PERSONAL FINANCE MANAGEMENT USING AGILE TECHNIQUES

Submitted by:

Akhil Gupta (081455)
Kailash Sharma (081449)
Ashish Janartha(081410)
Pankaj Chauhan(081332)

Supervised by:

Dr. Yashwant Singh

May 2012

Submitted in partial fulfilment of the Degree of

Bachelor of Technology

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING AND INFORMATION TECHNOLOGY

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT, SOLAN (H.P.)

INDEX

CERTIFICATE IV

ACKNOWLEDGEMENT V

SUMMARY VI

LIST OF ABBREVIATIONS VII

LIST OF FIGURES VIII

LIST OF TABLES IX

Chapter 1: Introduction 1

1.1 Project Vision 3

1.2 Project Scope 4

Chapter 2: Literature Survey 6

2.1 Agile Mnifesto 7

2.2 Principles 7

2.3 Agile Methods 8

2.3.1 Agile Modelling 8

2.3.2 Extreme Programming 9

2.3.3 DSDM 10

2.3.4 SCRUM 10

2.3.5 Sprint 11

2.3.6 FDD 11

2.3.7 Crystal 13

2.4 Agile vs traditional 13

Chapter 3: Implementation of Scrum 14

3.1 Sprints 15

3.2 Architecture Design 16

3.3 Planning 17

3.4 UML Diagram 18

3.4.1 Use Case Diagrams 22

3.4.2 Sequence Diagrams 23

3.4.3 Data Flow Diagram 24

3.4.4 Class Diagram 25

3.4.5 Deployment Diagram 28

3.4.6 Activity Diagram 30

3.5 Repository Design 31

Chapter 4: Conclusion And Future Scope **32**
4.1 Conclusion 33
4.2 Future Scope 34
APPENDIX A **40**
APPENDIX B **42**
REFERENCES **48**
RESUME OF STUDENTS **50**

CERTIFICATE

This is to certify that the work titled "**PERSONAL FINANCE MANAGEMENT SYSTEM USING AGILE SOFTWARE DEVELOPMENT METHODOLOGY**" submitted by **Akhil Gupta, Ashish Janartha, Kailash Sharma** and **Pankaj Chauhan** in partial fulfilment for the award of degree of B.Tech of Jaypee University of Information Technology, Waknaghat has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor: 29/05/2012

Name of Supervisor: ..Dr. Yashwant Singh

Designation: ..Assistant Professor.

Date: ...29/5/12.....

ACKNOWLEDGEMENT

Apart from the efforts, the success of any project depends largely on the encouragement and guidelines of many others. Therefore we take the opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We would also like to show our appreciation to our project guide Dr. Yashwant Singh. Without his able guidance, tremendous support and continuous motivation the project work would not be carried out satisfactory. His kind behaviour and motivation provided us the required courage to complete our project.

Special thanks to our project panel because it was their regular concern and appreciation that made this project carried out easily and satisfactorily.

Signature of Student
Name Akhil
Date 29/5/12

Signature of Student
Name Pankaj
Date 29/5/12

Signature of Student
Name Kailash sharma
Date 29/5/12

Signature of Student
Name Ashish Jamartha
Date 29/5/12

SUMMARY

The project taken by us Personal Financial Management using agile software methodology helps to meet personal finances of users in day to day life. It is innovative software which will also record and save passwords, any future plans and reminders. This software is also good for calculating income tax for male, female or senior citizens. It would simplify things as current scenario includes more of manual work and record keeping and be very difficult. It would have a very simple user interface and easier to maintain and handle. This project is basically a money manager which provides a complete view of your financial status anytime, anywhere. It helps to manage various bank accounts, credit cards, mutual fund investments, shares, Insurance policies, loans, Fixed Deposits etc., anything that revolves around your money. FMS has an automated Income tax wizard that helps in IT computation. It also helps you to manage your passwords and reminders. Furthermore it helps in managing your future plans and helping out with saving techniques. It is unique in its own way as users does not has to hustle around for records because it is a one stop for all your finance management objective of Personal Financial Planning is with an aim to make sure that we have enough money when it is most required. Day to day expenses and living .For educating children, perform marriages .At the time of Sickness and to lead a decent living on retirement. It helps to establish short- and long-term goals, determine current net worth.

Signature of Student

Name Akhil Gupta

Date 29/5/12

Signature of Student

Name Pankaj

Date 29/5/12

Signature of Student Kailash sharma.

Name Kailash sharma.

Date 29/5/2012

Signature of Student

Name Ashish Tanartha

Date 29/5/12.

LIST OF FIGURES

Figure	Figure Description	Page No.
Fig 2.1	Extreme Programming	03
Fig 2.2	DSDM	04
Fig 2.3	Scrum	07
Fig 2.4	FDD	23
Fig 2.5	Agile vs Traditional	24
Fig 6.3	Tier architecture	26
Fig 7	Use Case Diagram	27
Fig 8.0	Sequence Diagram	28
Fig 9.0	Activity Diagram	29
Fig10.0	Class Diagram	31
Fig 11.0	Data Flow Diagram	36
Fig 12.0	Deployment Diagram	37

LIST OF TABLES

Figure	Figure Description	Page No.
Table1.	Income	30
Table 2.	Expenditure	32
Table 3.	Daybook	33
Table 4.	Reminder	34
Table 5.	Future	35
Table 6.	Password	36

LIST OF ACRONYMS

Acronym	Description
FMS	Financial Management System
XP	eXtreme Programming
DSDM	Dynamic Systems Development Method
AUP	Agile Unified Process
EssUP	Essential Unified Process
RUP	Rational Unified Process
ExP	Exia Process
FDD	Feature Driven Development
OpenUP	Open Unified Process
FMI	Functional Model Iteration
DBI	Design & Build Iteration
BAD	Business Area Definition
SAD	System Architecture Definition
OPP	Outline Prototyping Plan
TDD	Test Driven Development
UML	Unified Modelling Language
IDE	Integrated Development Environment
JDBC	Java Database Connectivity

Chapter 1

1. Introduction

Financial Management System is specifically developed for comprehensive solution to manage personal finances. It facilitates user to manage all accounts in one place, schedule the payment of bills, Plan and project for your financial future, calculate income tax, manage passwords etc.Even though one of the most significant factors in our life is the state of our persona! finances, we rarely spend time on managing them since unlike businesses, we are not accountable to any one for our personal financial goals and results. At the very basic level of personal finance you are dealing with a budget; you make money and then you spend that money. Even if you haven't created a detailed and written budget you continue to do budgeting on a daily basis. When you are faced with spending money on something you think about it and realize that by spending that money you will not be able to spend that same money on something else.

The problem that stems from not having a detailed budget are that we are faced with so many financial decisions it is nearly impossible to keep track of and remember everything. This lack of understanding can lead to overspending, debt problems or even the inability to adequately plan for your future.The main goal of the FMS is to keep track of all the financial endeavours and ensure that the money is correctly managed all the time thus reducing paper work and improve the efficiency and enhance the productivity.It can keep the information of Saver, Loaner, Share Holder, Expenditure, Income and Contra.

The exciting part of this project is it displays the daybook, Profit and Loss account, Statistical Summary and Interest Information. In the existing system the transactions are done only manually but in proposed system we have to computerize all using the software Financial Management System. It is a money manager which provides a complete view of your financial status anytime, anywhere, bank accounts, credit cards, mutual fund investments, shares, Insurance policies, loans, Fixed Deposits etc. anything that revolves around your money. FMS has an automated Income tax wizard that helps in IT computation. We can make a much larger contribution in every area of our life when our personal finances, investments and taxation are properly plan edit also helps you to manage your passwords and reminders.

Furthermore it helps in managing your future plans and helping out with saving techniques. It is unique in its own way as users does not has to hustle around for records because it is a one stop for all your finance management objective of Personal Financial Planning is with an aim to make sure that we have enough money when it is most required for Day to Day expenses and

living .For educating children, Perform marriages, at the time of Sickness and to lead a decent living on retirement. It helps to establish short- and long-term goals, determine current net worth figure out current spending patterns examine the current tax situation insurance to manage risk set a budget to control spending and to invest surplus money to achieve financial goals.The methodology to be applied while developing this project will be Agile Software Development: Scrum. It is a new methodology where project is developed iteratively and incrementally with minimum documentation. At the end of a fixed time which is very short we will be able to develop working software. After which regular increments would be done. This project is basically a money manager which provides a complete view of your financial status anytime, anywhere. It Bank accounts, credit cards, mutual fund investments, shares, Insurance policies, loans, Fixed Deposits etc, anything that revolves around your money.

FMS has an automated Income tax wizard that helps in IT computation. It also helps you to manage your passwords and reminders. Furthermore it helps in managing your future plans and helping out with saving techniques. It is unique in its own way as users does not has to hustle around for records because it is a one stop for all your finance management objective of Personal Financial Planning is with an aim to make sure that we have enough money when it is most required. Day to day expenses and living .For educating children, perform marriages .At the time of Sickness and to lead a decent living on retirement. It helps to Establish short- and long-term goals, determine current net worth,.The system will provide the tools to create, maintain and manage all the expenses and income. Upon completion of the project, the user will be able to manage all his resource at one place.

FMS has following entities:

Income: Economic wealth that is generated in exchange for an individual's performance of agreed upon activities or through investing capital. Income is consumed to fuel day-to-day expenditures. It is used to input all the income which a user has from various sources. They can be from any source for example monthly income from govt source or business source, income from selling of property goods etc. It is a miscellaneous collection of your income.

Daybook: Day to day spending record is kept here. We spend quite a lot amount in daily expenses like baying vegetables, paying for small things, petrol, and leisure activities. This all accounts for a big sum and add up a great amount at the end of month..

Expenditure: The total spending's of entire month is managed in this entity. This accounts for spending on bills, house rent, insurance etc.

Income Tax: Income tax can be defined as all sources of income other than agricultural income which Central Government collects levies on that and shares the same with the states. As per Income Tax Act of 1961, all persons who are considered as an assess and their when their income exceeds the maximum exemption in the prescribed limit and the income tax will be levied at the prescribed rates according to finance act, such type of income tax has to be paid on the total income in the previous year to be paid in relevant assessment year. Calculate income Tax, which a user has to pay. We all have to pay income tax if we have some decent living. This module calculates all the income tax one has to pay during financial year. Using a acute algorithm this module calculates your income tax whether you are a man, woman or a senior citizen.

Reminders: Reminder is a type of time management computer software that is designed to alert the user of important events that they have input to the program. Most programs provide a calendar and/or list view of events, as well as a reminding technique. Most common reminding techniques are pop-up dialog boxes and auditory alarms. This entity lets user to keep in track anything, which he/she has to do reminders are an important aspect of our life. We can't remember everything. So this tool reminds us of anything which we have to do like paying of bills car insurance etc.

Future Plans: To keep track of any future planning. We all plan for our future. This module helps to plan for future and to modify it like we want.

View: To view any inputs from the user. Obviously one will want to view all the things he has input in the system. This module helps in screening all the entries made

Help: Contains information on how to use the system. It will open a notepad file containing all the help one needs while using the system.

1.1 Project Vision

The vision for the FMS Project is to provide software for managing Personal Financial records. The system will provide the tools to create maintain and manage all the expenses and income. Upon completion of the project, the user will be able to manage all his resource at one place

Our vision is to create an environment that:

- Keeps track of all there financial endeavours and ensure that their money is correctly managed all the time.
- Manage all of your accounts in one place.
- Control all of your spending.
- Schedule the payment of bills.
- Plan and project for your financial future.

1.2 Project Scope

A clear scope is important to any project because it defines those areas that will be addressed by the project as well as those areas that will not be included in the project. Scope management will be a critical task during the project in order to ensure an on-time and on-budget implementation. Any changes in the defined scope of this project could potentially cause an increase in cost and extension of the project timeline. The scope of the FMS implementation project is to evaluate the current business process to meet the needs of all users who want an easy method to manage their finances. The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces the manual work. In the existing system the savings are done only manually but in proposed system we have to computerize all the savings using the software Financial management system Previously the Finance management had to be done manually. There was more of manpower. It was time consuming, more paperwork, had manual calculations and there was no proper record. So with the proposed system we have security of data, data accuracy, faster processing, greater efficiency and minimum time required. It is user friendly and interactive.

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It will help in the application of the principles of finance to the monetary decisions of an individual or family. It addresses the ways in which individuals or families obtain, budget, save, and spend monetary resources over time, taking into account various financial risks and future life events. When planning personal finances there are many financial products one might consider: such as banking products (checking, savings accounts, credit cards and consumer loans) or investment and insurance products (stock market, bonds, mutual funds) (life insurance, health insurance, disability insurance) or participation and monitoring of individual or employer sponsored retirement plans, social security benefits, and income tax management.

Chapter 2

2. Literature Survey

Agile methodologies are used to produce higher quality software in a shorter period of time. Agile methodologies were developed to streamline the development process and remove barriers to accepting business requirement changes during the development process. Agile methodologies do not require that business requirements and design details be locked in for the duration of development.

Dictionary meaning of Agile is moving quickly and lightly. It is a method that tries to be responsive to the needs of the software development process. Primary goal of agile methods is to develop working software.

Agile is a group of software development methodologies based on iterative and incremental development. Common theme about agile methodologies is that they are all focused on trying to produce a working solution and be able to respond to changing user/client requirement.[1]

2.1 The Agile Manifesto

1. Individuals and interactions over processes and tools.
2. Working Software over comprehensive documentation.
3. Customer collaboration over contract negotiation.
4. Responding to change over following a plan.

2.2 Principles

In February 2001, 17 software developers met at the Snowbird, Utah resort, to discuss lightweight development methods. They published the *Manifesto for Agile Software Development* to define the approach now known as agile software development. Some of the manifesto's authors formed the Agile Alliance, a nonprofit organization that promotes software development according to the manifesto's principles.

The basic principle of agile software development methodology is enlisted here:

- Highest priority is to satisfy the customer.
- Deliver working software frequently.
- Business people and developers must work together daily.
- Build projects around motivated individuals.
- Working software is the primary measure of progress.
- Sustainable development
- Continuous attention to technical excellence and good design enhances agility.
- Introspection – teams should regularly review itself and its processes to try and improve.[2]

2.3 Agile Methods

Agile methods have proven their effectiveness and are transforming the software industry. As agile methods evolve and extend, Agile Alliance fosters a community where organizations and individuals find ways to transition to and advance Agile practices, regardless of methodology. Agile Alliance organizes the largest, most diverse and comprehensive agile conferences each year. Agile methods have proven their effectiveness and are transforming the software industry. As agile methods evolve and extend, Agile Alliance fosters a community where organizations and individuals find ways to transition to and advance Agile practices, regardless of methodology. Agile Alliance organizes the largest, most diverse and comprehensive agile conferences each year.

The following are the core methods of Agile Software Development being worked on:

2.3.1 Agile Modelling
2.3.2 Extreme Programming (XP)
2.3.3 DSDM (Dynamic System Development Method)
2.3.4 SCRUM
2.3.5 FDD
2.3.6 CRYSTAL

2.3.1 Agile Modelling

It tries to find an appropriate balance between too little modelling and too much modelling, at the design stage. Modelled enough to explore and document your system effectively, but not so much that it becomes a burden. It provides guidelines on how to create effective models and how to

be an efficient modeller. It does not necessarily means that less modelling will be performed whiie adopting Agile Modelling.[3]

2.3.2 Extreme Programming (XP)

It was originally designed as a way of supporting small development teams working within uncertain and changing requirements. It was designed as an approach based on software engineering principles, but focused on the timely delivery of software that meets user's requirements. An important aspect of XP is the empowerment of the actual developers- they should be able to react immediately to changing customer requirements, even late in the development life cycle. It also places great emphasis on the software development team and teamwork. One of its aims is that team should communicate and constantly pay attention to all the details necessary to make sure that the software being developed matches the user requirements. The main focus of the extreme team is on giving the working software to the customer as soon as possible rather then wasting time in doing the documentation.[4]

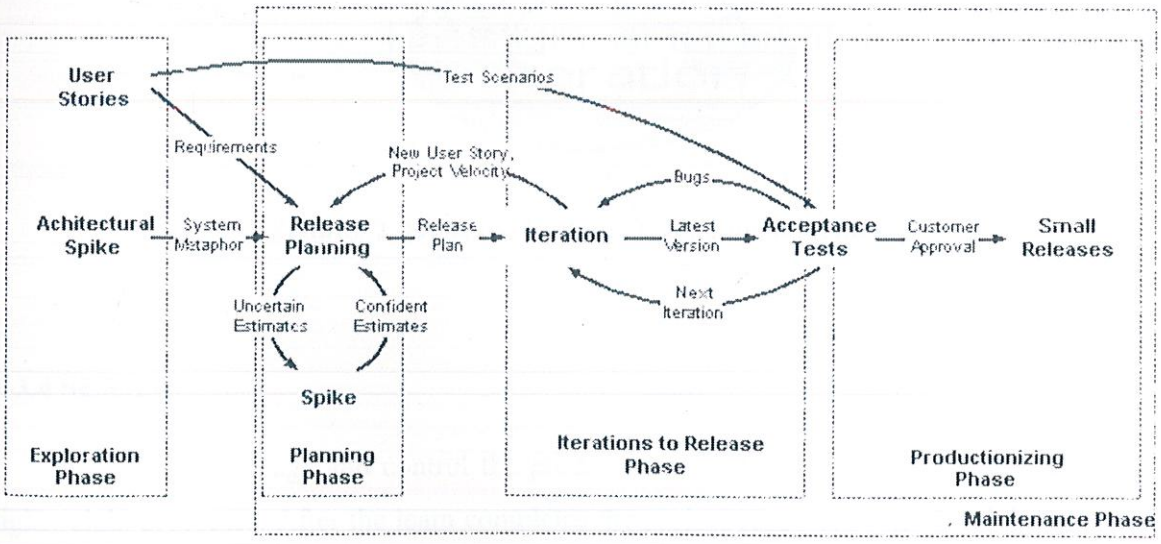


Figure2.1: Extreme Programming Lifecycle [5]

2.3.3 DSDM (Dynamic System Development Method)

It is particularly suitable for application development projects that need to develop complex business solutions within tight timeframes. In DSDM, time is fixed for the life of the project, and resources are fixed as far as possible.

It is based on following principles:

- Active user involvement is imperative.
- The team must be empowered to make decisions.
- The focus is on frequent delivery of products.
- Iterative and incremental development is necessary to converge on an accurate business
- All changes during development are reversible.
- Requirements are base lined at a high level.
- Testing is integrated throughout the life cycle.
- Collaboration and cooperation between all the stakeholders is essential. [7]

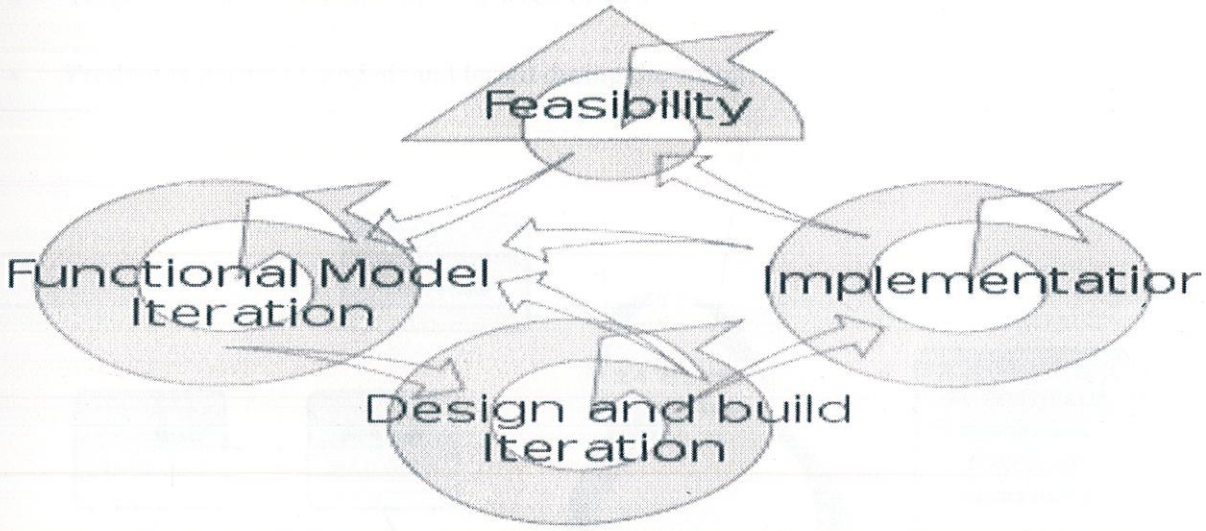


Figure2.2: Dynamic System Development method [6]

2.3.4 SCRUM

It aims to manage and control the production of software using iterative, incremental and lightweight processes. After the team completes the project scope and high-level designs, it divides the development process into a series of short iterations called 'sprints'. Each sprint aims to implement a fixed number of backlog items. At the end of a sprint, the team reviews the sprint to check progress. During a sprint, the team has a daily meeting called a scrum. Each team member describes the work to be done that day and progress from the day before. When enough of the backlog has been implemented so that the end users believe the release is worth putting into production, management closes development. The team then performs integration testing, training and documentation as necessary for product release. Every two weeks to a month anyone can see

real working software and decide to release it as is or continue to enhance for iteration. It allows us to rapidly and repeatedly inspect actual working software (every two weeks to one month). It has self-organizing teams. Product progresses in prints. Requirements are captured as backlog.[8]

Scrum has self organizing teams Product progresses in a series of month-long "sprints". Requirements are captured as items in a list of "product backlog .No specific engineering practices prescribed in scrum.

2.3.5 Sprints

- Scrum projects make progress in a series of "sprints" Analogous to XP iterations
- Target duration is one month +/- a week or two
- Product is designed, coded, and tested during the sprint.

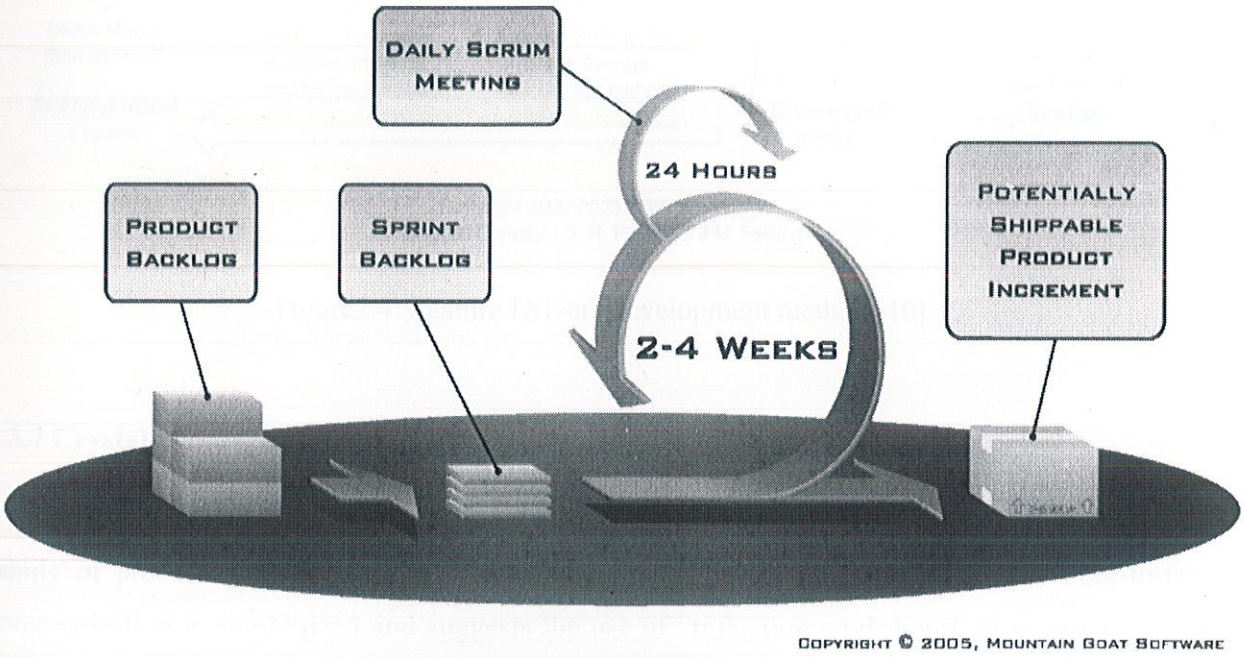


Figure2.3: Scrum Lifecycle [9]

2.3.6 Feature Driven Development

It starts with creation of a domain object model in collaboration with domain experts. Using information from modelling and from requirements activities the developers create a features list. Then a rough plan of development is drawn up and responsibilities are assigned. Then small

dynamically formed teams develop the features by repealtheduly performing design and build iterations that last no longer than 2 weeks.Originally proposed by Peter Coad .It is applied to an Iterative Development Process.

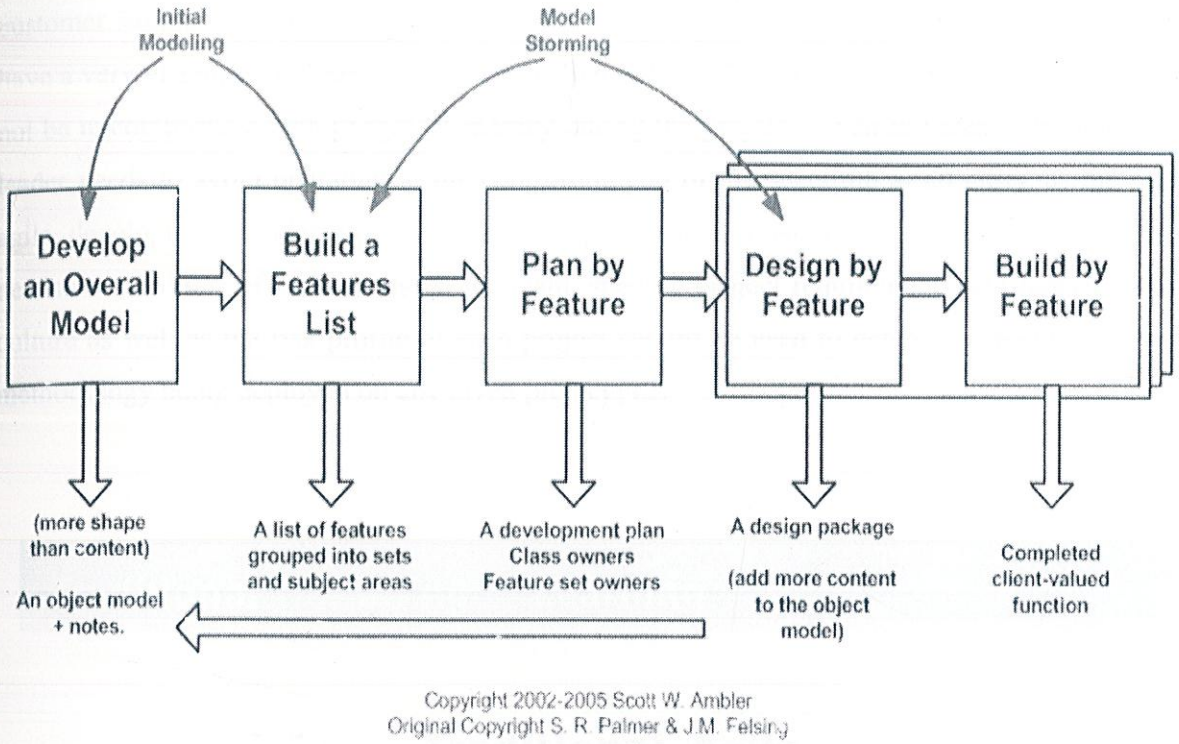


Figure2.4: Feature Driven Development method [10]

2.3.7 Crystal

Cockburn and High smith proposed it. Crystal has distinguishing features. It is Actually a family of process models that allow "manoeuvrability" based on problem. In this Face-to-face communication is emphasized and suggests the use of "reflection workshops" to review the work habits of the team.[11]

2.4 Agile v/s Traditional method

A significant differentiator between agile and traditional approaches is that each requires very different management styles. Traditional methodologies require more management and less leadership, whilst the agile approaches require more leadership and less management. As

a process-based approach, the traditional methodologies focus heavily on what is supposed to be done, in what order, with what inputs, processes and outputs, and, with a desired deliverable as overriding goal. On the other hand, the agile approaches are completely outcomes-based. The sole goal of the agile approaches is to deliver software that exceeds customer expectations with little or no concern with interim processes unless these processes add directly to the final outcome of customer satisfaction.On the other hand, the optimum leader of an agile approach project would have a very visionary outlook, would be directing at times (when needed) and would most certainly not be uncomfortable with phases of anarchy during the project – even in times of trouble an agile leader needs to avoid operating at the controlling end of the spectrum at all costs, as the basis of agile development leadership is courage, empowerment, trust and a focus on the customer's needs.Personalities of individuals on the team, specific project requirements and indeed corporate culture as well as the risk profile of each project should be used to determine the optimal type of methodology being deployed on any given project.[12]

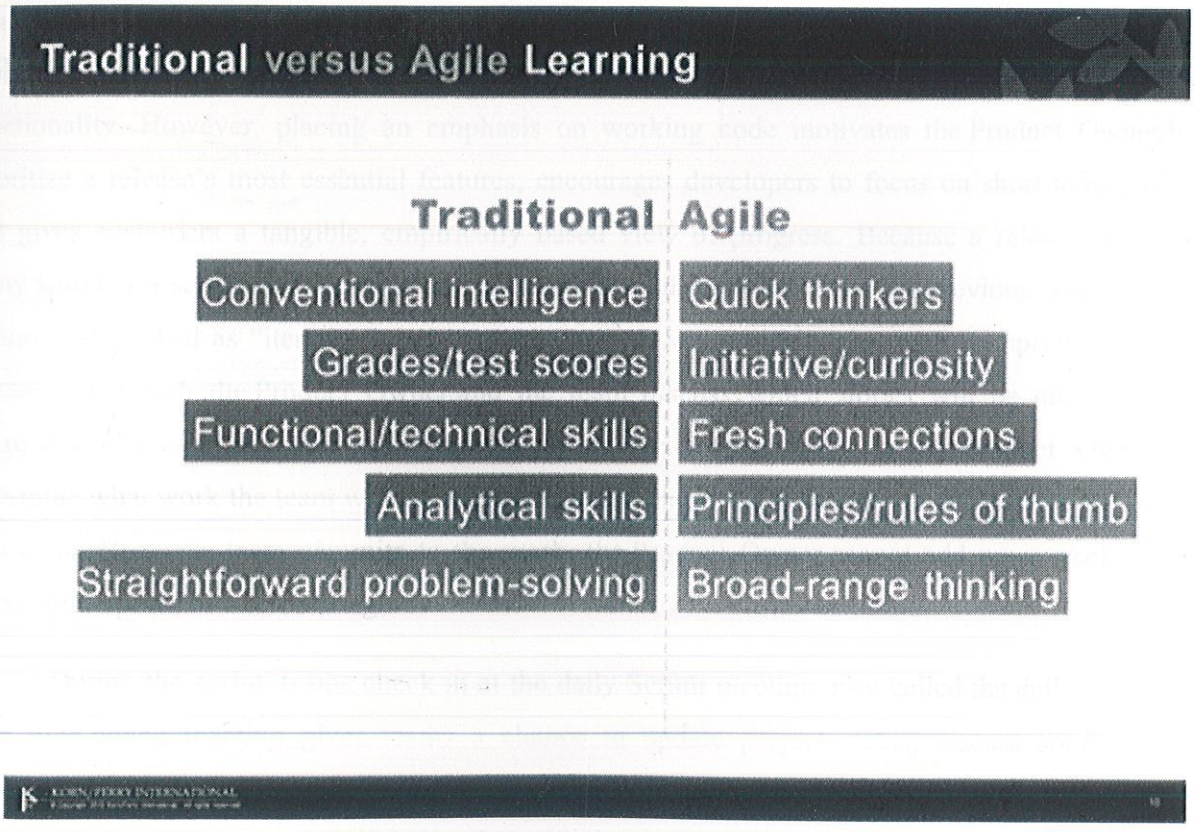


Figure2.5: Traditional v/S agile Development method [13]

Chapter 3

3. Implementation of SCRUM

3.1 Sprints

In agile we use a technique called sprint. In the Scrum method of agile software development, work is confined to a regular, repeatable work cycle, known as a sprint or iteration. In by-the-book Scrum, a sprint is 30 days long, but many teams prefer shorter sprints, such as one-week, two-week, or three-week sprints. But how long each sprint lasts is something for the team to decide, who must weigh the advantages or disadvantages of a longer or shorter sprint for their specific development environment.

The important thing is that a sprint is a consistent duration. During each sprint, a team creates a shippable product, no matter how basic that product is. Working within the boundaries of such an accelerated timeframe, the team would only be able to build the most essential functionality. However, placing an emphasis on working code motivates the Product Owner to prioritize a release's most essential features, encourages developers to focus on short-term goals, and gives customers a tangible, empirically based view of progress. Because a release requires many sprints for satisfactory completion, each iteration of work builds on the previous. This is why Scrum is described as "iterative" and "incremental". Every sprint begins with the sprint-planning meeting, in which the Product Owner and the team discuss which stories will be moved from the product backlog into the sprint backlog. It is the responsibility of the Product Owner to determine what work the team will do, while the team retains the autonomy to decide how the work gets done. Once the team commits to the work, the Product Owner cannot add more work, alter course mid-sprint, or micromanage.

During the sprint, teams check in at the daily Scrum meeting, also called the daily standup. This time-boxed meeting gives teams a chance to update project status, discuss solutions to challenges, and broadcast progress to the Product Owner (who may only observe or answer the team's questions).Just as every sprint begins with the sprint planning meeting, the sprint concludes with the sprint review meeting, in which the team presents its work to the Product Owner. During this meeting, the Product Owner determines if the team's work has met its acceptance criteria. If a single criterion is not met, the work is rejected as incomplete. If it satisfies the established criteria,

then the team is awarded the full number of points. Because certain sprints are hugely successful and others less than ideal, a team also gathers at the end of each sprint to share what worked, what didn't, and how processes could be improved. This meeting is called the sprint retrospective meeting.

Personal Financial Management System will have following work to be done:

- Finance Management should have a login screen.
- Expenditure Module which should include:
    - expenses which is related general operation of business
    - short period expenses(shares, debentures)
    - Long period expenses (property purchase, vehicle or jewellery purchase)
    - Expenses of recurring nature.
    - Expenditure on insurance facilities, e.g. - Health Insurance, Car Insurance.
- Income Module which should have monthly income through any source via- any govt. or private sector service business selling of shares (if it went for profit)
- Additional income sources
- Daybook module which contains amount spent on day to day activities:
    - Electricity bill and telephone bills
    - Food items
    - Movable and immovable items
    - Hospital bills
    - Fuel
- Income tax module that calculates income tax for salaried employee, woman and senior citizen.
- View module that recollects the whole data which user has entered and then display it.
- Reminders Module which Analyzes if the user wants anything to be reminded
- Help module that gives directions on how to use the system.

3.2 Architectural Design

- **Two TIER CLIENT SERVER SYSTEM**

In Financial Management System, we need a client and a server. Everything will be handled on the server side. So we can say that it is a Two Tier Thin Client Server System. Design is shown in fig 3.1.

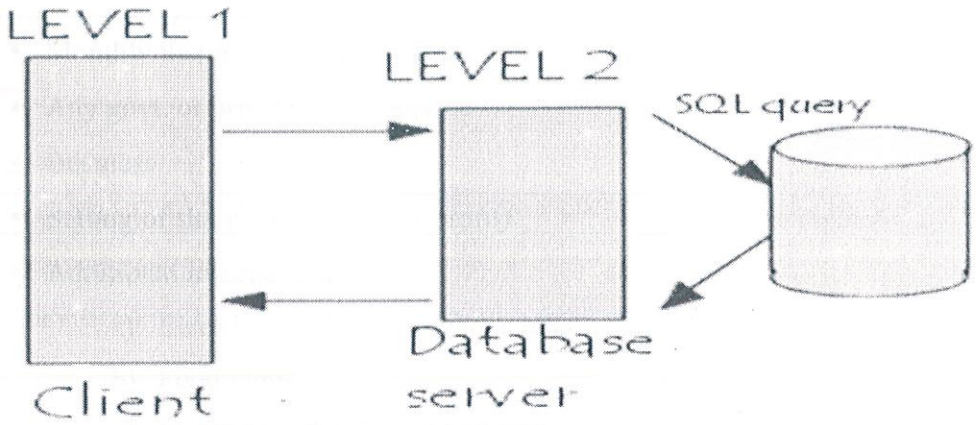


Fig 3.0: Architectural Design[14]

3.3 Planning

Planning is an important phase in developing of software. Since we are using agile methodology scrum we have a set defined planning technique. Product will be developed in iterations using sprints which are explained as here. The Project will be completed in:

Sprint 1:

In this sprint, we will complete following:

- Create a login panel.
- User inputs his credentials.
- Will be directed to main system.

Sprint 2:

In this sprint, we will complete expenditure module where user inputs:

- expenses which is related general operation of business
- short period expenses(shares, debentures)
- Long period expenses (property purchase, vehicle or jewellery purchase)
- Expenses of recurring nature.
- Expenditure on insurance facilities, e.g. - Health Insurance, Car Insurance.

Sprint 3:

In this sprint, we will complete Income module and daybook where user interacts:

- Monthly income through any source.
- Any govt. or private sector service
- Business
- Selling of shares (if it went for profit)
- Additional income sources
    a) Electricity bill and telephone bills
    b) Food items
    c) Movable and immovable items
    d) Hospital bills
    e) Fuel

Sprint 4:

In this sprint we will cover income tax module where one can calculate has tax to be paid.

Sprint 5:

View module where one can view various inputs made earlier through income, expenditure, daybook etc.

Sprint 6:

Covering all remaining modules that are future plan, reminders, password manager, help.

3.4 UML Diagrams

3.4.1Use case diagram:

Use case diagram share important for visualizing, specifying, and documenting the behaviour of an element. use case diagrams overview the usage requirements for a system. They are useful for presentations to management and/or project stakeholders, but for actual development you will find that use cases provide significantly more value because they describe "the meat" of the actual requirements.

Use case diagrams commonly contain:

– Use cases

–Actor, dependency, generalization, and association relationships

Income: use cases are monthly income, income through selling of property or assets, additional income sources and selling of shares.

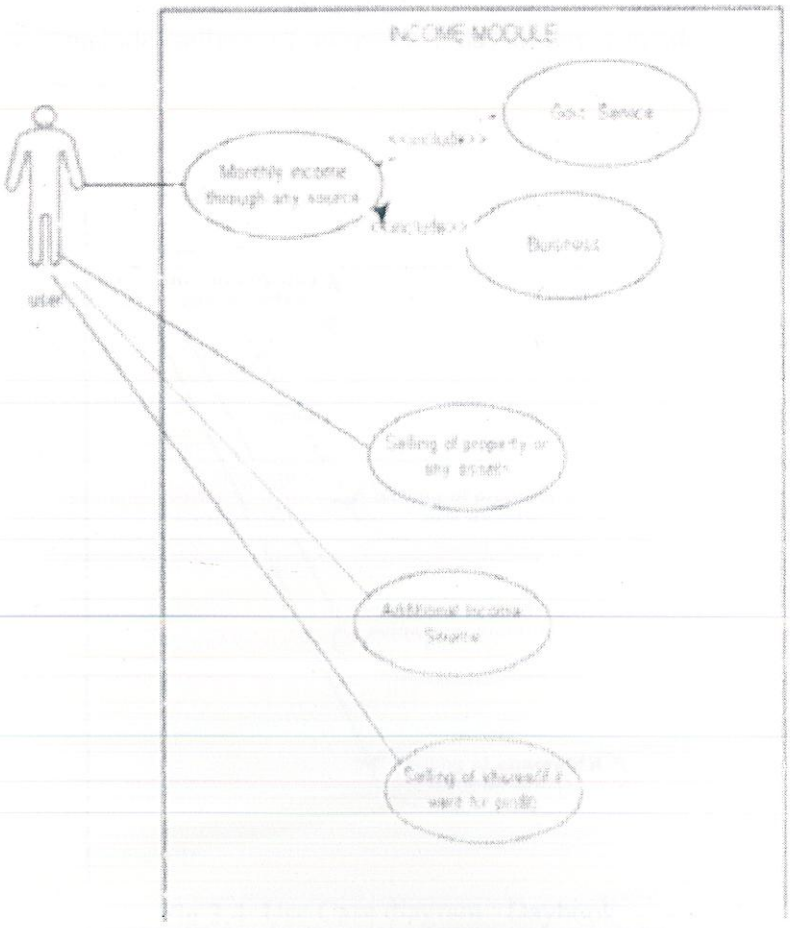


Fig 3.1: Use Case diagram: Income

Expenditure: Use cases are periodic expenses, recurring expenses, and surplus expenses.

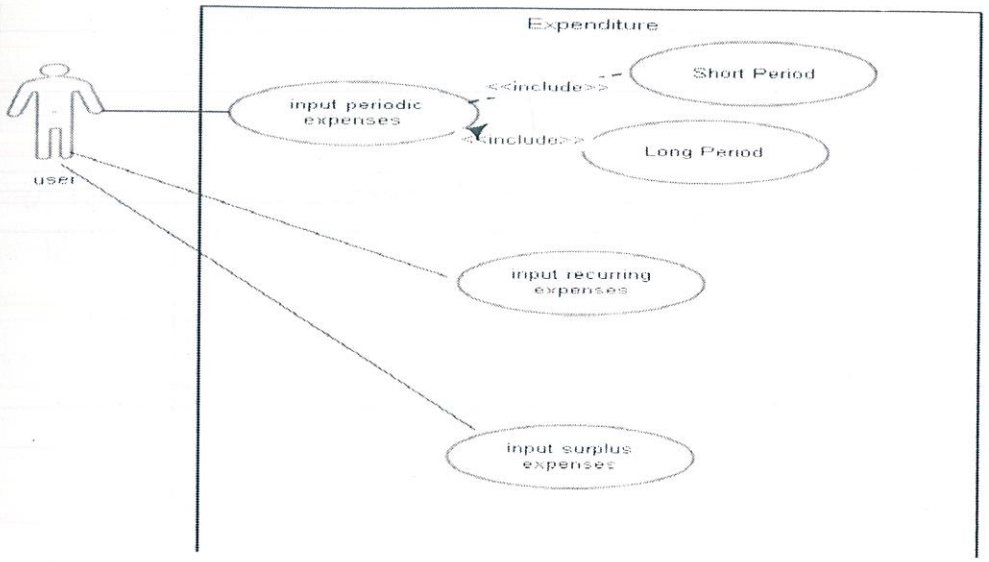


Fig 3.2: Use Case diagram: Expenditure

Daybook: Use cases include selling of property , day-to-day activates via telephone bills, electricity.

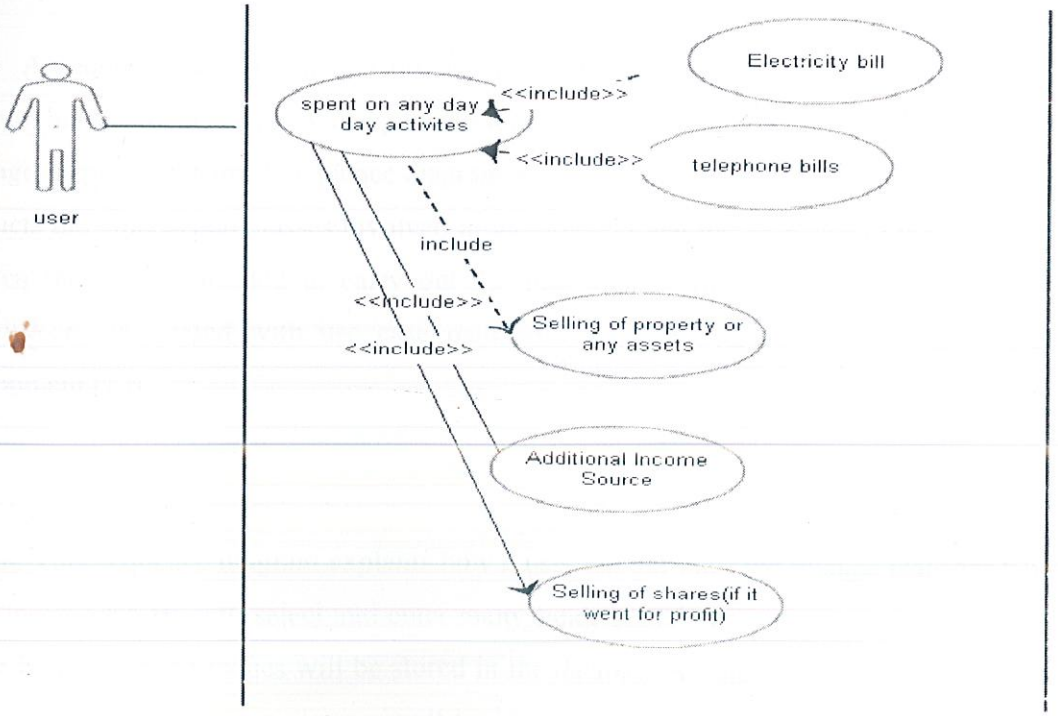


Fig 3.4: Use Case diagram : Daybook

basic diagram of income case which is in the system.It contains Finance Manager, income button, system and database as predicted here.User will click on income button whih will guide him to the income field where he will input all the details.

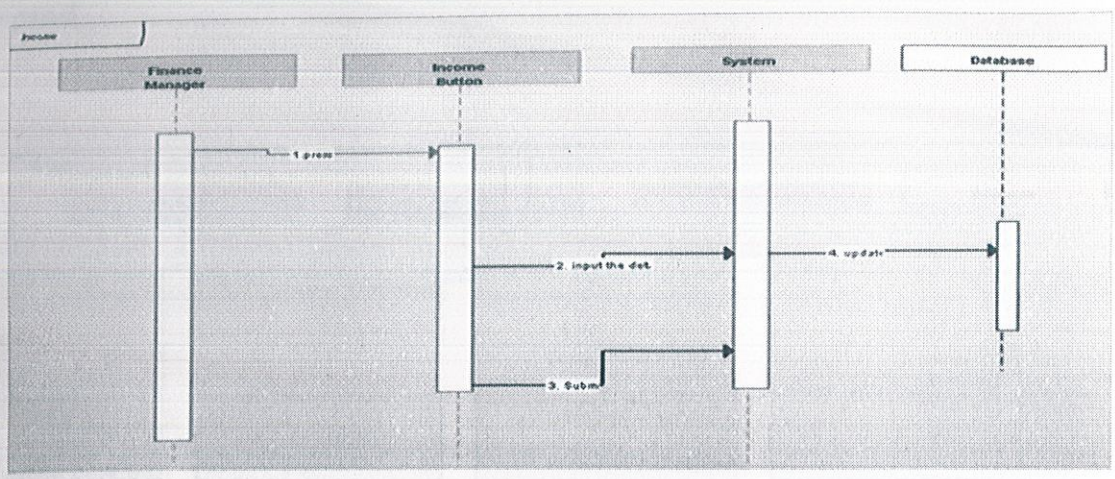


Fig 3.6: Sequence diagram: income details

Expenditure:This sequence diagram explains how a user will get into the finance manager tab and will select expenditure tab.He will select and enter many fields containing various inputs.He will select the submit button and the values will be stored in the database.At later time they cn be retrieved from view panel. So the sequence diagram will be in simple steps as depicted in this diagram.This is a basic diagram of income case which is in the system.It contains Finance Manager, income button, system and database as predicted here.User will click on expenditure button whih will guide him to the income field where he will input all the details.

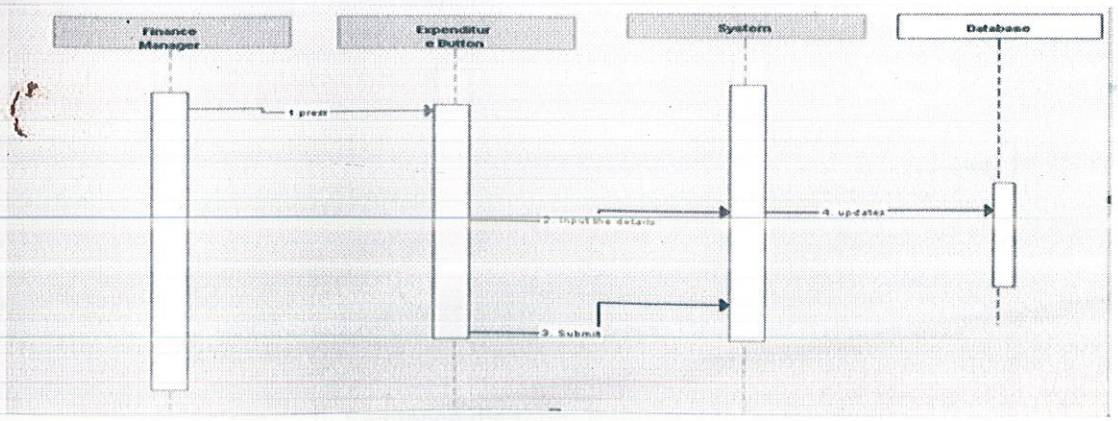


Fig 3.7: Sequence diagram: expenditure details

Daybook: This sequence diagram explains how a user wil get into the finance manager tab and will select daybook tab.He will select and enter many fields containing various inputs.He will select the submit button and the values will be stored in the database.At later time they cn be retrieved from view panel.

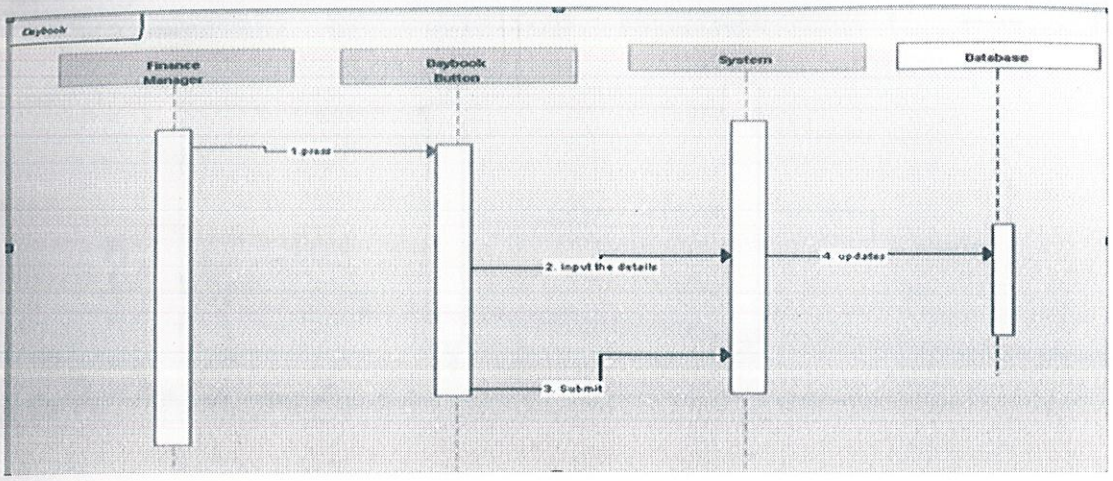


Fig 3.8: Sequence diagram: Daybook details

Reminders: This sequence diagram explains how a user wil get into the finance manager tab and will select reminders tab.He will select and enter many fields containing various inputs.He will select the submit button and the values will be stored in the database.At later time they cn be retrieved from view panel.

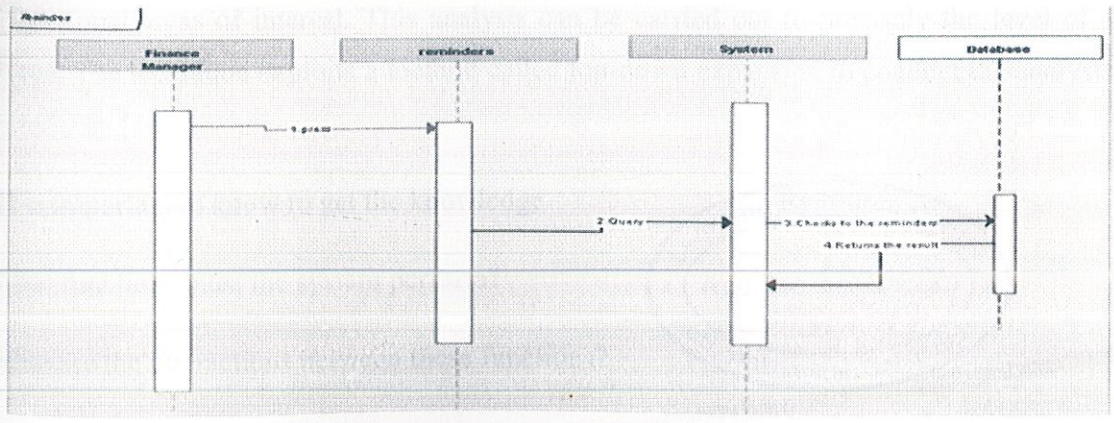


Fig 3.9: Sequence diagram: Reminder input by user

Future plans: This sequence diagram explains how a user wil get into the finance manager tab and will select future plans tab.He will select and enter many fields containing various inputs.He will select the submit button and the values will be stored in the database.At later time they cn be retrieved from view panel.

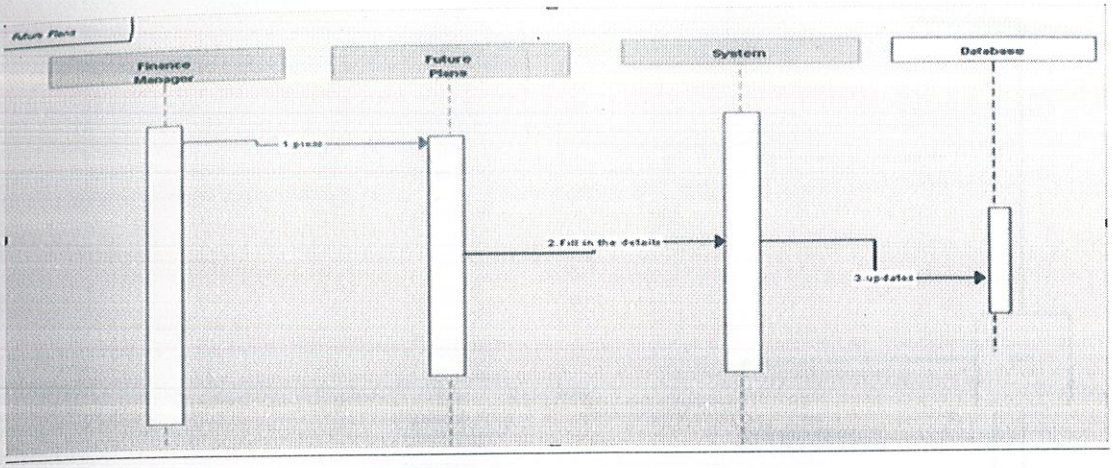


Fig 3.10: Sequence diagram: Future Plan input by user

3.4.3. Data Flow Diagram

Data flow diagrams can be used to provide a clear representation of any business function. The technique starts with an over all picture of the business and continues by analysing each of the functional areas of interest. This analysis can be carried out to precisely the level of detail required. The technique exploits a method called top-down expansion to conduct the analysis in a targeted way.[16]

DFD is important to know to get the knowledge

- What functions must the system perform?
- What are the interactions between these functions?
- What transformations must the system carry out?
- What inputs are transformed into what outputs?

- What kind of work does the system do?
- Where does it get the information to do its work?
- Where does it deliver the results of the work?

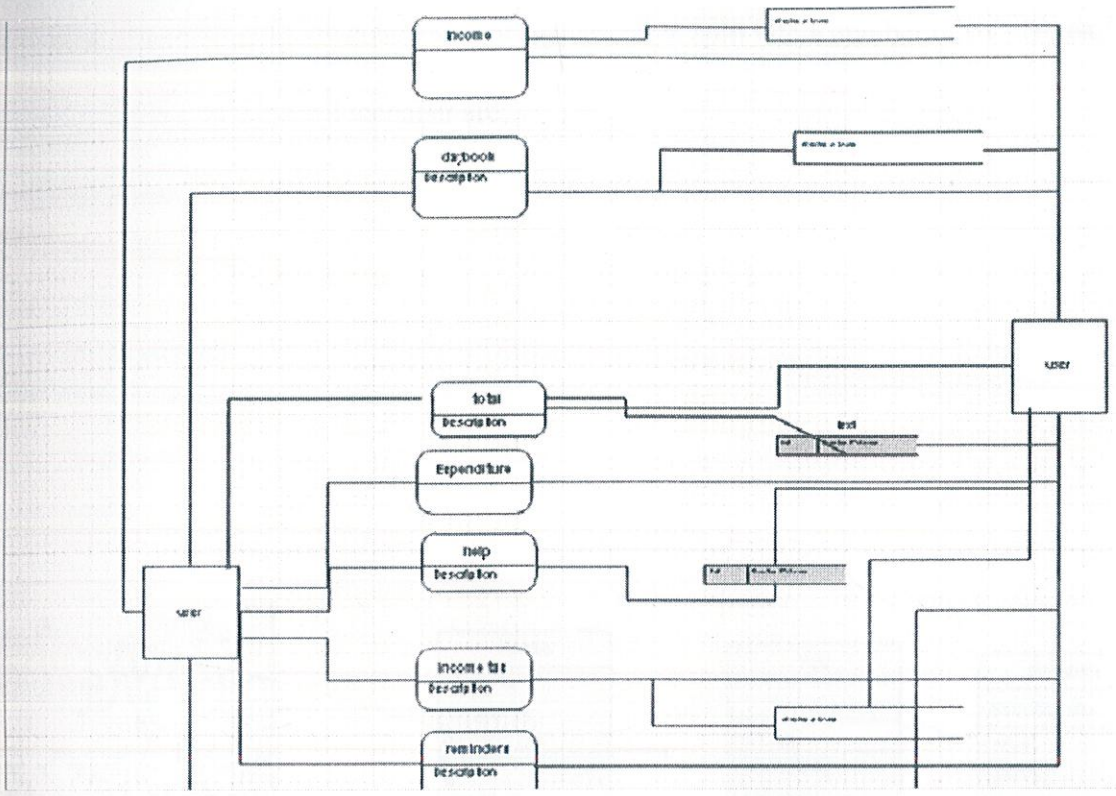


Fig 3.10: Data Flow Diagram: Financial Management System

3.4.4 Class Diagram

The class diagram is the main building block in object oriented modelling. It is used both for general conceptual modelling of the systematics of the application and for detailed modelling translating the models into programming code. The classes in a class diagram represent both the main objects and or interactions in the application and the objects to be programmed. In the class diagram these classes are represented with boxes which contain three parts.[17]

A class with three sections:

- The upper part holds the name of the class.
- The middle part contains the attributes of the class.
- The bottom part gives the methods or operations the class can take or undertake.

In the system design of a system, a number of classes are identified and grouped together in a class diagram which helps to determine the static relations between those objects. With detailed modelling. The classes of the conceptual design are often split into a number of subclasses.

The basic classes of financial manager are:

- Daybook
- Income
- View
- Expenditure
- Future Plans
- View
- Help

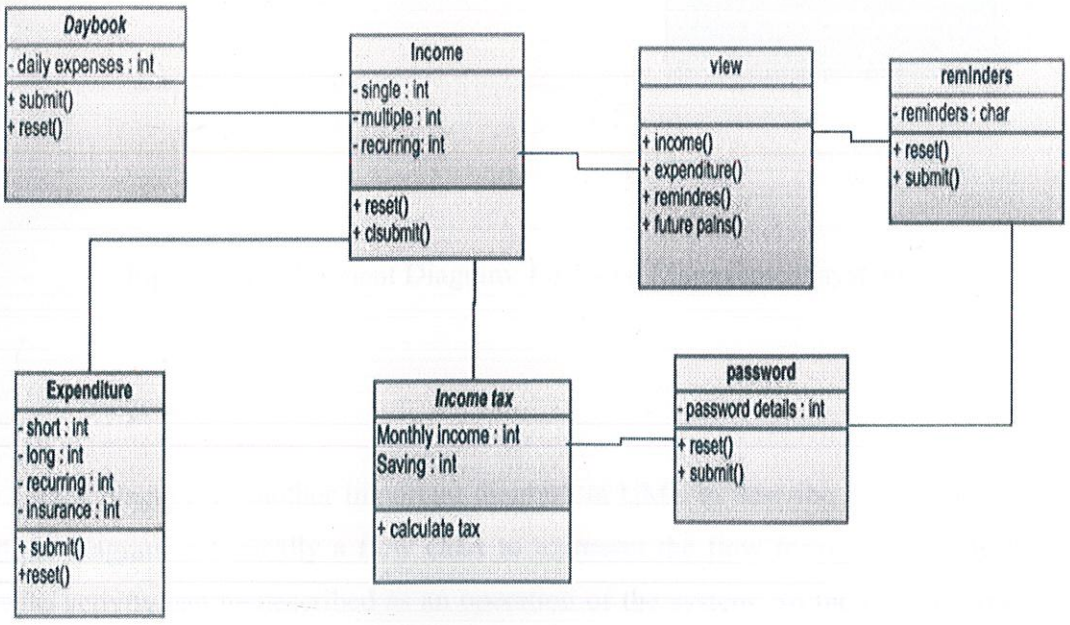


Fig 3.11: Class Diagram: Financial Management system

3.4.5 Deployment Diagram

A UML 2 deployment diagram depicts a static view of the run-time configuration of processing nodes and the components that run on those nodes. In other words, deployment diagrams

show the hardware for your system, the software that is installed on that hardware, and the middleware used to connect the disparate machines to one another. You want to create a deployment diagram for applications that are deployed to several machines, for example a point-of-sales application running on a thin-client network computer which interacts with several internal servers behind your corporate firewall or a customer service system deployed using a web services architecture such as Microsoft's .NET. Deployment diagrams can also be created to explore the architecture of embedded systems, showing how the hardware and software components work together. In short, you may want to consider creating a deployment diagram for all but the most trivial of systems.[18]

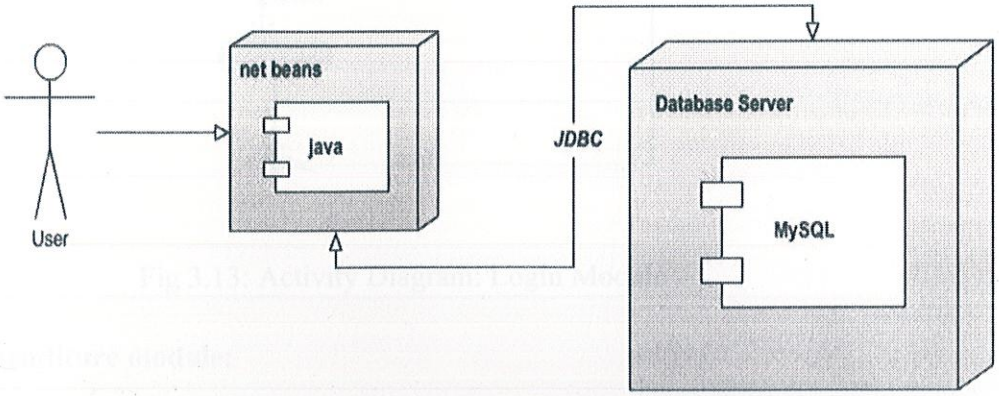


Fig 3.12: Deployment Diagram: Financial Management system

3.4.6 Activity Diagram

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. This diagram is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deals with all type of flow control by using different elements like fork, join etc.[19]

- **Login**

This activity diagram explains how the login module works. First a user enters his

credentials. If it found to be correct then he will be directed to finance manager home screen otherwise he will have to enter again.

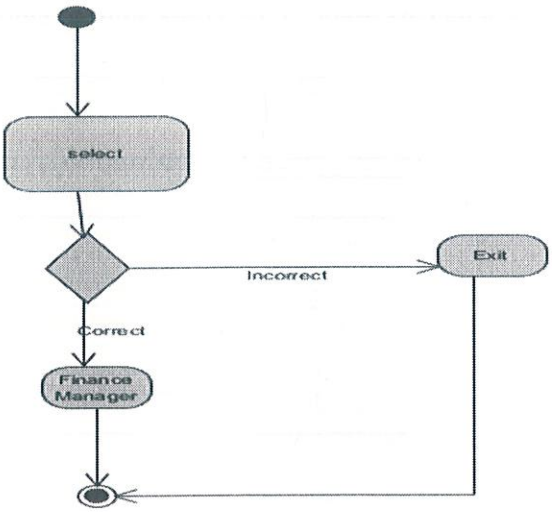


Fig 3.13: Activity Diagram: Login Module

- **Expenditure module:**

This activity diagram explains how an expenditure module works.User will select expenditure tab from finance manager home screen.After than he will be direcyed to expenditure module where he will enter various fields which will be stored in database.He can also reset and reenter.

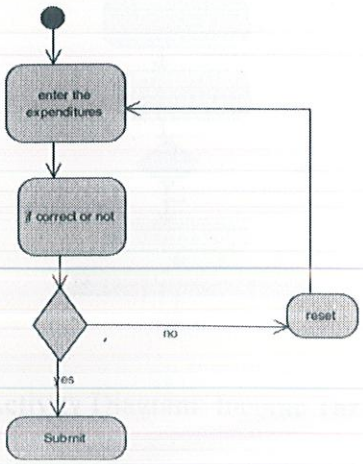


Fig 3.14: Activity Diagram: Expenditure Module

- **Income Module:**

This activity diagram explains how an income module works.User will select income tab from finance manager home screen.After than he will be direcyed to expenditure module where he will enter various fields which will be stored in database.He can also reset and reenter.

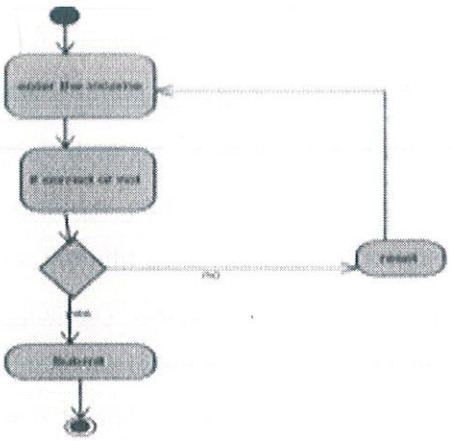


Fig 3.15: Activity Diagram: Income Module

- **Income Tax Module:**

This activity diagram explains how one will calculate income tax. It is simple and easy. User will just have to monthly income and charity plus savings and finance manager will calculate your income tax.

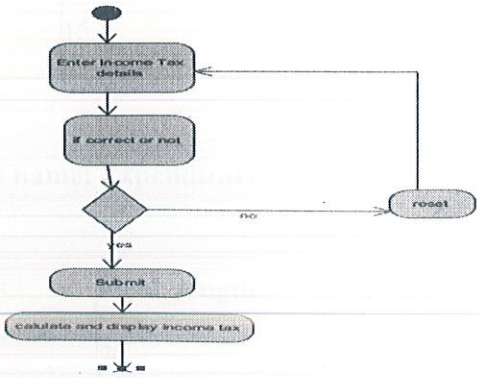


Fig 3.16 Activity Diagram: Income Tax Module

3.5 Repository Design

This design gives us basic details of how our database will work. The database name is finance management. It contains various tables which are described below. These tables have various entities and primary keys. Some fields are integer some are varchar, some are float etc. **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. The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data. 20].So the repository design is a simple one pertaining a few tables. In these tables most of the entries are in integer as income is always a integer field. Some are varchar.The database used is mysql and the name is finance management.

Table name: Income

Rowname	Datatype	Field_length	Is primary key
Income I.d	integer	20	Yes
Monthly income	integer	20	No
Income through other source	integer	12	No
Date/Month/year	integer	15	Yes

Table name: Expenditure

Rowname	Datatype	Field_length	Is primary key
Expenditure I.d	integer	20	Yes
Short Term	integer	20	No
Long Term	integer	12	No
Recurring	Integer	12	No

Table name: Daybook

Rowname	Datatype	Field_length	Is primary key
Daybook I.d	integer	20	Yes
Home	integer	20	No
Grocery	integer	12	No
Dining	Integer	20	No
Health	Integer	20	No
Materials	Integer	20	No
Entertainment	Integer	20	No
Miscellaneous	Integer	20	No
Date/Month/year	integer	15	Yes

Table name: Reminders

Rowname	Datatype	Field_length	Is primary key
Reminder I.d	Integer	20	Yes
Reminder	Varchar(45)	45	No
Date/Month/ye	Integer	15	Yes

Table name: Future Plans

Rowname	Datatype	Field_length	Is primary key
Future Plan I.d	Integer	20	Yes
Future Plan	Varchar(45)	45	No
Date/Month/year	Integer	15	Yes

Table name: Password

Rowname	Datatype	Field_length	Is primary key
Password I.d	Integer	20	Yes
Utility	Varchar(45)	45	No
Password	Varchar(45)	45	No
Details	Varchar(45)	45	No
Date/Month/year	Integer	15	Yes

Chapter 4

4. Conclusion and Future Scope

4.1 Conclusion

Finance Management System is developed as a part of the degree completion process in the fourth year. It was really tricky initially as the group was naïve when it came to the implementation of the Agile Methodology of the software development process..The task in hand was to develop the system and thereby to widen our horizons in the field of Agile Methodologies. The best practice was to go step by step and sticking to the basics learnt in the past 3 years which really worked well for the group. After going through a lot of research, there came a belief that we can implement Scrum methodology on our project, Finance Management System.

After completing this project, we can proudly declare that we have learned how to go for a project systematically by first analysing the things around us and learning from the mistakes to give the best possible output. We hope that it will definitely help us in future and help us to reach heights in our professional careers

The skills learned in through this project can be summarized as:

- Scrum which is one of the Agile Methodologies to develop a project.
- How to work on JAVA Swings and how different components in that can be used as and when required.
- JAVA classes.
- Had a very good experience of working on Netbeans IDE 7.0.
- Mysql in which database of the project is created and managed.
- JDBC connectivity to the Mysql Express Edition.

4.2 Future Scope

Finance Management System is a very vast project. In this project, we have compieted the work related to managing of personal household budgeting for individual as well as families. In future, the project can be extended by including some more entities like generation of reports and

graphs on tracking the spending and income. It can also guide user about how to spend the money efficiently and do proper budgeting so that you can save your money for future while leaving a good present life.At the beginning of the year, you have to enter your annual budgeted expenses per month under thirteen main expenditure categories vise: food and groceries, vehicle fuel and maintenance, electricity, clothes, education, rent, travel, conveyance, entertainment, telephone, medical, taxes, and miscellaneous. You feed in your daily expenditure for these categories. Every time you feed in your expenses, the Expense Record Book calculates category wise total expenditure for the month till date, category wise total expenditure for the year till date, and a variance report which shows you where you have overspent and/or under spent - for which month and under which category.

Appendix A

Codes

1) FM.java

```
public class FM {
public static void main(String[] args) {
LoginFrame frame=new LoginFrame();
Frame.setBounds(400,400,400,400);
frame.setVisible(true);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}}
```

2) Login.java

```
public class Login
{
   public ArrayList<String> matchLoginDetails(String str[]){ }
}
```

3) FinanceManagementGUI.java

```
public class FinanceManagementGUI
{
   public void incomeGUI() { }
   public void expenditureGUI() { }
   public void daybook GUI { }
   public incometaxGUI()  { }
   public void helpGUI() { }
```

```
        public void viewGUI() { }

        public void remindersGUI { }

        public futureplansGUI()  { }

        public void helpGUI() { }

        public void remindersGUI() { }

}
```

Appendix B

Login Screen

Input:Username

Input:Password

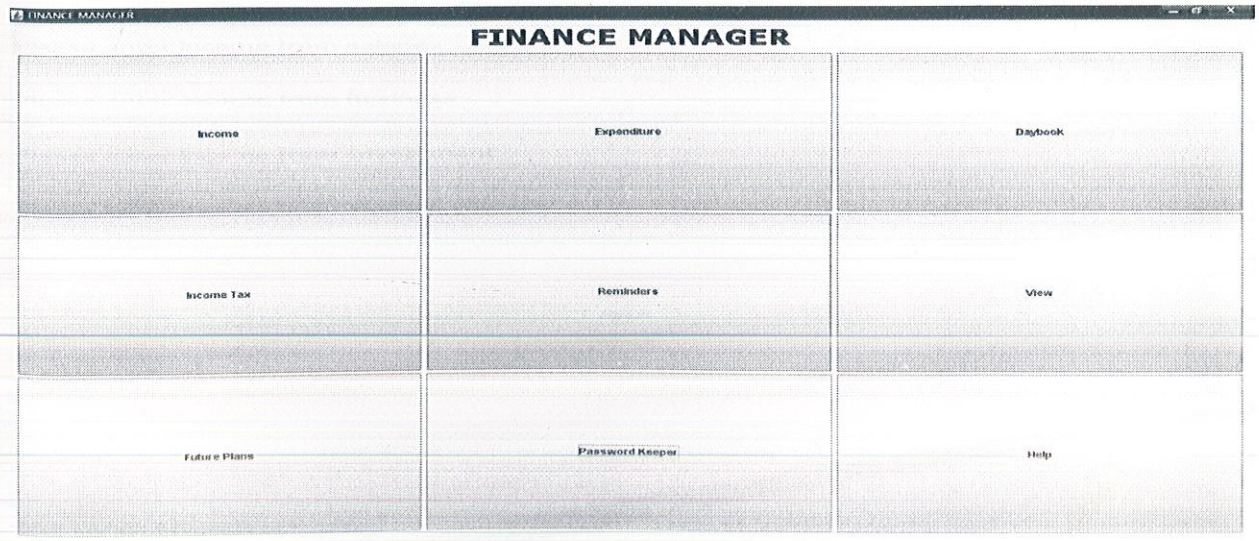
Output:Finance Manager Home Screen

Snapshot 1: Login Screen

Finance Manager:

Input:Select From any tab

Output:will be directed to the respective field

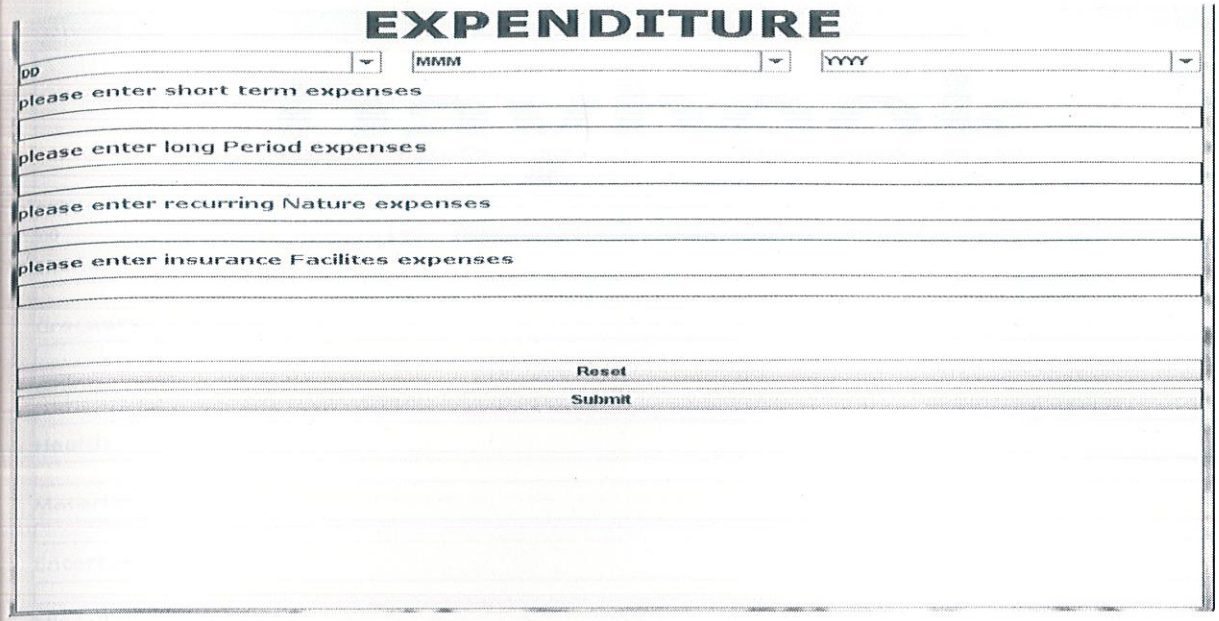


Snapshot 2: Finance Manager

Expenditure

Input:date,Month,Year,various fields

Output:The inputs will be saved in database.

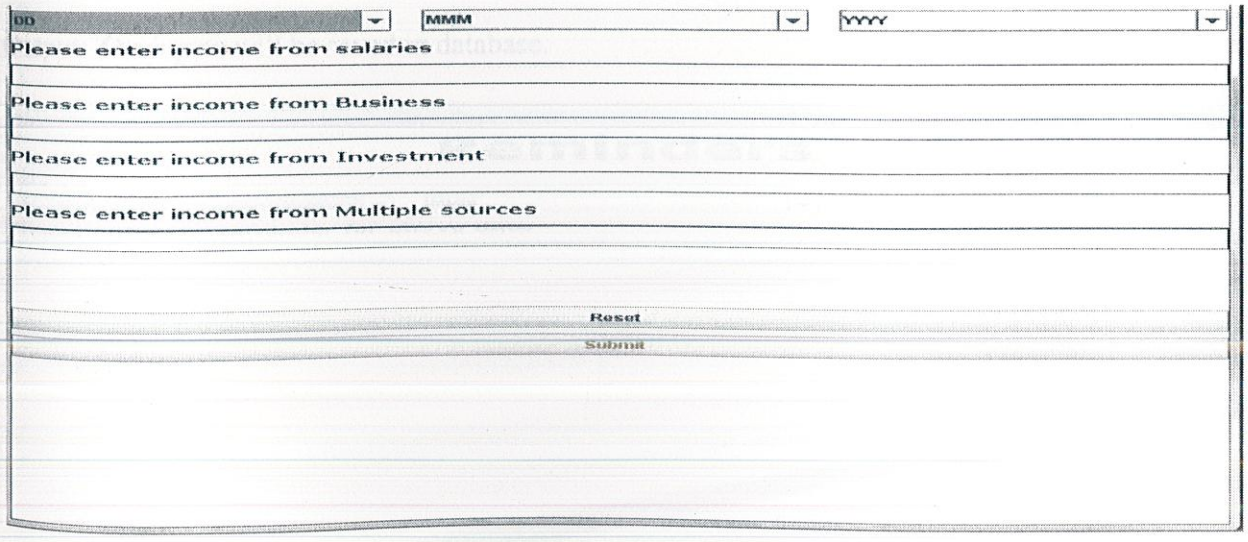


Snapshot 3: Expenditure Screen

Income

Input:date,Month,Year,various fields

Output:The inputs will be saved in database.

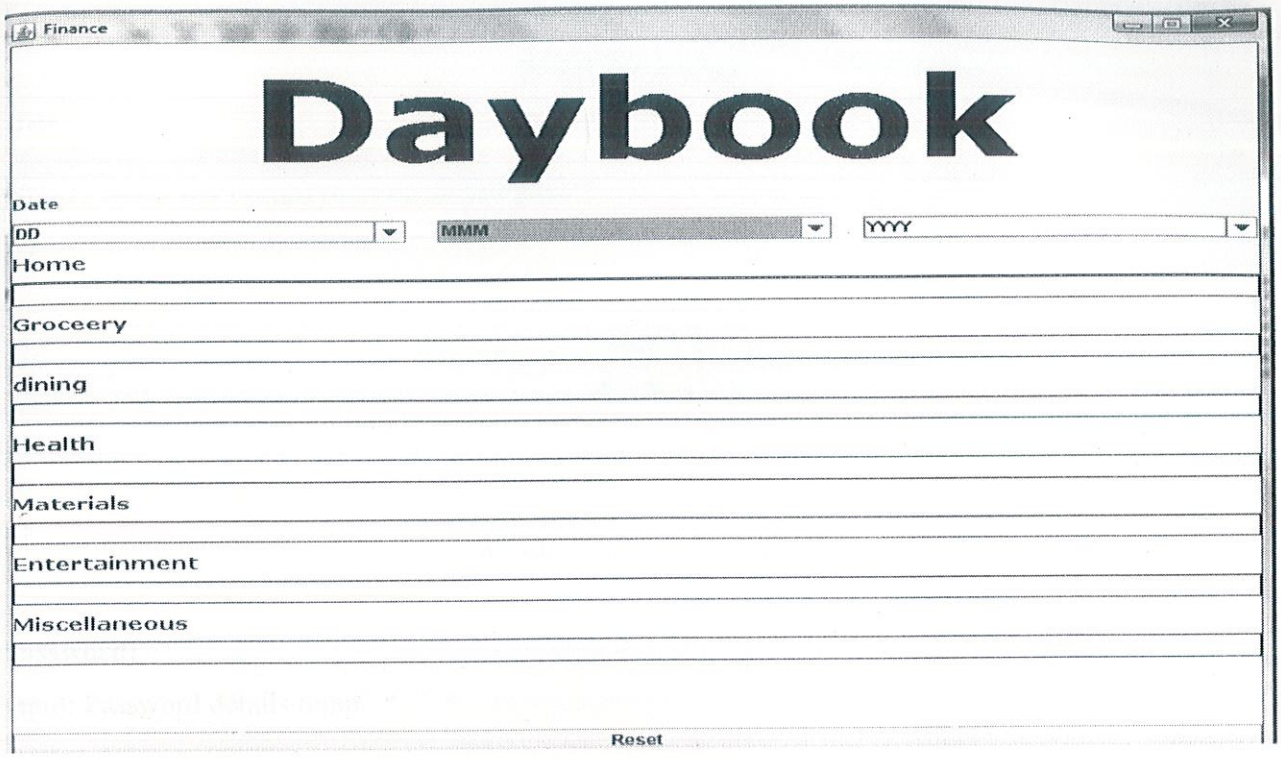


Snapshot 4: Income Screen

Daybook:

Input: date, Month, Year, various fields

Output: The inputs will be saved in database.

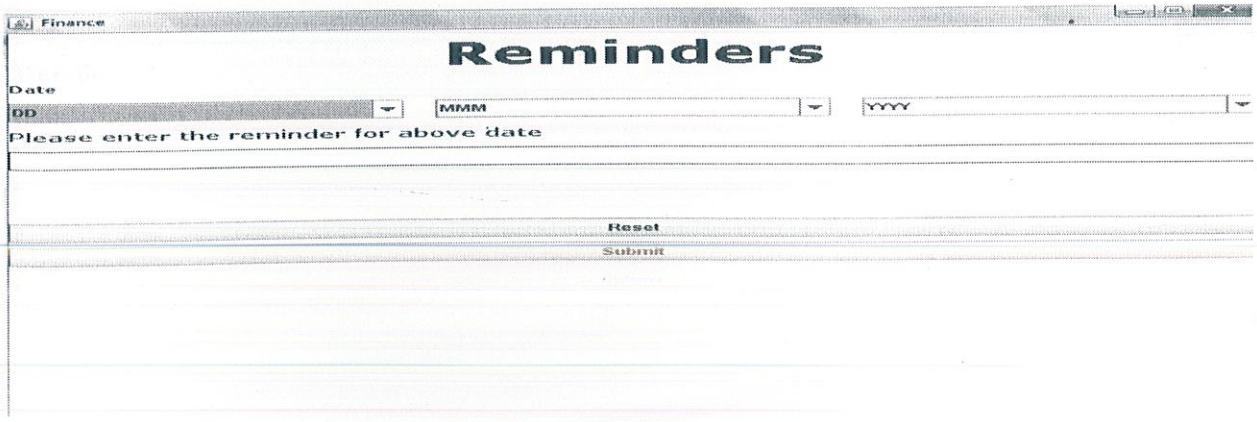


Snapshot 5: Day book Screen

Reminders:

Input: date, Month, Year, various fields

Output: The inputs will be saved in database.

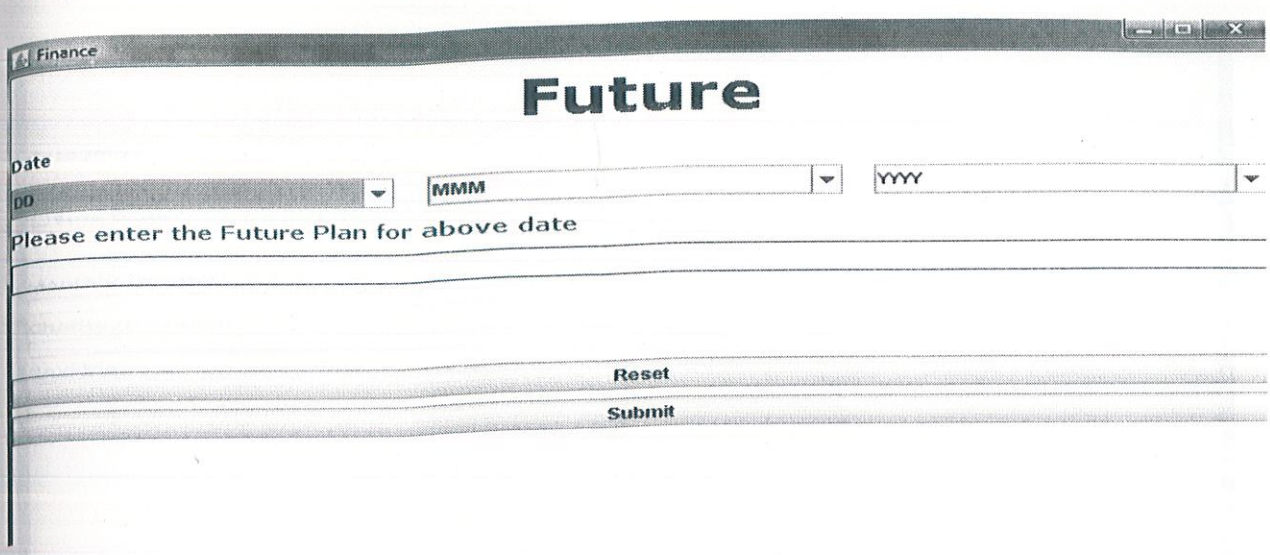


Snapshot 6: Reminders Screen

Future

Input: date, Month, Year, various fields

Output: The inputs will be saved in database.

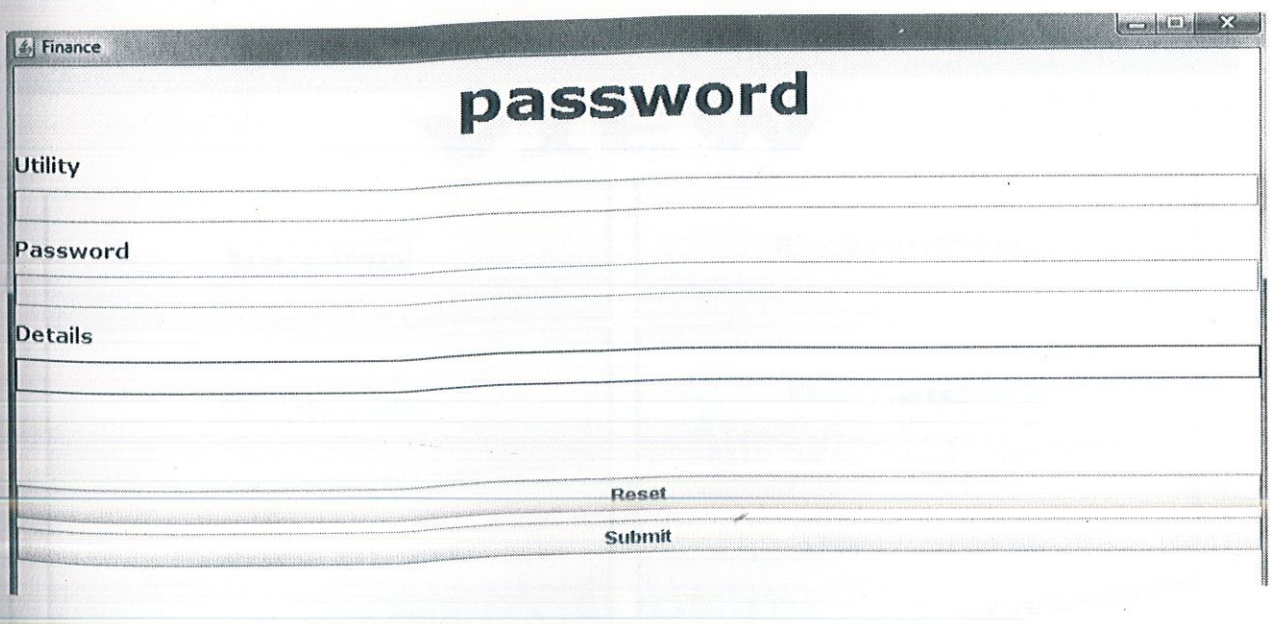


Snapshot 7: Future Screen

Password:

Input: Password details input.

Output: The inputs will be saved in database.

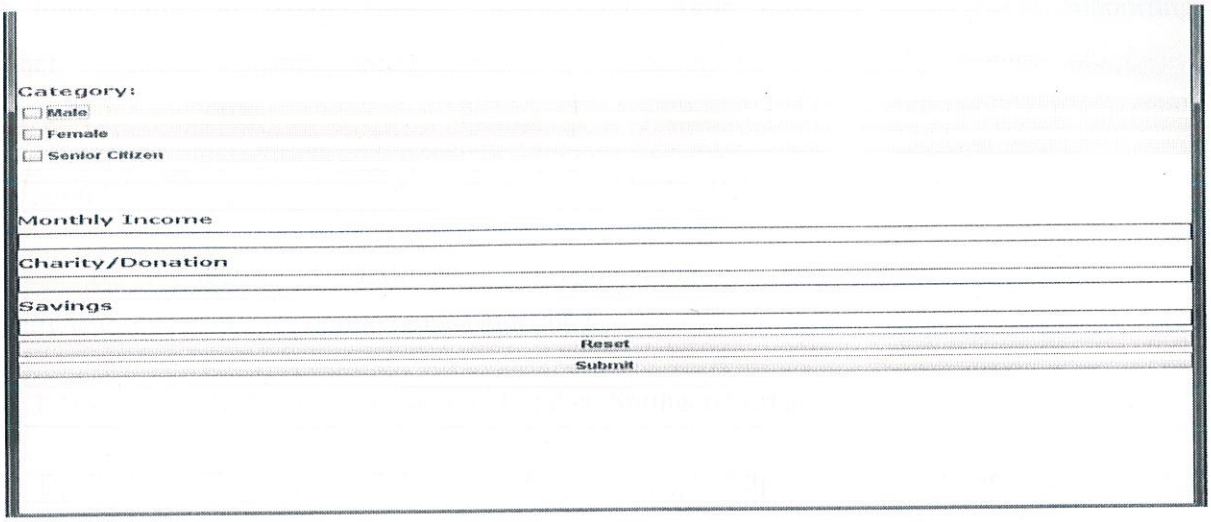


Snapshot 8: Password Screen

Income Tax

Input: Monthly income, Savings, charity.

Output: Calculation of income tax.

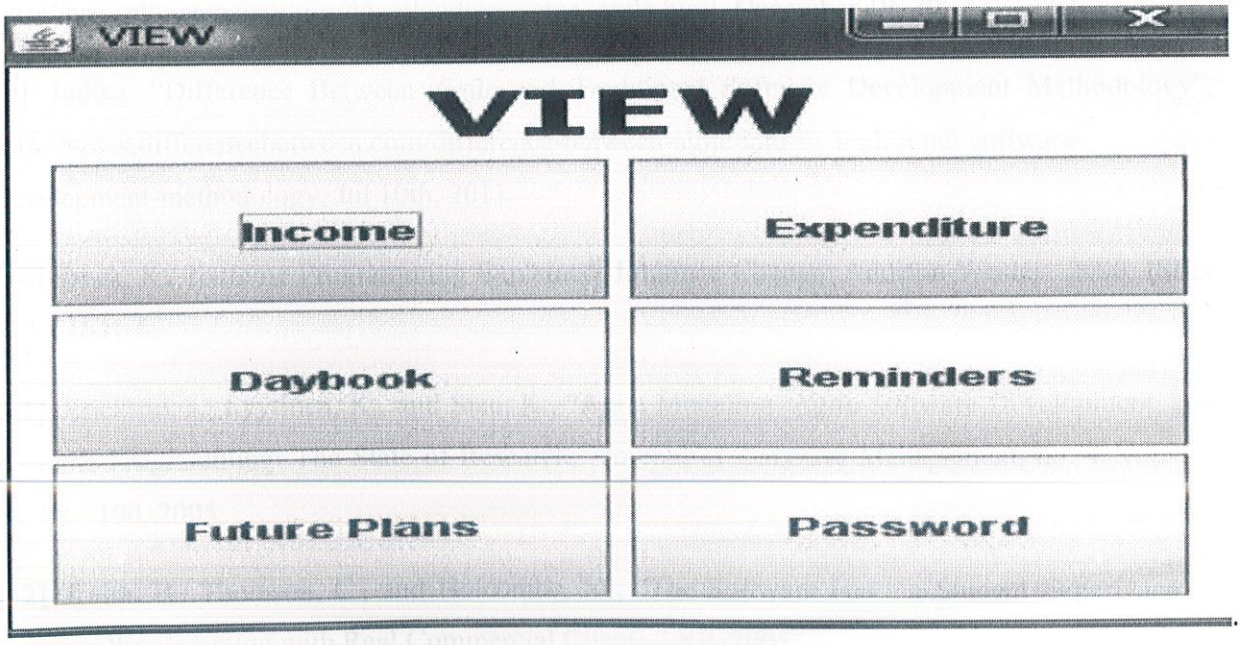


Snapshot 9: Income tax Screen

View

Input: user will select from various tabs.

Output: viewing of the saved inputs



Snapshot 10: View Screen

References

[1]. Jeffrey A. Livermore, "Factors that Significantly Impact the Implementation of an Agile Software Development Methodology", Journal of Software, Vol. 3(4), pp. 31-36, April 2008

[2]. Boris Matijaševi, Hrvoje Ronevi, Ognjen Orel, "Agile Software Development Supporting Higher Education Reform", International Conference on Information Technology Interfaces, pp.375-380, June 2007

[3]. Scott W. Ambler, "Agile Database Techniques", Wiley Publishing Inc, 2003.

[4]. D. Wells, "Extreme Programming: A gentle introduction", http://www.extremeprogramming.org, March 2007

[5]. J. Hunt, Agile Software Construction, London: Springer-Verlag press, 2006

[6]. D. Wells, "The Rules of Extreme Programming", http://www.extremeprogramming.org, September 28, 2009

[7]. Sheetal Sharma, Darothi Sarkar, Divya Gupta, "Agile Processes and Methodologies: A Conceptual Study", International Journal on Computer Science and Engineering, Vol. 4(5), pp. 892-898, May 2012

[8]. D. Laurance, D. Mancl, "Extreme Programming and Agile Processes", http://princetonacm.acm.org/downloads/extreme_agile.html, December 19, 2002

[9]. Indika, "Difference Between Agile and Traditional Software Development Methodology", http://www.differencebetween.com/difference-between-agile-and-vs-traditional-software-development-methodology, Jul 10th, 2011

[10]. Beck, K., Extreme Programming Explained: Embrace Change: Addison-Wesley, 2000, ISBN 201-61641-6.

[11]. Erickson, J., Lyytinen, K., and Siau, K., "Agile Modeling, Agile Software Development, and Extreme Programming: The State of Research," Journal of Database Management, no. 4, vol. 16, pp. 88 - 100, 2005.

[12]. Kalra, B., Thomson, C., and Holcombe, M., "The Software Hut - a Student Experience of Extreme Programming with Real Commercial Clients," XP, 2005.

[13]. Schneider, J. G. and Johnston, L., "Extreme Programming – Helpful or Harmful in Educating Undergraduates," Journal of Systems and Software, vol. 74, pp. 121-132, 2005.

[14]. Shukla, A. and Williams, L., "Adapting Extreme Programming for a Core Software Engineering Course," in Proceedings. 15th Conference on Software Engineering Education and Training (Csee&T), 2002, pp. 184-191.

[15]. Stephens, M. and Rosenberg, D., "Extreme Programming Refactored: The Case against XP", Berkeley, CA: Apress, 2003, ISBN 1-59059-096-1.

[16]. T. Gaddis, Starting out with JAVA, New Delhi: Wiley Dreamtech India Pvt. Ltd., 2005

[17]. H. Schildt, The Complete Reference JAVA, 7th edition. New Delhi: Tata McGraw-Hill Publishing Company, 2007

[18]. "Creating a Frame", http://www.roseindia.net/java/example/java/swing/Swing_index.shtml, April 17, 2011.

[19]. "Java Swing Tutorials", http://www.roseindia.net/java/example/java/swing, March 20, 2008.

[20]."Create a JRadioButton Component in Java" http://www.roseindia.net/java/example/java/swing/CreateRadioButton.shtml, April 14, 2007.