St. Mary's University Workers Saving and Credit Association Information System Automation

Addisu Anbessa and Bereketab Birhanu
St. Mary's University

1. Introduction

1.1. Background

Credit and saving association of SMU workers was established in 1996 EC. It is among the youngest and most important saving institutions established through Addis credit and saving association in Addis Ababa. SMU staff credit and saving association is locally organized group that meets gaps on saving and providing credit for its members. Every month, the members save starting from 5% up to 50% of their income. When a member needs to borrow a desired amount of money; he/she applies according to the association rule and regulation agreed upon. The credit and saving association of SMU workers is also very important to increase the members' habit of saving money and as a result improve their living conditions. In this study, considering the case of credit and saving association of SMU workers, we presented the method of automating the information system of credit and saving association in the following sections.

1.2. Statements of the Problem

Credit and saving association of SMU workers currently faces the following problems.

These are manually:

- Managing and retrieving a member's profile are resource consuming and not convenient.
- Computing interest for each member consumes resources (time, paper, labor etc…) and subjected to error.
- Computation of interest due for those who completely payback credit before the promised date is error prone.
- Managing of transactional expense and income is error prone (salary expense).
- Balancing members' accounts when the member terminates his membership from the association is also susceptible to error.

Members can't save some random amount voluntarily. Profit dividend management in the association has always been a problem.

1.3. Alternative Solutions, Assessment of Project Feasibility and Selection of the Optimal Solutions

For the problems of the association dictated above, there can be several methods or options to solve. Every solution may have benefits with certain drawback/defect. Therefore, this study will try to assess the available alternatives and select the optimal one.

1.3.1. Alternative Solutions

The study has generated the following three solutions that are meant for solving the problems prevailing in the current system.

We proposed the following three competing solutions to solve the problems prevailing in the current system.

1. Employ at least four full time workers and handling all activities manually.

2. Using MS-Words and MS-Excel.

3. Developing domain specific software system like in this study.

1.3.2. Assessment of Project Feasibility and Selection of the Optimal Solutions

The problem with the first option is high cost of resource and error prone. The second option results in scattered information system which is not suitable to manage. The third option is a software system helps to make well organized, automated and managed information system for the St. Mary's University workers credit and saving association. It can be implemented by using current technology and the resource the company owned. Since the company is in our university, all the requirements are easily accessible within scheduled resources. The software system that we are going to develop doesn’t violet the business logic and rules of the association. Therefore the third option above is preferable and feasible for the association than the others. Because, it reduces resource consumption and load on the workers on the other hand it increases efficiency and accuracy of the activities. Therefore, the selected optimal solution has tangible benefit in error reduction, increased flexibility and speed of activity. Generally, the solution has a very significant and feasible benefit in long term as the association grows up.

1.4. Objectives of the project

1.4.1. General Objective

The objective of our project is to develop a software system that helps to make well organized, automated and managed information system for the St. Mary's University workers credit and saving association.

1.4.2. Specific Objectives

The specific objectives of this project are to develop a software system that will be able to:-

- Register new members at any time
- Retrieve, update and manage members profile easily.
- Update amount of saving monthly.
- Calculate interest monthly.
- Generate reports at any time.
- Update the balance monthly.
- Access information online.
- Manage loan.
- Deactivate member’s profile that left the association.

1.5. Software Development Methodology

To develop the proposed system, first the suitable system requirement techniques were identified and then, the development methodology and engineering approaches were identified. In the next sections we deal with these activities.

1.5.1. Methods of Determining System Requirements

To determine the requirement of the proposed system, we used the following data collection methods. These are interview of stakeholders, actual observation of the work environment, questioners, document analysis and brain storming among the group members.

1.5.2. Software Development Process Model

Agile software development is a type of software development methods in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development, early delivery, continuous improvement, and encourages rapid and flexible response to change *(Wikipedia, the free encyclopedia).*

1.5.3. Software Engineering Methodology

Object oriented software engineering is the most widely used software design approach. In this approach, all the things and concepts in the project under development are treated as a class and an object which can model them as they exist in the real world. There is also a good feature of object oriented design (OOD) called **inheritance (code reuse)** which helps a programmer in developing the code. Generally, **Object Oriented Design** works on classes, their attributes and behaviors (methods). A class is a generalized description or representation of an object. An object is an instance (real occurrence) of a class. A class consists of all the attributes which an object can have and methods defines the functionalities of the object. Therefore, for our project, we prefer the **Object Oriented Design** methodology as the best option.

2. Survey of Technologies

There are different available technologies to develop our **SMU workers credit and saving association information System Automation**. This system is an integrated desktop based and web-based application. Since it needs appropriate software development technologies, we have to survey suitable technologies among the available for the desktop application such as Java, C#, C^{++}, eclipse etc. and for web-based application PHP, ASP, ASP.NET etc.

2.1. Web-based Application Development Technologies - PHP

PHP is a **server-side scripting language** designed for web development but also used as a general-purpose programming language. While PHP originally stood for Personal Home Page, it now stands for PHP: Hypertext Preprocessor, which is a recursive background.PHP code, can be simply mixed with HTML code, or it can be used in combination with various tempting engines and web frameworks. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge. PHP includes various free and open-source libraries in its source distribution, or uses them in resulting PHP binary builds. PHP is fundamentally an Internet-aware system with built-in modules for accessing File Transfer Protocol (FTP) servers and many database servers, including MySQL, Microsoft SQL Server and SQLite (which is an embedded database).

ASP

Active Server Pages (ASP), also known as Classic ASP, was Microsoft's first server-side script engine for dynamically generated web pages. The use of ASP pages with Internet Information Services (IIS) is currently supported on all supported versions of IIS. Microsoft's ASP technology runs only on Windows platforms. A number of products emulate some of the functionality of Classic ASP on non-Microsoft web servers. Apache::ASP for example ports Classic ASP to the Apache Web Server, but does not interpret Visual Basic or other scripting languages supported by ASP.

ASP.NET

ASP.NET is an open source **server-side Web application framework** designed for Web development to produce dynamic Web pages. It was developed by Microsoft to allow programmers to build dynamic web sites, web applications and web services. ASP.NET is built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET language.ASP.NET is in the process of being re-implemented as a modern and modular web framework, together with other frameworks like Entity Framework. The new framework will make use of the new open-source .NET Compiler Platform and be cross platform. ASP.NET Web pages, known officially as Web Forms, are the main building blocks for application development. ASP.NET's **Code-behind** model marks a gap from **Classic ASP** in that it encourages developers to build applications with separation of presentation and content in mind. ASP.NET aims for performance benefits over other script-based technologies by compiling the server-side code the first time it is used to one or more DLL files on the Web server.

ASP.NET simplifies developers' transition from Windows application development to Web development by offering the ability to build pages composed of controls similar to a Windows user interface. A Web control, such as a button or label, functions in very much the same way as its Windows counterparts: code can assign its properties and respond to its events. ASP.NET encourages the programmer to develop applications using an event-driven GUI model, rather than in conventional Web-scripting environments like ASP and PHP. The framework combines existing technologies such as JavaScript with internal components like "View State" to bring persistent (inter-request) state to the inherently stateless Web environment.

Therefore, for our project the technology which best fits among the available technologies is ASP.NET because of the above mentioned features and functionalities it provides.

2.2. Desktop Application Development Technologies C# versus other Programming Language

C# is a multi-paradigm programming language encompassing strongly typed, imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. C# is one of the programming languages designed for the Common Language Infrastructure. C# is intended to be a simple, modern, general-purpose, object-oriented programming language. The language is intended for use in developing software components suitable for **deployment in distributed environments**. C# is intended to be suitable for writing applications for both **hosted** and **embedded** systems, ranging from the very large that use sophisticated operating systems, down to the very small having dedicated functions.

Java and C# are similar programming languages that are statically, strongly, and manifestly typed, both are class-based object-oriented, both are designed with semi-interpretation or runtime compilation in mind, both use garbage-collection, and both are "curly brace languages" like C and C++.

Though C# and VB.NET are syntactically very different, that is where the differences mostly end. Microsoft developed both of these languages to be part of the same .NET Framework development platform. They are developed, managed, and supported by the same language development team at Microsoft. They compile to the same intermediate language (IL), which runs against the same .NET Framework runtime libraries. Although there are some differences in the programming constructs, their differences are primarily **syntactic** and, almost every command in VB has an equivalent command in C# and vice versa. Lastly, both languages reference the same Base Classes of the .NET Framework to extend their functionality. As a result, with few exceptions, a program written in either language can be run through a simple **syntax converter to translate** to the other.

Therefore, for our project the technology which best fits among the available technologies is C# because of the above mentioned features and functionalities it provides.

3. Requirement Analysis

3.1. Introduction

Software requirement analysis is one of the major phases of software development life cycle. This software engineering phase is the stage at which the project team starts to communicate with the stakeholders to get a better and precise understanding of what the client actual needs from the proposed project, by analyzing how the current system is operating. In doing so, it is possible to clearly identify what features and functionalities of the existing system will be captured in to the proposed system and the features that cannot be handled by the system. Therefore, in the coming sections of this chapter, we will focus on requirement specification, hardware and software requirement, functional, structural and behavioral modeling of the system.

3.2. Requirements Specification

Software requirements are the statements of what features, functionalities and services the proposed system should provide. To gather these requirements, the relevant stakeholders from the client side are involved to reflect what they need from the proposed system. The project team used several requirement gathering techniques like interviewing, questionnaire, actual observation the current system to acquire users' requirements. The requirements are then stated clearly in a structured language so that both the client and the project team can understand equally and agree up on. Finally, a document of software requirement is produced and become requirement specification of the system when validated, completed and accepted by the client and the project team.

Member Registration

The association accepts a new member only twice a year on January and July. The acceptance of a newly requesting member is approved by administrative committee of the association. When a new member get acceptance the association registers his/her profile like full name, age, sex, address, monthly saving (at least 5% of monthly income), registration date and legal inheritors. The association does the registration manually on the format prepared for this purpose and create new ledger and file folder for each member. The proposed automation system would handle the registration and filing activity effectively but cannot approve the acceptance of a new member.

Saving

The association manually collects saving money from each member monthly using revenue collection receipt and records it in to the member's file (ledger). The association also manually compute interest amount every six month on saving balance. All the collection of saving and computation of interest on the saving balance can be done with better accuracy and efficiency by the proposed automation system.

Credit

The association provides credit for the members up on request. For a member, to request for a credit he/she has to save at least for successive six months starting from registration date. In addition, the member has to present guaranty (assurance persons) and convincing reasons to get the credit. The association grants member a credit of four folds of his/her saving balance to be completely paid back within a given months. The maximum credit amount the association can provide is 30,000 Birr for 60 months. In the association, a credit is permitted to a requesting member by credit letting committee. When a credit is granted to a member, it is recorded to the member's file. The association computes an interest of 7.5% annually on the credit amount. Within the specified months, the association collects back the credit and interest monthly on average basis. The proposed system can automate all the above manual activities efficiently except the granting of the credit.

Transaction

The association executes several transactions such as collecting revenues and payment of expenses. In the association, revenues are collected in the form of monthly saving, registration fee, share, credit payback, interest on credit and other miscellaneous. This activity is accomplished using revenues collecting receipt and recorded in to account journal as revenue by the accountant clerk of the association. The association also has some expenses to be paid as salary of workers, purchasing of stationery items, maintenance, permitted credit, repayment of saving when a member leaves the association and other miscellaneous. At the end of every month, all the revenues and expenses in that month are summed up and posted to the general ledger of association. The proposed automation system can handle the above manual activities (computations and filings).

Reports

The association prepares a report of every financial activity (financial statements) that express:-

-Saving: -the total sum of monthly collected saving money.
-Credit: -the balance of the total credit amount given to the requested members.
-Registration fee: -the total balance of the money collected as registration fee, from members who newly join the association.
-Share: -the total balance of the money collected as share, from members who newly join the association.
-Revenue: -the total balance of all the revenues of the association.
-Expense: -the total balance of all the expense of the association.
-Profit:-the total annual profit balance of the association.
-Net profit:-the total annual net profit balance of the association.
-Dividend profit:-60% of the total annual net profit balance of the association.
-Non-Dividend profit:-30% of the total annual net profit balance of the association.

3.3. Software and Hardware Requirements

Hardware Requirement

To implement this software system the hosting machine should have a minimum capacity of-Processor Speed: Intel® Core ™ Dual Core 1GHZ, installed Memory (RAM): 1GB and Hard Disk: 80GB

Software Requirement

To execute this software system the hosting machine should have **Window 7 Ultimate** and later versions of **Windows family** as platform.
-System Type: 32bit Operating System

3.4. Functional Modeling

Use Case

Use case is a standard technique for gathering requirements in many modern software development methodologies. It represents the functional requirements of a system under development and captures the business processes carried out in the system. A use case describes ***what*** a system does; not ***how.***

The Use case diagram is used to identify the primary elements and processes that form the system. Pertinent to our project, **SMU Workers Credit and Saving Association Information system automation,** the following use case are included and described as below.

Log in

The system shall permit users to log into the system by using their user name and password given by the administrator.

Log out

The system shall permit users to log out from the system and return back to the log in interface when they finish their work.

Register Member

The system shall permit the accountant to capture all necessary information about new members. The system shall enable the accountant to register the name, sex, age, address, etc. of the member on the space provided and add to the database.

Register Inheritor

The system shall permit the accountants to capture all necessary information about legal inheritors. The system shall enable the accountant to register the name, sex, age, address, etc. of the legal inheritor on the space provided and add to the database.

Collect Revenue

The system shall permit the accountants to collect all revenues of the association. It enables the accountants to record saving and credit payback information to the database.

Maintain Record

The system shall permit the accountants to maintain or update records.

Record Credit Amount

The system shall permit the accountants to record information about credit amount taken by the members from the association to the database.

Record Expense

The system shall permit the accountant to record expense information of the association to the database.

Compute Profit

The system shall permit the accountants to compute the profit of the association.

Generate Report

The system shall permit the accountants and managers to generate report.

View Income

The system shall permit the accountants, managers to view the income of the association.

View Expense

The system shall permit the accountants, managers and auditors to view expense of the association.

View Profit

The system shall permit the accountants, managers and auditors to view the profit of the association.

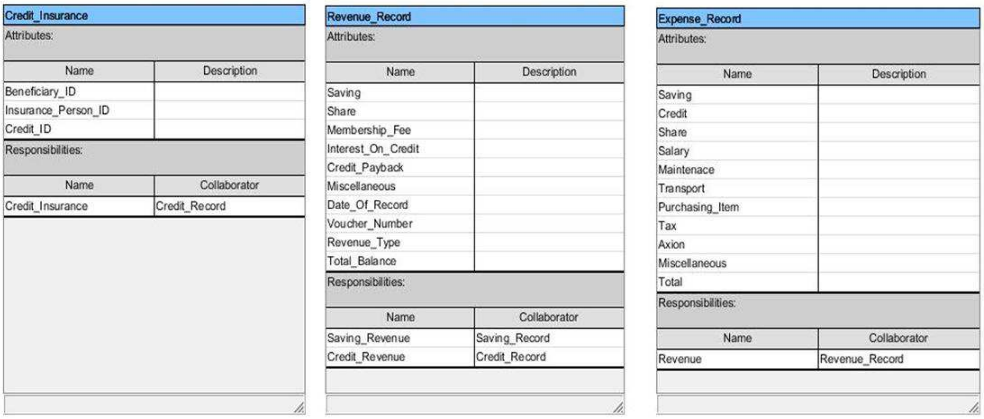
View Report

The system shall permit the accountants, managers and auditors to view the overall reports of the association.

Write Comment

The system shall permit auditors to write comments about their findings to the association.

Simple Class Diagram



Credit_Insurance

Attributes:

Name	Description
Beneficiary_ID	
Insurance_Person_ID	
Credit_ID	

Responsibilities:

Name	Collaborator
Credit_Insurance	Credit_Record

Revenue_Record

Attributes:

Name	Description
Saving	
Share	
Membership_Fee	
Interest_On_Credit	
Credit_Payback	
Miscellaneous	
Date_Of_Record	
Voucher_Number	
Revenue_Type	
Total_Balance	

Responsibilities:

Name	Collaborator
Saving_Revenue	Saving_Record
Credit_Revenue	Credit_Record

Expense_Record

Attributes:

Name	Description
Saving	
Credit	
Share	
Salary	
Maintenace	
Transport	
Purchasing_Item	
Tax	
Axion	
Miscellaneous	
Total	

Responsibilities:

Name	Collaborator
Revenue	Revenue_Record

Fig.1 CRC Diagram

4.System Design

4.1. Introduction

A software design is a description of the structure of the software to be implemented, the data which is part of the system, the interfaces between system components and, the algorithms used. It is not possible to arrive at a finished design stage immediately but design development is done iteratively through a number of trial versions. The design process may involve developing several models of the system at different levels of abstraction. In the coming subsections of this chapter, we use class and method designs to illustrate class specifications (using class diagrams) and method specifications. Next, the data management layer is addressed by designing the actual database or file structure to be used for object persistence, and a set of classes that will map the class specifications into the object persistence format chosen.

4.2. Class and Method Design

Software Design is a process to conceptualize the software requirement into software implementation. It takes the user requirements as problems and tries to find optimal solution. In our project, we follow the object oriented (OO) software design approach to find the optimal solution. Object Oriented Design works on classes, their attributes and behaviors (methods). A class is a generalized description or representation of an object. An object is an instance (real occurrence) of a class. A class consists of all the attributes which an object can have and methods defines the functionalities of the object. In the design of solution attribute are stored as a variables and functionalities are defined by means of methods.

Adding Specifications

In our problem domain SMU Workers Credit and Saving Association we have following classes with their specifications.

Member Profile

This class is used to define the attributes and methods for the member profiles. The attributes are Full Name, Age, Sex, Date of Registration, Salary, and Saving Percentage. The methods are Record Member Profile and Display Member Profile.

Member Inheritor

This class is used to define the attributes and methods for the member's inheritor profiles. The attributes are Full Name, Age and Sex. The method is Record inheritor Profile.

Address

This class is used to define the attributes and methods for the address of members and their inheritors. The attributes are City, Sub-City, Kebele, House Number and Phone Number. The method is Record Address.

Saving Record

This class is used to define the attributes and methods for the saving record class. The attributes are Saving Amount, Saving Type, Saving Interest, Saving Balance, Saving ID, Date of Collection and Voucher Number. The methods are Compute Saving Amount, Compute Saving Balance, Record Saving and Display Saving.

Credit Record

This class is used to define the attributes and methods for the credit record class. The attributes are Credit Amount, Credit duration of month, Credit Monthly Payback Amount, Payback Round, Credit ID, Date of issue, Reason of Issue and Voucher Number. The methods are Credit Monthly Payback Amount, Record Credit and Display Credit.

Credit Insurance

This class is used to define the attributes and methods for the credit insurance class. The attributes are Beneficiary ID, Insurance person ID and Credit ID.

Revenue Record

This class is used to define the attributes and methods for the revenue record class. The attributes are Saving, Share, Membership Fee, Interest on Credit, Credit Payback, Miscellaneous , Date of Record, Voucher Number, Revenue Type and Total Balance. The methods are Compute Revenue Balance, Compute Total Revenue, Record Revenue and Display Revenue.

Expense Record

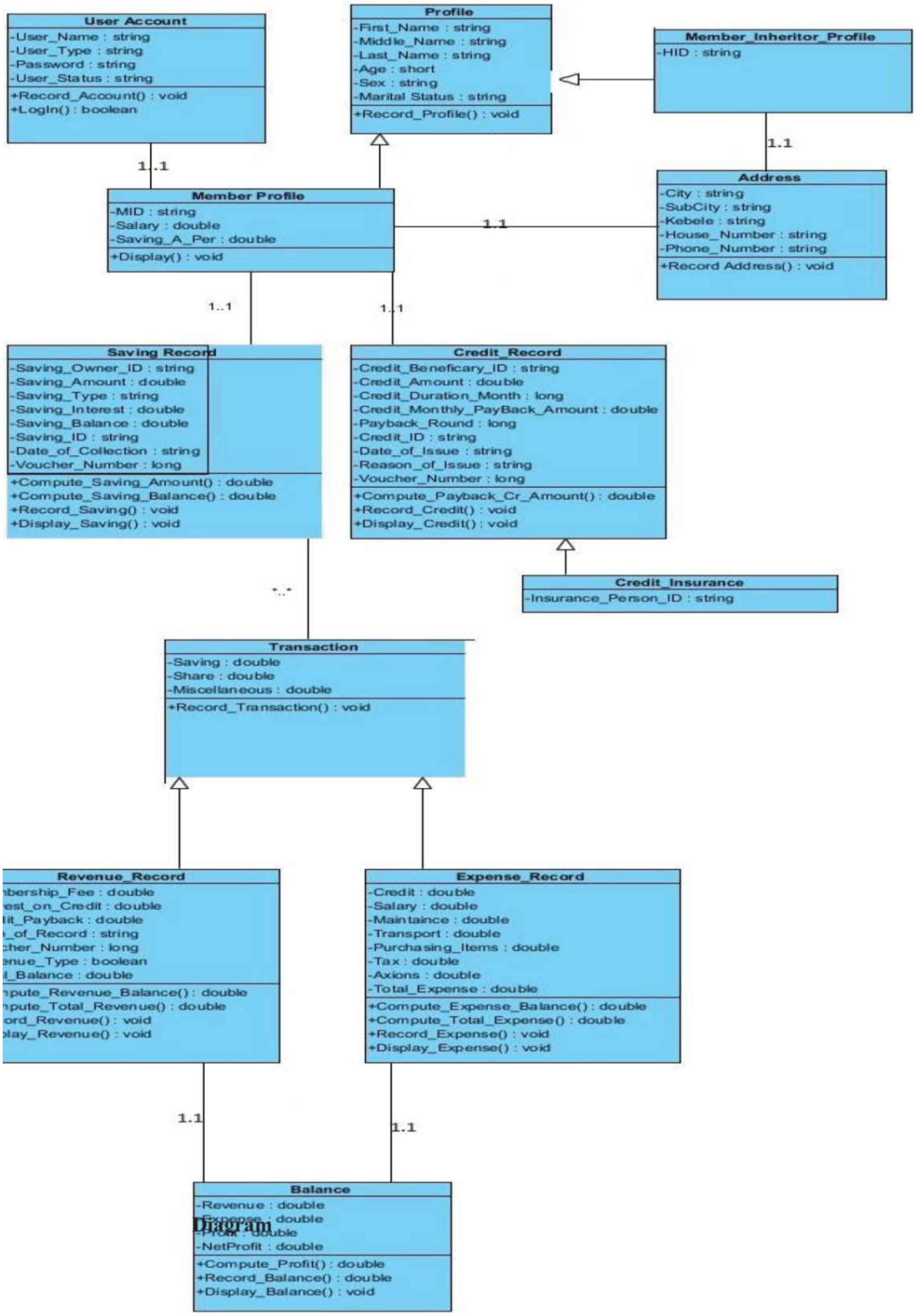
This class is used to define the attributes and methods for the expense record class. The attributes are Saving, Credit, Share, Salary, Maintenance, Transport, Purchasing Item, Tax, Axion, Miscellaneous and Total. The methods are Compute Expense Balance, Compute Total Expense, Record Expense and Display Expense.

Balance

This class is used to define the attributes and methods for the balance class. The attributes are Revenue, Expense, Profit and Net Profit. The methods are Compute Profit, Record Balance and Display Balance.

User Account

This class is used to define the attributes and methods for the user account class. The attributes are User Name, User Type, Password and User Status. The methods are Record Account and Log In.For our problem domain classes, we have modeled the class diagram model as shown in the next Fig. 4.1



Diagram

Algorithm Specification

Algorithm specification is used to develop a systematic process for business logic of a software application. In our project **SMU Workers**

Credit and Saving Association Information System Automation, the core algorithm specification is as follows.

```
ComputeCredit()
{
    Interest=(CreditAmount*0.625)/100
    Totalinterest= Interest
    MoonthlyPaybackcredit= CreditAmount/NumberOfMonth
    For(i=1; i<= NumberOfMonth;i++)
    {
        CreditAmount= CreditAmount- MoonthlyPaybackcredit
        Interest=(CreditAmount*0.625)/100
        Totalinterest= Totalinterest+ Interest
    }
    AverageInterest= Totalinterest/ NumberOfMonth
}
ComputeSaving()
{
    Interest=(SavingBalance*0.417 )/100
    SavingBalance= SavingBalance+ Interest+MonthlySaving
}
```

4.3. Data Management Layer Design

For every problem domain classes there is **Back End** data management layer that will be developed using **Database Management System (DBMS)**. In our project we use **SQL Server2008** to design Database. For each class in the problem domain there is a corresponding **database table** which is used to hold all the attributes of the class. An instance of a domain class affects a single of **row of the table**.

5. Conclusion

The preparation of SMU workers credit and saving association information system automation document has passed through four stages up to this end. These stages are Introduction, surveys of technologies, Analysis and System design. In the introductory part, first, we looked at identifying and stating the problems of the system. Next to that, we went to search for possible options to resolve the problems and selected our project by assessing its feasibility. Lastly, we have stated the main objectives of our project and identified the alternative development methodology and engineering approaches to achieve the objectives. In the survey of technology part, what we did was, assessing the available technologies based on their features and functionalities in order to come up with the one that is suitable for our project. As per our assessment, we have selected C# programming languages and SQL server 2008.

At the analysis part, the major task was, identifying and specifying all the things in the problem domain and gathering system requirements through different techniques. Regarding our project, we used interviewing the stakeholders, document analysis and actual observation of the work environment for requirement gathering. To model the system we have used unified modeling language diagrams like use case diagram for functional modeling, CRC (class responsibility collaborator) for structural modeling, sequence and activity diagram for behavioral modeling. Finally, at the system design the mechanisms how to implement the system. The core tasks were, mapping the problem domain classes to the implementation language and data storage relational database. At the end, the core algorithm specifications and physical architecture design of the system were done.

6. Recommendation

SMU Workers Credit Saving Association Information System Automation an integrated window and web based application. This application can be used with little or no modification other similar associations. In our opinion for an association with less number computers and no branches the pure desktop application (Window based application) is the most convenient. Because the web based application needs more resources. But for larger associations with enough resource capacity the web based application is suitable.

References

Ian Sommerville: Software Engineering, Eighth Edition

Programmer’s Heaven C# School, First Edition By FarazRasheed

Software Engineering Tutorial, from WWW.tutorialspoint.com

Wikipedia, the free encyclopedia.