CATTLE MANAGEMENT SYSTEM

UMAYA LATHA A/P RAVINDREN

A report submitted in partial fulfillment of the requirement
for the award of the
Degree of Bachelor of Computer Science
(Software Engineering)

Faculty of Computer Systems & Software Engineering
University Malaysia Pahang

APRIL 2010

ABSTRACT

Nowadays, the cattle farming are being a profitable sector in the growth of our nation. Therefore, many are getting involved in cattle farming business. Cattle Management System (CMS) is a system that fulfills the needs of a system for cattle farming. Cattle Management System (CMS) has functions such as edit cattle record, update the expanses of the cattle farm and update the immune table for the cattle. The prototype of CMS enables the users to manage the cattle farm. The methodology that will be used to develop CMS is Software Development Life Cycle that suits the web based system development. The scripting language for CMS is VB.Net, developed in Microsoft Visual Studio.Net 2008. The database for CMS was created by using Microsoft SQL Server 2005. This system is to be used by the administrator and staff, who is the farmer in the cattle farm.

ABSTRAK

Baru-baru, pertanian ternakan merupakan sektor yang menguntungkan dalam pertumbuhan ekonomi negara kita. Oleh kerana itu, ramai yang terlibat dalam perusahaan tani ternakan. Cattle Management System (CMS) adalah suatu sistem computer yang memenuhi keperluan pengendalian pertanian ternakan. Selanjutnya, sistem berkomputer untuk pengurusan usaha tani ternakan digunakan secara meluas di luar negeri. Cattle Management System (CMS) akan menyediakan fungsi seperti merekod data-data yang diperlukan untuk rujukan penternak. Metodologi yang digunakan untuk mengembangkan CMS adalah Software Development Life Cycle yang berpadanan dengan pembangunan sistem berasaskan Visual Basic.Net adalah bahasa skrip untuk sistem CMS dengan menggunakan perisian Microsoft Visual Studio.Net 2008. Microsoft SQL Server 2005 digunakan sebagai penyimpan data CMS. Sistem ini akan digunakan oleh pentadbir dan petani di ladang lembu.

TABLE OF CONTENTS

LIST OF TABLE

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Cattle are domesticated ungulates mammal animals. Cattle are Bovinae subfamily and originated from Bos genus. In many places, cattle is been raised well by the farmers for the beef and dairy products, which is profitable. Besides that, cattle which is known as livestock, also act as draft animals, it pulls carts and plows. Cattle as well kept for other products such as leather and dung for manure or fuel. This reveals the important of cattle business in today's life.

Cattle Management System (CMS) is a system for the cattle farmers that can ease the cattle management. CMS acts as record keeper for the farmer to supervise the cattle farmer. CMS also helps the farmer to observe the health level of the cattle. Besides that CMS can be refer as a guidelines for new cattle farmer to be acquainted with the cattle development and the management.

1.2 Problem Statement

Nowadays, many new farmers are not aware about the real way of running and caring a cattle farmer. Those people of olden days are educated by the antecedent about the cattle rising and development. This is also due to the tradition, where the cattle caring are done by a family and being thought to the young in order to keep it as the family business. Appropriate to the technology development many those people who did the cattle rising as their family business was not practicing it anymore. At this instant, the cattle farming are being very profitable sector in the growth of our nation. A perceptive system is desirable to support such situation. Thus, many are taking up this cattle farming as their business for this ages. However there are some who are inexperience in this field and they need lot of guidelines before getting involve in cattle farming.

Besides that, difficulty does occur if happens a situation where the farm caring has to be turned to other hand due to certain unpleased reason. Therefore a regimented system with aptitude to keep data about daily management of cattle farmer should be used by farmer. If the farm being handed over to a new farmer, this system will be a fine source for the new farmer to learn and know about the current flow of the management. Besides that the system also could play a role as an education for the new and rawness farmer to be exposed about the cattle management. The health and the caring of rising cattle also have to be considering in the system. A firm system for cattle farming will ease the management of cattle farms and it also enables the productivity of cattle farming to increase.

1.3 Objective

The objectives of this project are to:

i. Develop a prototype web-based system for cattle management.
ii. Develop a data management system for cattle farm.
iii. Provide basic information about cattle management.
iv. To apply 10 Heuristics in designing interface by Jacob Nielson.

1.4 Scope

The scopes of this project are:

i. The system is developed as guides for the new farmers in Malaysia.

ii. The users of this system are specifically cattle farmers, who take cares of cattle for its beef and dairy; and for those who are interested to know about the running of cattle production.

iii. The system is developed to provide the channel for running the cattle farm.

iv. This system will be developed using Visual Studio.NET.

1.5 Organization of Thesis

This thesis consists of six chapters. Chapter 1 is the introduction of the thesis. Introduction to the project is includes the problem statement, objectives of the project and the scopes of the project. Chapter 2 is the literature review. Research and literature review related to the project is presented. Chapter 3 is about the system methods or methodologies that will be applied. Procedures such as project planning, project analysis, design, testing and implementation will be discussed throughout this Chapter 3. Chapter 4 is the implementation of system including database, codes and interface design. Chapter 5 is about result and discussion. Its advantages and disadvantages, further enhancement is also discussed in Chapter 5. Chapter 6 is the conclusion of this project.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter discuss about the literature review of Cattle Management System (CMS). Techniques and method used in this system is explained throughout this chapter. Studies based on the information in several websites, article and journals were done to assemble needed particulars for this literature review

2.2 Existing System

This section is to discuss about the existing system that related to Cattle Management System (CMS). A brief explanation about manual cattle management and CattleMaxCs is stated in this section.

2.2.1 Manual Cattle Management

Those days, cattle are managed manually by farmers. The special physical appearances of each cows is used by the farmer as the cattle identification. The cattle behavior and grazing is used as the data of knowing the cattle health and preceding condition. At that timing, cattle are being well observed and studied by the farmer in order to know and remember the information that needed to run the management appropriately. Moreover many were skilled about the cattle farming and raising by their family and forebear at that era.

2.2.2 CattleMaxCs

CattleMaxCs is a software that used by oversea cattle farmers. CattleMaxCs offer a technology-based solution that allows you to keep up in the rapidly-changing cattle business. This program was created to meet the record keeping needs of cattle farmer. CattleMaxCs is based on Microsoft Access technology for maximum compatibility with Microsoft Windows. In addition, records and reports can easily be exported to Excel, Word/text and PDF formats.

CattleMaxCs support all the data about the cattle management. CattleMaxCs support and keep many adequate data about cattle management. The amount of cows can be always updated by using CattleMaxCs if there is any cow born or dead. CattleMaxCs keeps the sufficient data about the cattle and its improvement. Furthermore the health can be monitored by CattleMaxCs system and it will be recorded in the database. CattleMaxCs helps the farmers to keep an eye on the growth of cattle and the development of the management.

2.3 Development Methodology

A system development methodology in software engineering is a framework that is used to structure, plan, and control the process of developing an information system. This framework consists of multiple tools, models and methods which assist system development.

2.3.1 Software Development Life Cycle

Systems Development Life Cycle (SDLC) is any logical process used to develop an information system, including requirements, validation, training, and user ownership. SDLC is also known as the process of creating or altering systems, and the models and methodologies that used to develop systems. The concept usually refers to computer or information systems. In general, the phrases involved in SDLC are Planning, Analysis, Design, Implementation and Maintenance. Each of these phases is connected in a cycle form starting from Planning phases and ends at Maintenance phases. Figure 2.1 shows the SDLC phases and the flow. [1]

2.3.1.1 Planning

At this stage, the goals of the project are determined. Projects are typically evaluated in three areas of feasibility: economical, operational, and technical. It is also used as a reference to keep the project on track and to evaluate the progress of the system. This phase is also called the analysis phase. [1]

2.3.1.2 Analysis

This step involves breaking down the system in different pieces and drawing diagrams to analyze the situation. Analyze project goals, break down functions that need to be created, and attempt to engage users so that definite requirements can be defined. Requirement gathering sometimes require individual from client as well as service provider side to get a detailed and accurate requirements. [1]

2.3.1.3 Design

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. [1]

2.3.1.4 Implementation

Modular and subsystem programming code will be accomplished during this stage. The code is tested at various levels in software testing. Unit testing and module testing are done in this stage by the developers. System and user acceptance testing are often performed also. [1]

2.3.1.5 Maintenance

Maintaining the system is an important aspect of SDLC. As key personnel change positions in the organization, new changes will be implemented, which will require system updates. [1]

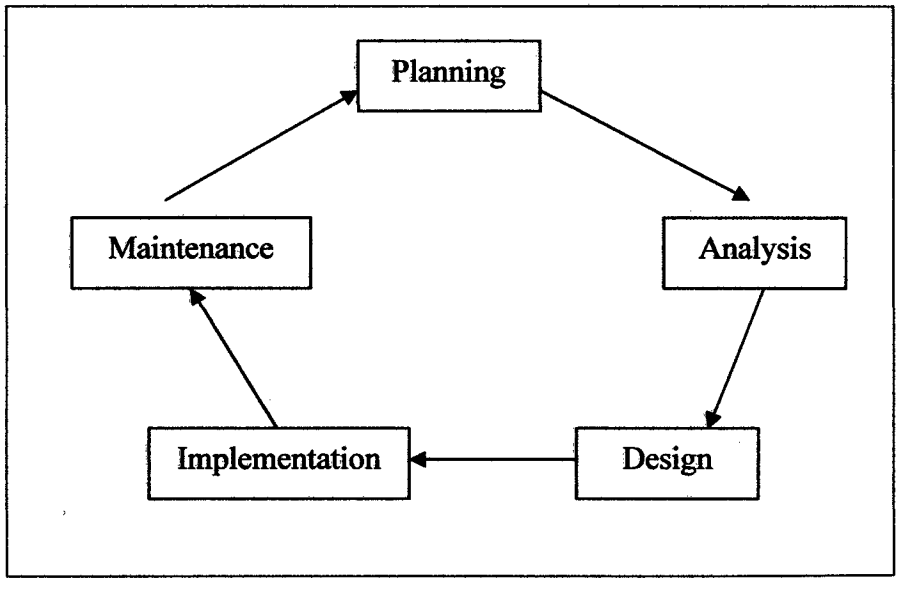

Figure 2.1: Software Development Life Cycle

2.3.2 Types of Software Development Life Cycle (SDLC)

There are few different types of Software Development Life Cycle (SDLC) model. In this section, explanation about the types of Software Development Life Cycle model will be discussed.

2.3.2.1 Waterfall Model

The waterfall model is a sequential software development process, in which progress is seen as flow gradually downwards through the phases of Requirement, Design, Implementation, Verification and Maintenance. In the waterfall model, one proceeds from one phase to the next in a purely sequential manner. Thus the waterfall model maintains that one should move to a phase only when its preceding phase is completed and perfected. [2]

In waterfall model, program requirements should be set in stone before design is started, the program's design should be perfect before people begin work on implementing the design. Otherwise they are implementing the wrong design and their work is wasted. The waterfall model provides a structured approach; the model itself progresses linearly through discrete, easily understandable and explainable phases and thus is easy to understand. It also provides easily markable milestones in the development process. Figure 2.2 shows the Waterfall Model. [2]

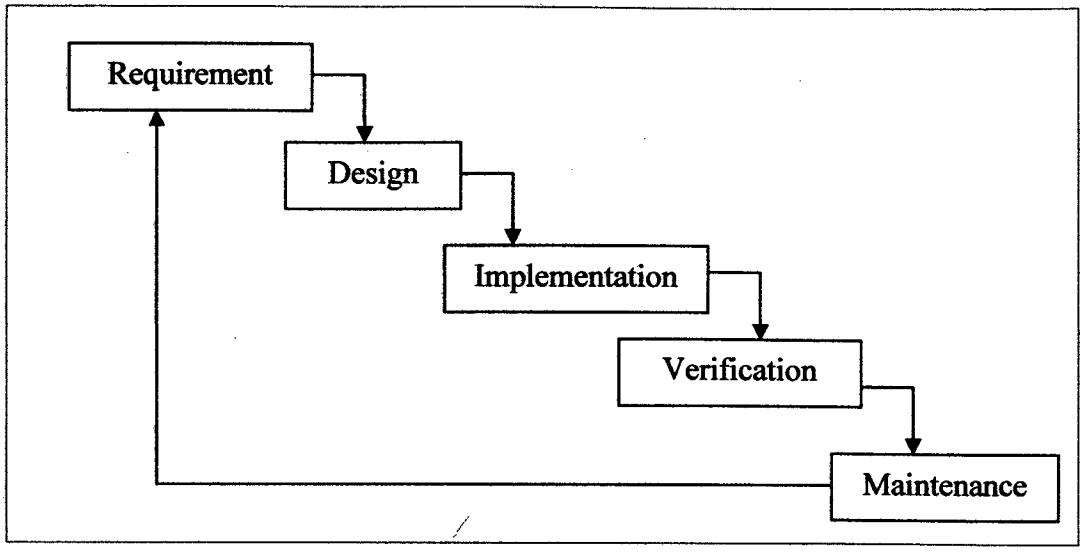

Figure 2.2: The Waterfall Model

2.3.2.2 Iterative Development

Iterative development is a cyclic software development process developed to resolve the weaknesses of the waterfall model. It starts with an initial planning and ends with deployment with the cyclic interaction in between. The iterative development is a vital part of the Rational Unified Process (RUP), the Dynamic Systems Development Method, Extreme Programming and generally the agile software development frameworks. Iterative development slices the system functionality into iterations. In each iteration, a slice of functionality is delivered through cross-discipline work, starting from the model/requirements through to the testing/deployment. [3]

2.3.2.3 Extreme Programming (XP)

Extreme Programming (XP) is a software engineering methodology which is intended to improve software quality and responsiveness to changing customer requirements. It advocates frequent "releases" in short development cycles, which is intended to improve productivity and introduce checkpoints where new customer requirements can be adopted. [4]

Other elements of Extreme Programming includes programming in pairs or doing extensive code review, unit testing of all code, avoiding programming of features until they are actually needed, a flat management structure, simplicity and clarity in code, expecting changes in the customer's requirements as time passes and the problem is better understood, and frequent communication with the customer and among programmers. The methodology takes its name from the idea that the beneficial elements of traditional software engineering practices are taken to "extreme" levels, on the theory that if some is good, more is better. [4]

2.3.2.4 Prototyping Model

Prototyping is an attractive idea for complicated and large systems for which there is no manual process or existing system to help determining the requirements. Prototyping is known as effective method to demonstrate the feasibility of a certain approach. This might be needed for novel systems where it is not clear that constraint can be met or that algorithms can be developed to implement the requirements. [5]

The basic idea of Prototyping model is that instead of freezing the requirements before a design or coding can proceed, a throwaway prototype is built to understand the requirements. Development of the prototype obviously undergoes design, coding and testing. But each of these phases is not done very formally or thoroughly. In prototyping model, errors can be detected much earlier as the system is mode side by side. Practically, this methodology may increase the complexity of the system as scope of the system may expand beyond original plans. Figure 2.3 shows the Prototyping Model.

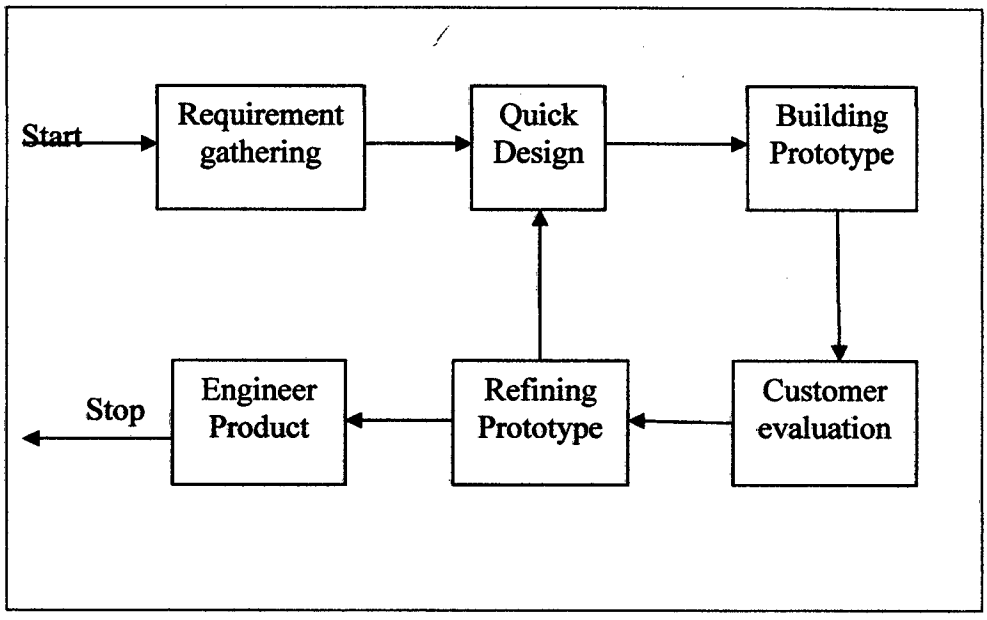

Figure 2.3: The Prototyping Model