSION

8.0 Conclusion

Valuable skills, experiences, and knowledge have been gained in a great "amount" throughout the development of the system, including skills in programming languages, database management, networking, concurrency access, system analysis and design, and information gathering.

Overall, the project has achieved and fulfilled the objectives and requirements of the credit information system as determined during the system analysis phase. It is reliable and ease of use.

Finally, the project has "open" a path for individual who is creative and innovative to continue the never-end developing of the system to fulfill the future users need. The enhancing of the system will be more robust to errors, reliability, ease of use, and new features to speed-up the loan approval procedures.

REFERENCE

Ref. No	*References*
1	Bourke, P., & Shanmugam, B. (1990). An Introduction To Bank Lending. Australia: Addison-Wesley Publishing Company.
2	Choosing A Database Product. Access on July 11, 2001 from *http://databases.about.com/cs/index.html*
3	Consumer Loans. Access on July 11, 2001 from *http://www.bcb.com.my/menus/consumer_index.htm*
4	Consumer Loans. Access on July 11, 2001 from *http://www.maybank2u.com.my/maybank_group/products_services/consumer_loan*
5	Dennis, Marshall W. (1978). Fundamentals of Mortgage Lending. Virginia: Reston Publishing Company.
6	Deitel, H.M., & Deitel, P.J. (1999). Java. How To Program. New Jersey: Prentice Hall, Inc.
7	Deitel, H.M., & Deitel, P.J. (1999). Visual Basic 6. How To Program. New Jersey: Prentice Hall, Inc.

8 Hempel, G.H. & Simonson, D.G. (1999). Bank Management: Text And Cases. US: John Wiley & Sons, Inc.

9 Interview with Sivanasvaran, N. (Executive Officer) from Bumiputra-Commerce Bank Berhad (Section 14), on 02/07/2001 at 3.45pm.

10 Loan. Access on July 11, 2001 from *http://mortgagestart.com/mortgage_prof.html*

11 Marakus, George M. (1999). Decision Support System. New Jersey: Prentice Hall, Inc.

12 Masters, Gary, & Santoro, Denise (1999). Visual Basic 6 Complete. US: SYBEX.

13 Microsoft Access Database Fundamentals. Access on July 11, 2001 from *http://databases.about.com/cs/access/index.html*

14 Mortensen, Lanse & Sawtell, Rick (1999). MCSE: SQL Server 7 Administration Study Guide. US: SYBEX.

15 Mortgages. Access on July 11, 2001 from *http://www.bankrate.com/brm/rate*

16 Ong, Kok Bin (1993). Banking Securities In Malaysia: A Handbook for Lawyers, Bankers, Executives and Businessmen. Malaysia: Pelanduk Publications.

17 Opinions: Choosing Between Oracle and SQL Server: Would you rather be prefect or pragmatic. Access on July 11, 2001 from *http://www.infoworld.com/community/t_opinions.html*

18 Oracle, MSSQL, or Sybase. Access on July 11, 2001 from *http://databases.about.com/ab-databases/myforums/msg.html*

19 Pfleeger, S.L. (2001). Software Engineering: Theory And Practice. New Jersey: Prentice Hall.

20 Stallings, William & Slyke, R.V. (1998). Business Data Communications. New Jersey: Prentice Hall.

21 Thillainathan, R. (1985). Are Malaysian Bank Lending Rates Too High? Malaysia: ISIS.