Chapter 4

Approach to Solve the Problem

4.1 Introduction

The previous chapter was about what technologies/methodologies could be used to solve the concern problem. This chapter describes my own work in the project using selected technologies. Selecting the suitable technologies out of technologies that described in the previous chapter is also discussed.

4.2 The First Approach

In order to start the work of software process the requirements gathering in needed. The main stakeholder of the project is the CCD of the People's Bank. The existing system is operated in the branch wise. To identify the weakness of the existing system it has to consider branch staff as the stakeholder. The departments like Finance, Audit, and Branches etc are also taking as stakeholders because cheque clearing process is considered as a main functionality in banking.

Firstly in order to requirement gathering it was collected all the supporting documents already in the printed format. The LCPL has published a manual including the exchanging file format. And the interviewing all the stakeholders was great help for my first step. It had the luxury of many banks involving in identical business. So it could visit the Bank of Ceylon and made validation of requirement which were gathered.

At this point of time it was needed to select a software process model in order to carry out the project progress in a systematic manner. Following that it was done a comparison on several software models available.

4.3 Selecting Software Process Model

As per the project is concern there was completion selecting a party to carry. Several vendors were also there to acquire the opportunity. Then rapid progress as well as quality work was required. The stakeholders defined some milestones and they need deliverables at each point of milestones. And they continually asked for the visibility of the project. The comparison of software process models in the Table 4.1.

Feature	Waterfall Model	Evolutionary Model	Component base Model
Well Known Requirements	Need	Not need	Need
Concurrent Activities	Low	High	Medium
Versions Product	Mainly One	Three	One
Simplicity	High	Low	Low
Respond to Change of Requirements	Medium	High	Very-Low
Human Resources	Low	High	High
Visibility of Progress	High	High	Low

Table 4.1 : Comparison of Software Models

Based on the comparison table, the stake holder's requirements and nature of the project the Waterfall model is selected.

4.3 The System Analysis and Design Methodology

Wide available methodologies are Object Oriented Analysis & Design -OOAD and Structured System Analysis and Design Modelling -SSADM. Again go for a comparison on these two methodologies would be worth to selecting one. The said comparison is in the Table 4.2.

Feature	OOAD	SSADM
Document Overhead	Medium	High
Short Term Benefits	Low	High
Respond to Change Requirements	High	Low
Length of Process	Medium	High
Respond to Change of Requirements	Medium	High
Life Cycle Coverage	High	Low
Visibility of Structures	High	High
Maturity	Medium	High

Table 4.2 : Comparison of System Analysis and Design Methodologies

This comparison compelled me to select the OOAD methodology as system analysis and designing. In maturity wise the SSADM is in favor than the OOAD, but support for OOAD still available.

4.4 Unified Modeling Language -UML

The Unified Modeling Language (UML) is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system. The Unified Modeling Language offers a standard way to write a system's blueprints, including conceptual things such as business processes and system functions as well as concrete things such as programming language statements, database schemas, and reusable software components [7].UML includes a set of graphical notation techniques to provide structure diagrams and behavior diagrams. In the concerned project it is hope to provide use case, activity and sequence diagrams as behavior diagrams and class diagram as structure diagram.

4.4.1 Use Case Diagram- for Functional Requirements

The behaviour of the system under development is documented in a use case model that illustrate the system's intended functions (Use cases),its surrounding (Actors), and

[7] http://en.wikipedia.org/wiki/Unified_Modeling_Language

relationship between the use cases and actors (use case diagrams). Figure 4.1 shows the components of a use case diagram.

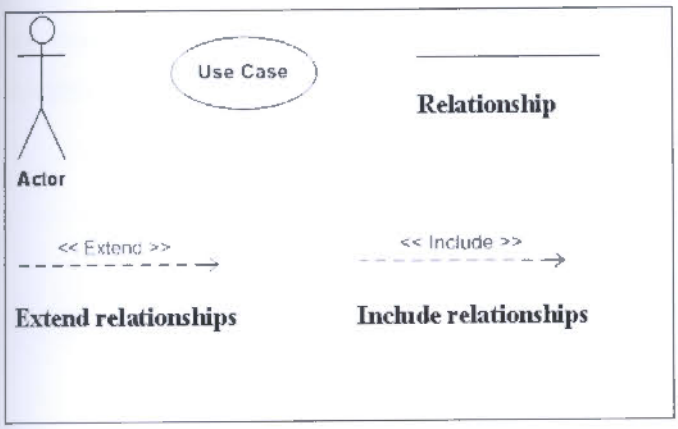

Figure 4.1 : Components of the use case diagram

An actor is someone or something outside the system that either acts on the system. An actor may be a person, a device, another system or sub-system. A use case represents a use case name provides a unique identifier for the use case. It should be written in verb-noun format. Relationships are the link between use case and the actors. In one form of interaction, a given use case may "include" another. The first use case often depends on the outcome of the included use case. In another form of interaction, a given use case (the extension) may "extend" another. This relationship indicates that the behavior of the extension use case may be inserted in the extended use case under some conditions.

Use Cases model a dialogue between an actor and the system. They represent the functionality provided by the system; that is what capabilities will be provided to an actor by the system. The Collection of use cases for a system constitutes all the defined ways the system may be used. With reference to Quatrani [22] (Quatrani, 2002), a use case is a sequence of transactions performed by a system that yields a measurable result of values for a particular actor.

The use case technique is used for capturing the functional requirements of the system. The use case developed for the proposed system is in the Appendix E.

---

[22] Quatrani T 2002, *Visual Modeling with Rational Rose 2002 and UML*, Pearson Education, New Delhi 110092

4.4.2 Activity Diagrams using Use Case Descriptions

For each use cases in the use case diagram use case descriptions were developed. Base on the description activity diagrams were developed. An activity diagram describes the dynamic behaviour of the system. Sample of activity diagram is shown in the Figure 4.2.

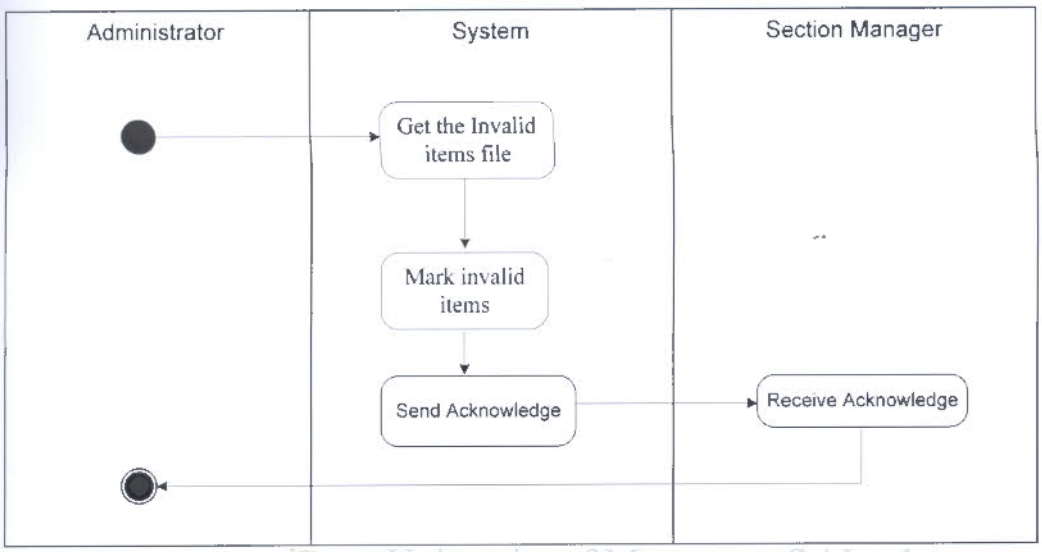

Figure 4.2 : Sample of Activity Diagram

In the Figure 4.3, Administrator initiates the process. Three activities were done by the "System" and pass to "Section Manager". Finally the Administrator terminates the process.

4.4.3 Sequence Diagrams using Activity Diagrams

UML sequence diagrams model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and are commonly used for both analysis and design purposes [8]. Sequence diagrams are the most popular UML artefact for dynamic modelling, which focuses on identifying the behaviour within your system. The activity diagrams pave the way to identify entity classes, interface classes and methods involved. Then the entire class diagram for the system can be developed. A sample of sequence diagrams is shown in the Figure 4.3.

---

[8] http://www.agilemodeling.com/artifacts/sequenceDiagram.htm

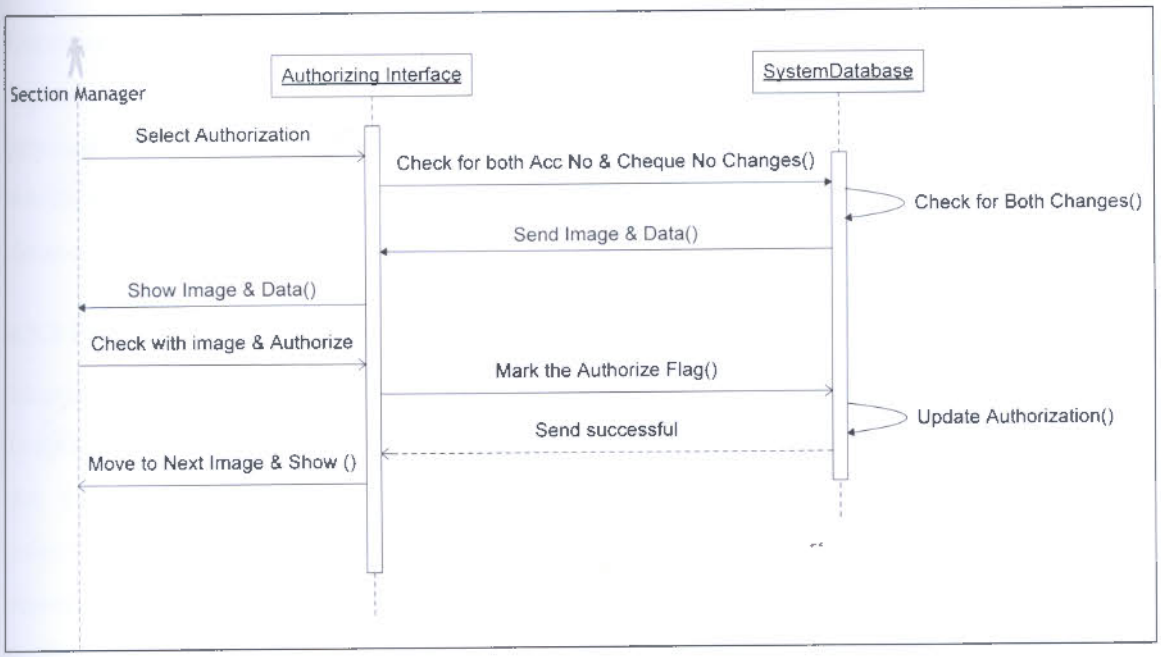

Figure 4.3 : Sample of Sequence Diagram

In the Figure 4.3 the actor "Section Manager" sends message call "Select Authorization" to interface class "Authorizing Interface". Then the "System Database" entity class calls a method call "Check for both Acc No & Cheque No Changes". Likewise sequence is moving unless the Actor gets the final result.

4.4.4 Class Diagram using Sequence Diagrams

In the Unified Modeling Language (UML), a class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes [2].A sample of class diagrams is shown in the Figure 4.4.

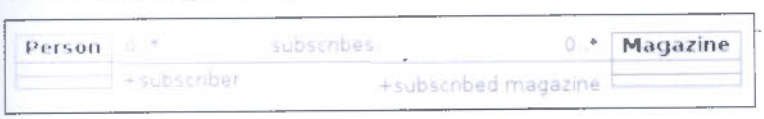

Figure 4.4 : Sample of Class Diagram

In the Figure 4.4 there are two classes called "Person" and "Magazine". Two classes were associates by "subscribe".

At this point Database Design could be started.

---

[2] http://en.wikipedia.org/wiki/Class_diagram

4.5 Database Design

Database design is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a Data Definition Language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

4.5.1 ER Diagram

Identifying the entities and their relationships is done using Entity Relationship-ER Diagram.

An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes. Using ER diagrams it can be identified the Primary Keys and Foreign Keys of respective relational tables. A sample of ER diagram is shown in the Figure 4.5.

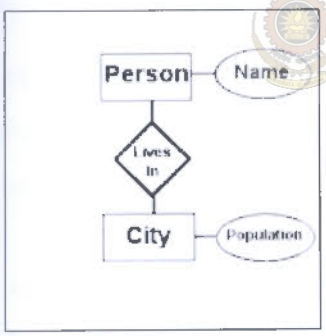

Figure 4.5 : Sample of ER diagram

4.5.2 Relational Database

In the relational database the entities become as tables and use relationships are used to create foreign keys. The primary keys are defined on the base table. In the Figure 4.5 "City" is the base table for "Person" Table. The columns of the respective tables are defined from there attributes.

4.6 Development Environment

By considering the facts describe in the chapter 3 section 3.6.4 and 3.6.5 the desktop environment is chosen for the main system development. Only the requirement "Customer cheque view" which is described in the Chapter 5 Section 5.1.1.1 Item Number (21), the web base is used.

4.6.1 Development Tools

Followings are the development tools used in the development. All the tools are MS Windows compatible and available within the People's Bank recourses.

Microsoft Visual Basic 6

A license copy of Microsoft Visual Basic 6 is available in the bank. The developer has 5+ years experience in this tool.

Microsoft Office Document Imaging – MODI

An image viewing tool is needed for cheque image view. The MODI is freely available with Microsoft Office package.

WAMP

For the web base development the Apache web server and PHP programming language is used. WAMP is also freely available. Adobe Dream weaver is used as interface design.

AlternaTIFF

A TIFF image viewer which is freely available was used in web base application.

Microsoft Word

The documentation is done using the Microsoft Word which is used for years for documentation.

Microsoft Visio

The diagrams, charts are created using Microsoft Visio which is also comes with Microsoft office package. A bit of practice was needed work with this package. It was so powerful for my work.

4.6.2 Operating System

As this development in desktop environment the Operating System is selected as Windows XP.

4.6.3 Database

The Microsoft SQL Server 2000 is used as the database. The bank has got the license copy of the said database.

4.7 Summary

This chapter describes technologies used in the software process for proposed system. In the selecting process the facts included in the Chapter 3 considered. The next chapter is discussed about system analysis and design of proposed system.