

Business Intelligence System for Banking and Finance

Tejas Vaidya, Sagar Ahirrao, Kiran Kadam, Kundan Lugade
*Department of Information Technology, University of Pune
 Pune, Maharashtra, India*

Abstract- Many firms develop an information technology strategy that includes the use of business intelligence software in the decision making process. In order to really achieve a solid return on investment on this type of software, the firm should have at least 10 years of detailed data on sales, purchases, staff costs, and other items that impact the overall cost of providing a service or good. Data cubes and reports can then be built to show trends, identify product success and failures, and provide a more holistic view of company activity. This paper describes such software "Business Intelligence System for Banking and Finance".

I. INTRODUCTION

Business Intelligence and Analytics combines two of the leading concepts in information technology into a powerful tool for both businesses and organizations. In order to utilize these concepts in a productive way, the organization requires skilled, dedicated business analysts, business intelligence software, and powerful analytics tools. The primary purpose of a business intelligence analytics system is to identify trends and patterns in consumer behavior. This information is used to increase profitability, focus the company on client's needs, and provide enhanced accuracy in cost benefit analysis.

Business Intelligence System for Banking and Finance help Bank People to answer questions like

- How do we succeed financially?
- How do we appear to our customers?
- At what processes must we excel?
- How do we sustain our ability to change and grow?

These points show abstract of keys of an organization. The perspectives of system is to decide the strategies, to take action along it, if the result get then measure it whether it is beneficial, in loss or average. It also helps to decide target on particular area.

II. PROBLEM DEFINITION

To employ Business Intelligence System for Banking and Finance to enhance the services provided to these sectors.

- To monitor the daily transactions in banking and finance.
- To give an insight in to the strategies used by the organization and detect gaps in them.
- To identify the different forms of competition faced by the organization.
- To provide a solution to the organization by using different BI techniques.

To design a BI Framework (i.e. Digital Dashboards, Balanced Scorecards), that can be used by Banking and Finance organization, to analyze existing data so as to support them in taking business strategic decisions.

III. APPLYING SOFTWARE ENGINEERING APPROACH

The development model that we will be following for implementing the software is Iterative and Incremental Model.

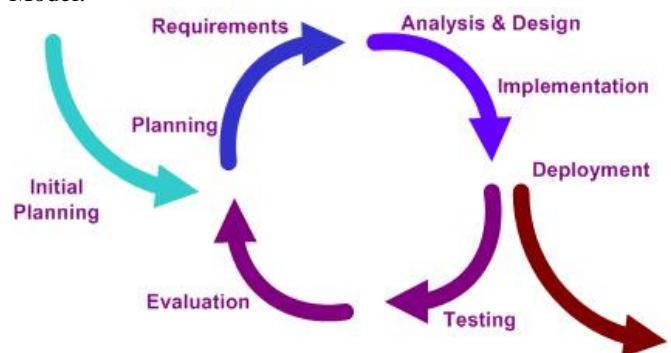


Figure 1.1: Software Engineering Approach

Software Engineering consists of few steps that help in sequential and proper making of any project. The method that we deployed was essentially an "iterative" approach and it includes steps like

1. Requirement gathering
2. Analysis & Design
3. Implementation
4. Testing
5. Deployment.

Iterative and Incremental development is at the heart of a cyclic software development process developed in response to the weaknesses of the waterfall model. It starts with an initial planning and ends with deployment with the cyclic interactions in between. Backtracking is possible in an Iterative and Incremental approach.

IV. LITERATURE SURVEY

Companies have to answer key questions every day. What impact do price changes have on buying behavior? Is the supply chain aligned with consumer demand? Though companies have the data that could answer these questions few properly leverage all of their resources. A company needs to first extract tangible information and then to make sure it gets into the right hands. To effectively and efficiently run the business a company must convert their data into knowledge.

Companies always come up with plans and roadmaps that will improve operations, increase profits, and streamline the process as a whole. Unfortunately 90% of companies fail to execute on their strategies, for a variety of reasons. At the core, a company's problems in executing probably stem from improper use of data due to lack of visibility.

1. NEED OF BUSINESS INTELLIGENCE

A strategic Business Intelligence platform puts the right information in the right hands at the right time, and gives managers and executives the ability to test various scenarios for business spending and investments while monitoring important operational drivers of company performance.

The financial services industry is rapidly changing. Factors such as globalization, deregulation, mergers and acquisitions, competition from non-financial institutions, and technological innovation, have forced companies to re-think their business.

Many large companies have been using Business Intelligence (BI) computer software for some years to help them gain competitive advantage. With the introduction of cheaper and more generalized products to the market place BI is now in the reach of smaller and medium sized companies. Business Intelligence is also known as knowledge management, management information systems (MIS).

Decision-making has always been a critical art in business. BI has been around for decades. The growing sophistication of technology solutions such as digital dashboards and balanced scorecards can substantially help executives make the best decisions possible by providing accurate, timely and easy to comprehend information to transform mere data into business wisdom.

The business value of reporting is simple: it communicates and helps you understand your company's performance. It leads you to take action on recent information. It creates a common context for decision-making across every department and at every level. At its best, reporting is an integral part of performance management.

For business users, the right reporting software is simply the solution that lets them work with information the way they want to. It means accessing relevant information is fast and easy. You are confident that the numbers you see are the same throughout the organization. The reports you receive are presented in context so you can make informed decisions rather than lose time debating what action to take. The right reporting software for business people removes the limits to fact-based, better business decisions.

2. GOALS OF DATA WAREHOUSE

One of the most important assets of any organization is its information. This asset is almost always kept by an organization in two forms: the operational systems of record and the data warehouse. Crudely speaking, the operational systems are where the data is put in, and the data warehouse is where we get the data out.

The users of an operational system turn the wheels of the organization. They take orders, sign up new customers, and log complaints. Users of an operational system almost always deal with one record at a time. They repeatedly perform the same operational tasks over and over. The users of a data warehouse, on the other hand, watch the wheels of the organization turn. They count the new orders and compare them with last week's orders and ask why the new customers signed up and what the customers complained about.

Users of a data warehouse almost never deal with one row at a time. Rather, their questions often require that hundreds or thousands of rows be searched and compressed into an answer set. To further complicate matters, users of a data warehouse continuously change the kinds of questions they ask. Thus goals of Data Warehouse are listed as follow:

- The data warehouse must make an organization's information easily accessible.
- The data warehouse must present the organization's information consistently.
- The data warehouse must serve as the foundation for improved decision making.
- The data warehouse must be adaptive and resilient to change.

2.1. Components of Data Warehouse

It is helpful to understand the pieces of Data Warehouse carefully before we begin combining them to create a data warehouse. Each warehouse component serves a specific function.

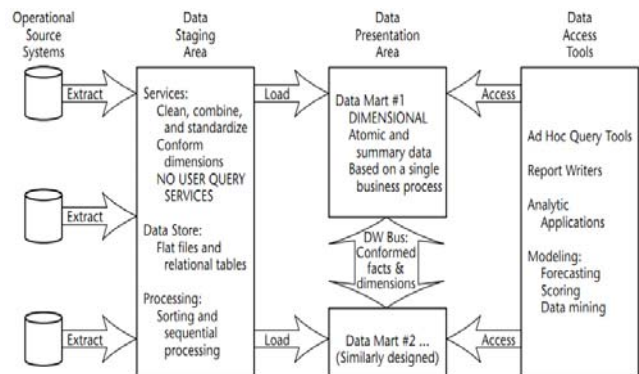


Fig 2.1: Components of Data Warehouse

There are four separate and distinct components to be considered as we explore the data warehouse environment. Let us discuss them one by one

2.1.1. Operational source systems

These are the operational systems of record that capture the transactions of the business. The source systems should be thought of as outside the data warehouse because presumably we have little to no control over the content and format of the data in these operational legacy systems.

2.1.2. Data staging area

The data staging area of the data warehouse is both a storage area and a set of processes commonly referred to as extract-transformation-load (ETL). The data staging area is everything between the operational source systems and the data presentation area.

Extraction is the first step in the process of getting data into the data warehouse environment. Extracting means reading and understanding the source data and copying the data needed for the data warehouse into the staging area for further manipulation. Once the data is extracted to the staging area, there are numerous potential transformations, such as cleansing the data (correcting misspellings, resolve domain conflicts, dealing with missing elements, or parsing into standard formats), combining data from multiple sources, deduplicating data, and assigning warehouse keys.

These transformations are all precursors to loading the data into the data warehouse presentation area.

2.1.3. Data presentation area

The data presentation area is where data is organized, stored, and made available for direct querying by users, report writers, and other analytical applications. Since the backroom staging area is off-limits, the presentation area is the data warehouse as far as the business community is concerned. It is all the business community sees and touches via data access tools.

It is acceptable to create a normalized database to support the staging processes; however, this is not the end goal. The normalized structures must be off-limits to user queries because they defeat understandability and performance. As soon as a database supports query and presentation services, it must be considered part of the data warehouse presentation area. By default, normalized databases are excluded from the presentation area, which should be strictly dimensionally structured.

Unfortunately, some data warehouse project teams have failed miserably because they focused all their energy and resources on constructing the normalized structures rather than allocating time to development of a presentation area that supports improved business decision making.

2.1.4. Data access tools.

The final major component of the data warehouse environment is the data access tool. We use the term tool loosely to refer to the variety of capabilities that can be provided to business users to leverage the presentation area for analytic decision making. By definition, all data access tools query the data in the data warehouse's presentation area.

3. SOFTWARE REQUIREMENT SPECIFICATION

3.1 Introduction

3.1.1 Project Scope

The aim of the project is to create a Decision Support System (DSS) which will provide a solution to banking and finance so that they can identify various trends of their product. In order to achieve the above objective the following has to be done

- Analyze data on hand
- Supply adequate information to user of the system so that he can take an appropriate decision.

The proposed system will be web based which will take into account the existing data and based upon this data help the banking and finance organizations to take decisions so as to enhance services provided by them.

3.1.2 User Classes and Characteristics

There are different kinds of Users using this Product. Each user is privileged and he is having limited access to reports.

a. Developer:

Individual developers should be able to submit issues, edit issues, and perform queries on the database to discover what issues are relevant to them, which issues are open (in the case of issues to which that is relevant, such as defect reports or unsatisfied requirements), etc.

b. Admin:

Admin have authority to add, modify, delete data from database. Those responsible for keeping track of open issues, assigning authority to clients and users.

c. Users:

These are the banking people who are interested only in reports like Digital Dashboards and Balanced Scorecards.

3.1.3 Operating Environment

The Product will be implemented using PHP. It should therefore run on any platform. We will avoid the use of platform-specific designs or code in doing so. However, the primary operating systems we'll be using are Windows XP.

As the end product will be Web Based, it will be compatible with browsers IE6.0, Mozilla Firefox and Chrome.

3.1.4 Design and Implementation Constraints

- The Product will be developed using PHP and the backend database for this is MySQL.
- The product is accomplished with login facility so that specific function is available to specific User.
- The Product Architecture should be resilient to changes in the future as demanded by Users.
- The GUI should appear consistent throughout the product and all functionalities should be easily locatable.

3.1.5 Assumptions and Dependencies

The product needs following third party product.

1. PHP as frontend
2. MySQL as backend.

We are going to assume that Banking Database is already imported at our end.

3.2 System Features

3.2.1 Database Storage

3.2.2 Description and Priority

Proposed Database is intended to store, retrieve, update, and manipulate information related to users which include

- Multidimensional Data Modeling
- Banking product information
- Sales details
- History of Product

3.2.3 Stimulus / Response Sequences

- *Stimulus:*

At the Banking end due to Transactions Data is generated.

- *Response:*

Data is stored in Database.

- *Stimulus:*

Front end activities processes Data and transform it in intuitive form.

- *Response:*

Data is made available and produced information is stored back.

3.2.4 User Login

Any user including admin can login and logout in system. When any user logs into the system, the system will check for validity of login .If the login id and password are valid, the response to this action is the user will be able to modify, view, add, deleting and all other functions that can be performed on the database according to his privilege..

3.2.5 Reports

Reports are generated on request of User to show trends, identify product success and failures, and provide a more holistic view of company activity. Reports are also presented in form of Dashboards and Scorecards.

3.2.6 Alerts

For every user alerts are generated for every desired and undesired event. Even User can send alert any other user belonging to our system.

3.3 External Interfaces Requirements:

3.3.1 User Interfaces

As there are number of levels in Banking User, each user have access to reports according to his position in Bank hierarchy. Each User is having its own interface to System.

Different levels in Hierarchy are:

- Top level(Chief executive officer, Chief financial officer)
- Middle Level(Chief operating officer, Head of departments)
- Low level(Staff)

3.3.2 Hardware Interfaces

- Operating System: Windows XP.
- Processor: Pentium III or 2.0 GHz or higher.
- RAM: 256 Mb or more

3.3.3 Software Interfaces

- Database: MySQL
- Application: PHP (Hypertext Preprocessor), jQuery Library
- Web Server: Apache Tomcat, Web server that provides a highly reliable, manageable, and scalable Web application infrastructure)

3.3.4 Communication Interfaces

The Customer must connect to the Internet to access the Website using any of:

- Dialup Modem
- Broadband Internet
- Dialup or Broadband Connection with a Internet Provider.

3.4 Nonfunctional Requirements

3.4.1 Performance Requirements

The Reports will be generated as and when required by the User in very small response time. Also system will be Web based and at a time accessed by various users located at different places. The numbers should be consistent throughout the system.

3.4.2 Safety Requirements

The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup or to Replicate Database.

3.4.3 Security Requirements

There are different categories of users. Depending upon the category of user the access rights are decided. It means if the user is an administrator then he can be able to modify the data, delete, append etc. All users will be having their own ID and password to login into system and other than staff only have the rights to retrieve the information about database.

3.4.4 Software Quality Attributes

Different Software Quality Attributes are listed below:

- GUI: The User Interface of the System should be consistent throughout.
- Database: The MySQL is faster than any other RDBMS.
- Security: Each User will be having his own ID and password to login into system and is having limited access to reports depending on his privilege.
- Alerts: For every user alerts are generated for every desired and undesired event.
- Response time: Reports are generated on demand in very less time i.e. response time is very low.
- Functionalities within the system are easily accessible and locatable.
- System architecture will be resilient to undergo changes demanded in future by its user.

3.5 Analysis Model

3.5.1 Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing.

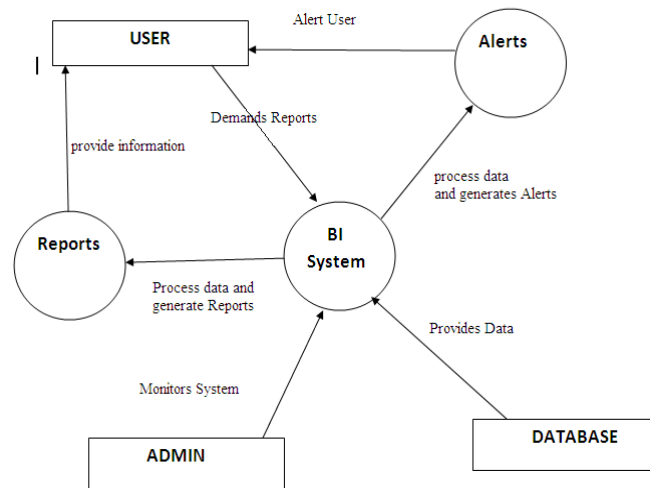


Figure 3.1: Data Flow Diagram

3.5.2 Class Diagram

The class diagram shows how the different entities (people, things, and data) relate to each other; in other words, it shows the static structures of the system. A class diagram can be used to display logical classes, which are typically the kinds of things the business people in an organization talk about.

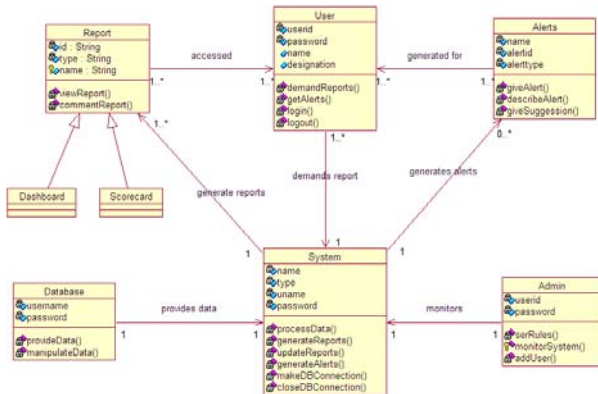


Figure 3.2: Class Diagram

3.5.3 E-R Diagram

An entity-relationship diagram is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system.

3.5.3.1 Deposit Schema

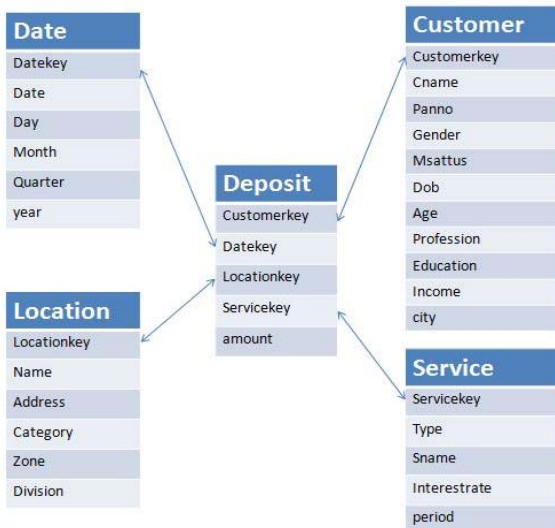


Figure 3.3: Deposit Schema

3.5.3.2 Loan Schema

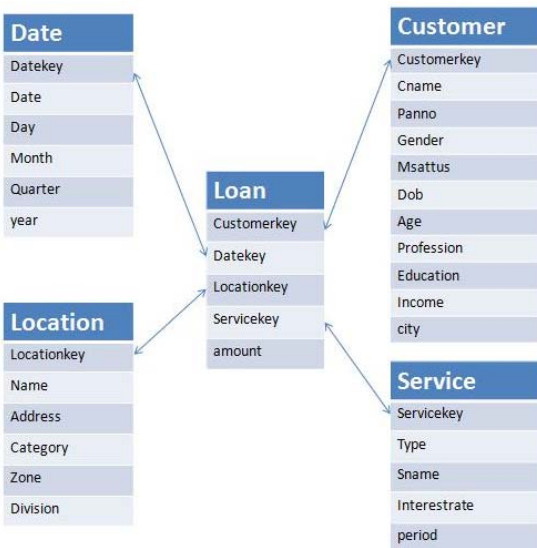


Figure 3.4: Loan Schema

3.6 System Implementation Plan

The purpose of System Implementation is making the new system available to a prepared set of users (the deployment), and positioning on-going support and maintenance of the system within the Performing Organization (the transition).

The System Implementation Plan is as follow:

Stage 1: Literature Study of –

- Bank Management concepts
- Bank’s internals (parameters) details study
- Study of PHP, JQuery Library, Multidimensional Data Modeling and scripting language Javascript
- Handling the webserver like WAMP
- Designing database on MySQL serve.
- Designing graphs using Google Chart APIs
- Designing of Excel Sheets Structure.

Stage 2: Implementation of–

- Database Schema Design and creation.
- User Administration Module
- PHP scripts for Mailing Functionality
- Creation of Excel sheets
- GUI using HTML, CSS and JQuery Library.

Stage 3: Implementation of –

- PHP scripts for handling ETL process
- Implementation of Reports and Graphs
- Implementation of Alerts
- Implementation of Client side Validations
- Implementation of Server side Validations
- Implementation of ETL module with Upload Functionality

Stage 4: Implementation of

- Design of Test cases
- Integration of ETL Module, Report and Graph Module into System
- Testing of each Module

4. SYSTEM DESIGN

4.1 System Architecture

The System will be web-based application that can be accessed via any web browsers over a network. It is a 3-tier solution in which the user interface, the business logic and the database are developed and maintained as independent modules.

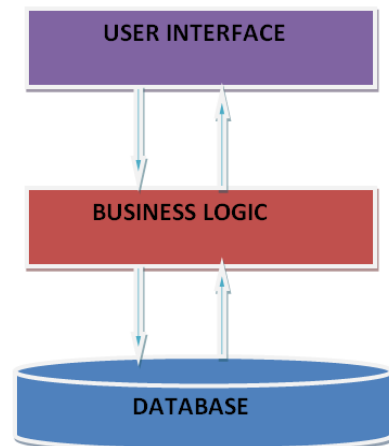


Figure 4.1: Three Tier Architecture

1. User Interface:

Top layer of the application, the interface main function is to display information received from other tiers in a user-friendly format. The processed form of data i.e. information is presented to end user in more intuitive form which helps to take quick decisions.

2. Business Logic:

This middle layer coordinates the application by handling and processing information exchanges between the database and the user interface, and by making logical decisions and performing calculations.

3. Data Management:

Bottom layer of the application, the data management consists of a database server that stores all system's data. It keeps data independent from application servers or business logic.

4.2 UML Diagrams

4.2.1 Use Case Diagram

4.2.1.1 User Login

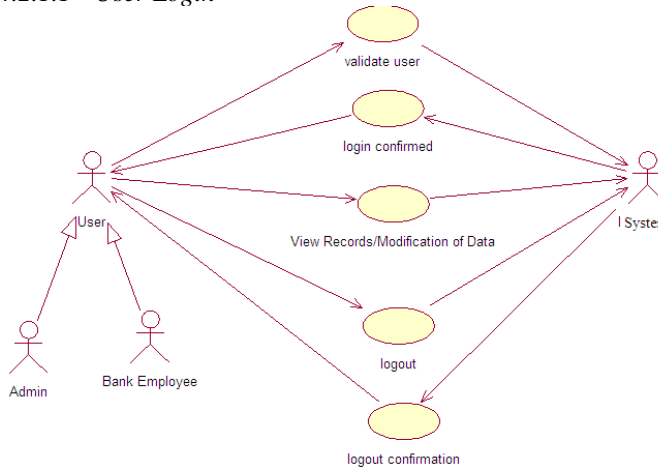


Figure 4.2: User Login- Use Case

4.2.1.2 System Behavior

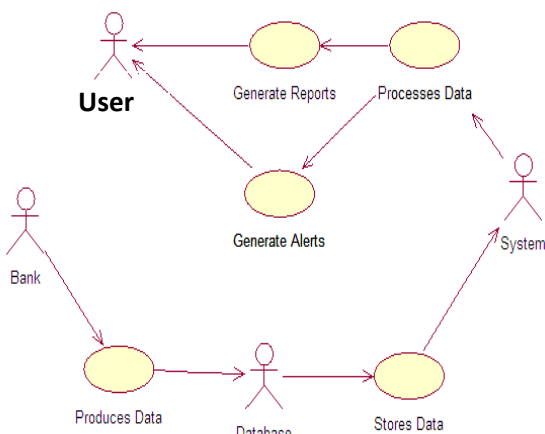


Figure 4.3: System Behavior- Use Case

A use-case diagram is typically used to communicate the high-level functions of the system and the system's scope. By looking at our use-case diagrams, you can easily tell the functions that our example system provides.

4.2.2 Sequence Diagram

Sequence diagrams show a detailed flow for a specific use case or even just part of a specific use case. They are almost self explanatory; they show the calls between the different objects in their sequence and can show, at a detailed level, different calls to different objects.

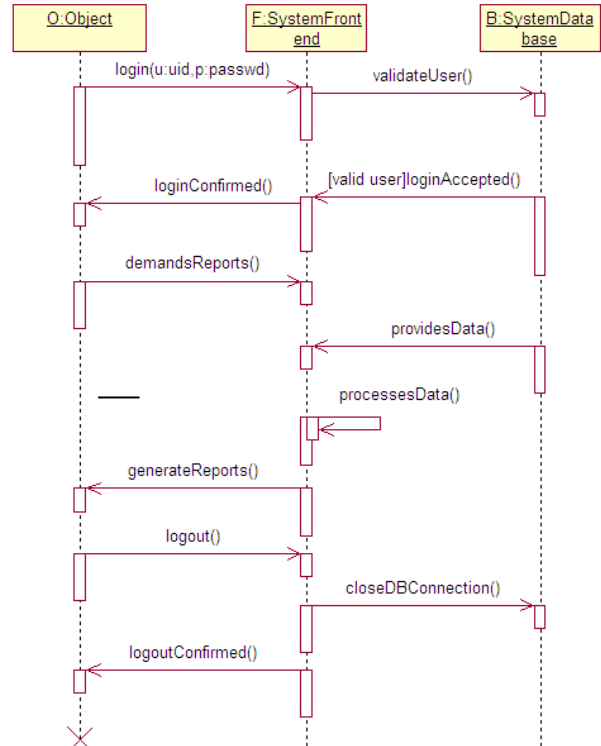


Figure 4.4: Sequence Diagram

4.2.3 Deployment Diagram

The deployment diagram shows how a system will be physically deployed in the hardware environment. Its purpose is to show where the different components of the system will physically run and how they will communicate with each other.

The notation in a deployment diagram includes the notation elements used in a component diagram, with a couple of additions, including the concept of a node. A node represents either a physical machine or a virtual machine node (e.g., a mainframe node).

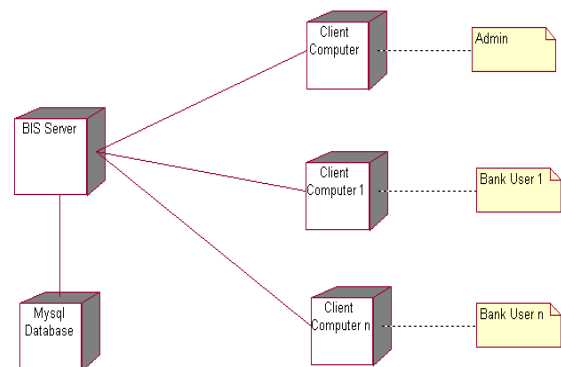


Figure 4.5: Deployment Diagram

5. TECHNICAL SPECIFICATION

5.1 Technology Details used in the Project

We are using following Technologies to develop our Application:

- PHP : for building Business Logic
- Mysql : to store and manage data
- JQuery & AJAX : for giving Feel of Desktop Application
- Google API : for generation of Graphs
- JavaScript : For Validating Client side Data

Let us see one by one shortly

5.1.1 PHP

PHP is one of the hottest scripting language to be found on the Internet, which grants many capabilities to the web programmer. In other language there are some difficulties in writing the script but in PHP it can be swiftly executed in few lines. In actual sense PHP is server-side, HTML –embedded, cross platform scripting language. In simpler terms PHP provides the facility to write instructions in the HTML files to create dynamic content. These instructions are read and computed by the Web Server and they never appear on the browser that is displaying the page. The server replaces the PHP code and only HTML code appears on the page.

PHP is a combination of Perl, Java and C concepts. The syntax is borrowed from C, which makes it an easy language to learn for a novice programmer. If someone has an experience with any of the above mentioned language, it will be easy for him to learn PHP. PHP performs sophisticated mathematical calculation, network information, regular expression capabilities, mail service and much more. In fact PHP provides a wide variety of functions that support everything from array manipulation to regular expression support.

PHP's strongest feature is its database interfacing capabilities. In PHP it is easy to connect database to the Internet. It supports many of the most popular database servers available in the market, including MySQL, Oracle, Sybase, Generic ODBC and PostgreSQL, to name a few. Through database connectivity it is possible to take a company's database of product and write a web interface to it using PHP. In particular here we will discuss the interfacing capabilities of PHP with MySQL, the most powerful database server found in the market.

To access database in a more efficient way PHP offers a persistent connection to the databases. If the database is having a persistent connection then there is no need for reconnecting to the databases every time. PHP will stay logged into the databases between requests and re-use connections wherever possible. This makes database access very fast.

PHP also has support for talking to other services using protocols such as IMAP, SNMP, NNTP, POP3 or even HTTP.

5.1.2 MySQL

Many relational database concepts are really quite simple. In fact, much of the appeal of relational databases stems from the simplicity of their foundational concepts.

Within the database world, MySQL is classified as a relational database management system (RDBMS). That phrase breaks down as follows:

The database (the "DB" in RDBMS) is the repository for the information you want to store, structured in a simple, regular fashion:

- The collection of data in a database is organized into tables.
- Each table is organized into rows and columns.
- Each row in a table is a record.
- Records can contain several pieces of information; each column in a table corresponds to one of those pieces.

The management system (the "MS") is the software that lets you use your data by allowing you to insert, retrieve, modify, or delete records.

The word "relational" (the "R") indicates a particular kind of DBMS, one that is very good at relating (that is, matching up) information stored in one table to information stored in another by looking for elements common to each of them.

The power of a relational DBMS lies in its capability to pull data from those tables conveniently and to join information from related tables to produce answers to questions that can't be answered from individual tables alone.

5.1.3 jQuery and AJAX

jQuery is an open-source JavaScript kit for building Web applications so dynamic they jump off the page. Filled with special controls like calendars and tab folders, and special effects like wipes and fade-ins, jQuery is gaining popularity rapidly. jQuery is a JavaScript library full of tools ready to be used which means that it's prewritten JavaScript, ready for you to put to work in your own Web pages.

jQuery specializes in letting you select elements in a page, and it does that better than any other JavaScript library. You'll see how to create wrapped sets of elements in jQuery, so you can handle multiple elements at the same time. You'll also see how to manipulate wrapped sets of elements by changing their appearance, style, visibility, text, and even their HTML.

Perhaps most important, jQuery gives you excellent support for Ajax applications. Ajax is what allows you to access a Web server from a browser without a page refresh that is, there's no blinking, no flicker when you download data behind the scenes with Ajax; you just download the data and then you can display it in a Web page using dynamic HTML techniques. No fuss no muss and the end result is an application that looks more like a desktop application than a Web application.

5.1.4 Google API

The Google Visualization API lets you access multiple sources of structured data that you can display, choosing from a large selection of visualizations. Google Visualization API enables you to expose your own data, stored on any data-store that is connected to the web, as a Visualization compliant data source. Thus you can create reports and dashboards as well as analyze and display your data through the wealth of available visualization

applications. The Google Visualization API also provides a platform that can be used to create, share and reuse visualizations written by the developer community at large.

- Embed visualizations directly into your website: Display attractive data on your website by choosing from a vast array of visualizations created by the developer community.
- Write, share, and reuse: Write visualizations and also wrap them as Gadgets with the simple Gadget Extensions of the API. Publish these here or in the Gadget Directory. Become an active participant in the developer community; reuse and share visualizations with others.
- Create dashboards and reports over your own data: Expose your data stores as Google Visualization-compliant data sources. Create dashboards and publish your insights (within your domain or publicly) anywhere on the web.
- Create extensions to Google products: Write visualization applications for Google products such as Google Docs. Syndicate your application with a growing list of products that support Gadgets.
- Use many data sources, one API: Visualizations created using the API are able to access any compliant server-side data source, or access data directly from the client using JavaScript, with no required code changes to your application.

5.1.5 Java Script

JavaScript is ubiquitous on the World Wide Web. We can use JavaScript both to make our Web pages more interactive, so that they react to a viewer’s actions, and to give our Web pages some special effects (visual or otherwise).

JavaScript often gets thrown in with Hypertext Markup Language (HTML) as one of the recommended languages for beginning Web developers (whether you use Web sites for business or pleasure). Of course, we can build a Web page by using only HTML, but JavaScript allows us to add additional features that a static page of HTML can’t provide without some sort of scripting or programming help.

5.2 Advantages

The Product digests large volumes of information and conveys information in an intuitive format, making it easy to classify and respond to critical, time sensitive events. They also help in exploring issues and trends without getting completely lost in huge data or reports.

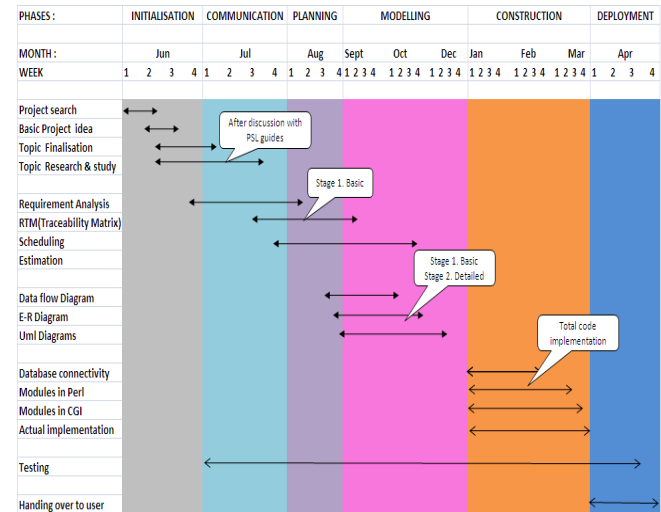
For business users, the right reporting software is simply the solution that lets them work with information the way they want to and saves their time.

5.3 Disadvantages

The System depends on OLTP server for input data in form of Excel sheets. Thus it is dependent on the structure of Excel sheets. If OLTP server changes the structure of Excel sheets then we have to manually configure the configuration file of ETL module.

6. PROJECT ESTIMATE, SCHEDULE

6.1 Schedule



6.2.2. Cost Estimate

Software Cost

WAMP, PHP, JQuery Library

7. SOFTWARE IMPLEMENTATION

7.1 Database

The Database Schema we have designed is of Multidimensional nature. We have designed Star Schema to store the data in Data Warehouse. Schema consists of following two kinds of Tables:

➤ Fact Tables:

A fact table is the primary table in a dimensional model where the numerical performance measurements of the business are stored.

➤ Dimension Tables:

Dimension tables are integral companions to a fact table. The dimension tables contain the textual descriptors of the business.

Four steps carried out for Multidimensional Design Process are

1. Select the business process to model.
2. Declare the grain of the business process.
3. Choose the dimensions that apply to each fact table row. Dimensions fall out of the question, “How do businesspeople describe the data that results from the business process?”
4. Identify the numeric facts that will populate each fact table row. Facts are determined by answering the question, “What are we measuring?”

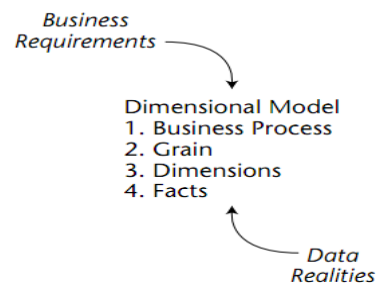


Fig 6.2: Key Input to Four Step dimensional design process

CUSTOMER[Dimension]					
Sr.No	COLUMN NAME	DATA TYPE	KEY	NULL	Comments
1	CUSTOMERKEY	INT(11)	Y	N	SELF GENERATED
2	CNAME	VARCHAR(30)			MANUAL
3	PANNO	INT(11)			MANUAL
4	GENDER	ENUM			MANUAL
5	MSTATUS	ENUM			MANUAL
6	DOB	DATE			MANUAL
7	AGE	TINYINT(3)			MANUAL
8	EDUCATION	ENUM			MANUAL
9	PROFESSION	ENUM			MANUAL
10	INCOME	DOUBLE(5,2)			MANUAL
11	CITY	VARCHAR(20)			MANUAL

Table 7.1 Structure of CUSTOMER table

DATE[Dimension]					
Sr.No	COLUMN NAME	DATA TYPE	KEY	NULL	Comments
1	DATEKEY	INT(11)	Y	N	SELF GENERATED
2	DATE	DATE			MANUAL
3	DAY	TINYINT(2)			MANUAL
4	MONTH	TINYINT(2)			MANUAL
5	QUARTER	TINYINT(1)			MANUAL
6	YEAR	YEAR(4)			MANUAL

Table 7.2 Structure of DATE table

DEPOSIT[Fact]					
Sr. No	COLUMN NAME	DATA TYPE	KEY	NULL	Comments
1	CUSTOMERKEY	INT(11)	Y	N	REFERENCES CUSTOMERKEY OF TABLE CUSTOMER
2	DATEKEY	INT(11)	Y	N	REFERENCES DATEKEY OF TABLE DATE
3	LOCATIONKEY	INT(11)	Y	N	REFERENCES LOCATIONKEY OF TABLE LOCATION
4	SERVICEKEY	INT(11)	Y	N	REFERENCES SERVICEKEY OF TABLE SERVICE
5	AMOUNT	DOUBLE(20,3)			MANUAL

Table 7.3 Structure of DEPOSIT table

LOAN[Fact]					
Sr. No	COLUMN NAME	DATA TYPE	KEY	NULL	Comments
1	CUSTOMERKEY	INT(11)	Y	N	REFERENCES CUSTOMERKEY OF TABLE CUSTOMER
2	DATEKEY	INT(11)	Y	N	REFERENCES DATEKEY OF TABLE DATE
3	LOCATIONKEY	INT(11)	Y	N	REFERENCES LOCATIONKEY OF TABLE LOCATION
4	SERVICEKEY	INT(11)	Y	N	REFERENCES SERVICEKEY OF TABLE SERVICE
5	AMOUNT	DOUBLE(20,3)			MANUAL

Table 7.4 Structure of LOAN table

LOCATION[Dimension]					
Sr.No	COLUMN NAME	DATA TYPE	KEY	NULL	Comments
1	LOCATIONKEY	INT(11)	Y	N	SELF GENERATED
2	NAME	VARCHAR(30)			MANUAL
3	ADDRESS	VARCHAR(30)			MANUAL
4	CATEGORY	ENUM			MANUAL
5	ZONE	VARCHAR(10)			MANUAL
6	DIVISION	VARCHAR(30)			MANUAL

Table 7.5 Structure of LOCATION table

SERVICE[Dimension]					
Sr.No	COLUMN NAME	DATA TYPE	KEY	NULL	Comments
1	SERVICEKEY	INT(11)	Y	N	SELF GENERATED
2	TYPE	ENUM			MANUAL
3	SNAME	VARCHAR(30)			MANUAL
4	INTERESTRATE	DOUBLE(5,2)			MANUAL
5	PERIOD	TINYINT(2)			MANUAL

Table 7.6 Structure of SERVICE table

USER					
Sr . N o	COLUMN NAME	DATA TYPE	KEY	NUL L	Comments
1	USERID	VARCHAR (20)	Y	N	SELF GENERATED
2	PASSWORD	VARCHAR (32)			MANUAL
3	FNAME	VARCHAR (15)			MANUAL
4	MNAME	VARCHAR (15)			MANUAL
5	LNAME	VARCHAR (15)			MANUAL
6	EMAIL	VARCHAR (30)			MANUAL
7	TYPE	ENUM			MANUAL
8	DATE	DATE			MANUAL
9	ISDELETE D	TINYINT (1)			MANUAL
10	LOCATION KEY	INT(11)	Y	N	REFERENCES LOCATIONKEY OF TABLE LOCATION

Important Modules

Important Modules of our system are as follow:

- User Administration :
It manages task like user login, add and delete users.
- Graphs :
It generates Graphs on user request
- Reports :
It generates Reports on user request
- Alerts :
It generates Graphs on Alerts user request
- ETL:
It accepts Excel sheets from OLTP server and store data in Multidimensional Database after ETL process.

Business Logic and Architecture

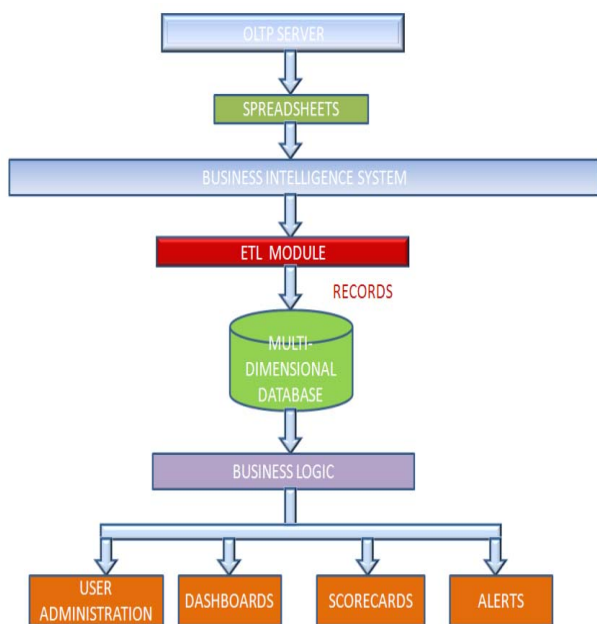


Fig 7.1: System Architecture

The diagram depicts the overall implementation of our System. As shown in figure the Flow of Implementation is as follow:

- The data produced by daily transactions in bank is processed by OLTP server and given to BIS in form of Excel sheets.
 - The ETL (Extract-Transform-Load) Module collects these Spreadsheets and after ETL process the data is loaded into Multidimensional Database.
 - Now the Business Logic takes data from data warehouse and produces intuitive Reports and Graphs in form of Dashboards and Scorecards.
- The User Administration Module takes care of Business Constraints like Security, User Management.

8. SOFTWARE TESTING

8.1 Introduction

8.1.1 Overview

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs (errors or other defects).

Software testing can also be stated as the process of validating and verifying that a software program/application/product that meets the business and technical requirements that guided its design and development.

Once the entire system has been built then it has to be tested against the “System Specification” to check if it delivers the required features. System testing verifies the entire product, after integrating all software and hardware components and validates it according to original project requirements. Testing also verifies the software operation from the perspective of the end user with different configurations or setup.

This section contains a log of the tests that we have carried out on the product as a part of the testing strategy:

We have prepared test cases for each of the module that has been designed by us. Considering how the end user would use the application we have designed the GUI in a very simple manner so that user will be able to use it with minimum efforts. The names that we have used are self explanatory and also he would finish his task within a short span of time.

8.1.2 Types of Testing

- Unit testing
- Integration testing
- Validation testing
- System testing

8.1.2.1. Unit Testing

Unit testing is a method by which individual units of source code are tested to determine if they are fit for use. A unit is the smallest testable part of an application. In procedural programming a unit may be an individual function or procedure. In object-oriented programming a

unit is usually a method. Unit tests are created by programmers or occasionally by white box testers during the development process

8.1.2.2. Integration Testing

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before system testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

8.1.2.3. Validation Testing

Validation testing is a Quality assurance process of establishing evidence that provides a high degree of assurance that a product, service, or system accomplishes its intended requirements. This often involves acceptance of fitness for purpose with end users and other product stakeholders.

8.1.2.4. System Testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code.

8.1.3 Testing Methodology

To be truly robust, application require more than simple functional testing before release into production. At least one and preferably all of the following types of testing before releasing application to customers should be performed.

- Performance Testing
- Load Testing
- Stress Testing

8.1.3.1. Performance Testing

Performance testing is used to determine the speed or effectiveness of a computer, network, software program or device. This process can involve quantitative tests done in a lab, such as measuring the response time or the number of MIPS (millions of instructions per second) at which a system functions. Qualitative attributes such as reliability, scalability and interoperability may also be evaluated. Performance testing is often done in conjunction with stress testing.

8.1.3.2. Load Testing

Load testing is the process of putting demand on a system or device and measuring its response. Load testing is performed to determine a system's behavior under both normal and anticipated peak load conditions. It helps to identify the maximum operating capacity of an application as well as any bottlenecks and determine which element is causing degradation.

8.1.3.3. Stress Testing

Stress test allows examining how the application behaves under a maximum user load. To stress test application, remove the think time for load scripts and execute the scripts against the server to overload use of the application. If there are unhandled exceptions an a stress test, the application may not be robust enough to handle a

sudden unexpected increase in user activity.

8.2 Test cases

8.2.1 Test Case for Login Functionality

Test Case Id	1
Description	Checking Login Functionality
Steps to Reproduce.	<ul style="list-style-type: none"> • Enter url to display home page. • Submit the credentials.
Expected Output	<ul style="list-style-type: none"> • For correct credentials user should be redirected to Menu page. • For incorrect credentials User should be redirected to login page with error messages.
Actual Output	<ul style="list-style-type: none"> • User Redirected menu page • User Redirected to login page
Status	Passed
Verified By	Akshay

8.2.2 Test Case for Client Side Validation :

Test Case Id	2
Description	Check client side validation
Steps to Reproduce.	<ul style="list-style-type: none"> • login to system • Navigate to Change Password menu
Expected Output	<ul style="list-style-type: none"> • For same password and confirm password user password should be change. • For different password and confirm password there should be error message.
Actual Output	<ul style="list-style-type: none"> • Error message displayed.
Status	Passed
Verified By	Arvind

8.2.3 Test Case for Logoff Facility :

Test Case Id	3
Description	Check Logoff facility
Steps to Reproduce.	<ul style="list-style-type: none"> • Enter url to display home page. • Login to system. • Click Logoff tab.
Expected Output	<ul style="list-style-type: none"> • Index page should be displayed with Logout message.
Actual Output	<ul style="list-style-type: none"> • Index page should be displayed with Logout message.
Status	Passed
Verified By	Salil

8.2.4 Test Case for Upload Excel File :

Test Case Id	4
Description	Check if all records inserted successfully
Steps to Reproduce.	<ul style="list-style-type: none"> Enter url to display home page. Login to system Select utility tab and choose option add data
Expected Output	<ul style="list-style-type: none"> If above option is clicked, it must be upload file
Actual Output	<ul style="list-style-type: none"> Success message must be displayed.
Status	Passed
Verified By	Chirag

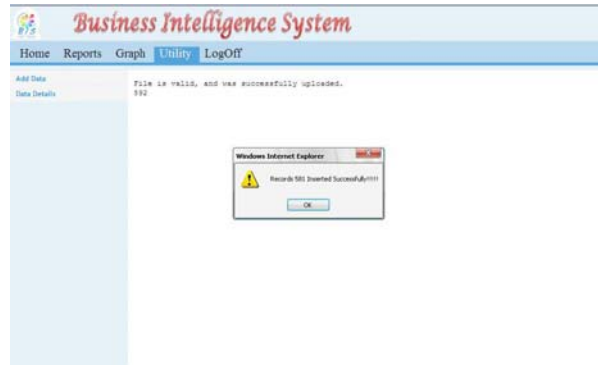


Fig 8.4: Test Case for Upload Excel File

10. RESULTS

Result Snapshots

9. SNAP SHOTS OF THE TEST CASES AND TEST PLAN



8.1: Test Case for Login Functionality



Fig 9.1: Login Page

Fig



Fig 8.2: Test Case for Client side Validation

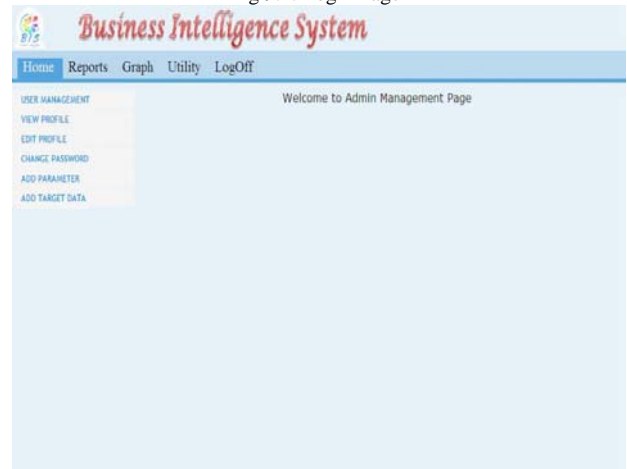


Fig 9.2: Admin Home Page (ss2.png appears hear in the center of the page)



Fig 8.3: Test Case for Logoff Functionality



Fig 9.3: User Management Page



Fig 9.4: Add User Page



Fig 9.8: Report



Fig 9.5: Graph: Pie Chart
(ss1.png appears here instead of Pie Chart followed by supporting table.)

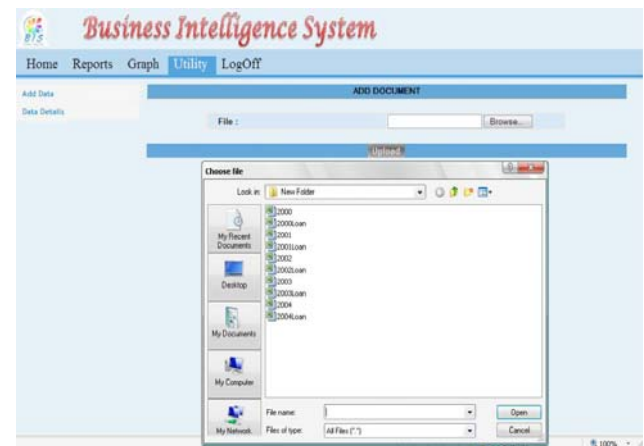


Fig 9.9: Upload Excel File

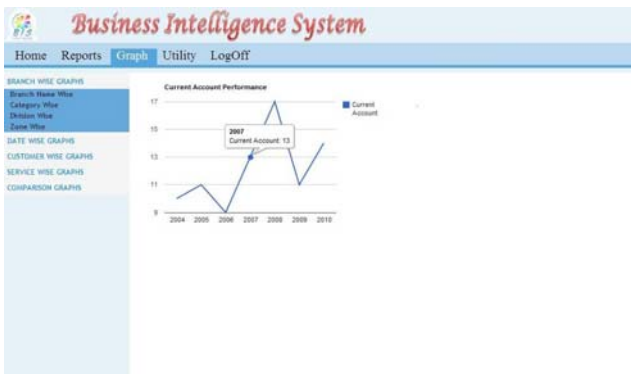


Fig 9.6: Graph: Line Chart
(ss1.png appears here instead of Pie Chart followed by supporting table.)



Fig 9.7: Graph: Bar Chart
(ss1.png appears here instead of Pie Chart followed by supporting table.)

11. DEPLOYMENT AND MAINTENANCE

12.

11.1 Installation And Uninstallation

11.1.1 Steps for Installation of Our System are:

- Install WAMP2.0 Server on Server Machine and make following configurations in Mysql:
 - Username : root
 - Password : root
 - Database : bis.
- Import bis.sql in Mysql using phpAdmin.
- Now Server is configured Successfully.
- On Client side just enter url : <http://localhost/BIS/index.php> in Browser.
- Now Application is ready to work.

11.1.2 Steps for uninstallation of Our System are:

- To uninstall our System just uninstall WAMP2.0 server on server machine or delete Project folder in www directory on WAMP server and then drop bis database in Mysql.
- Now Application is uninstalled.

11.2 User Help

- For user help extra tab is provided on index page. If index page is not available then ensure that server is on. If server is not on then index page cannot be loaded in browser on client machine.

V. CONCLUSION AND FUTURE SCOPE

Conclusion

Business Intelligence System for Banking and Finance is an application which will be providing a helping hand to Banking people to lets them work with information the way they want to. The growing sophistication of technology solutions such as digital dashboards and balanced scorecards can substantially help executives make the best decisions possible by providing accurate, timely and easy to comprehend information to transform mere data into business wisdom. It creates a common context for decision-making across every department and at every level. At its best, reporting is an integral part of performance management. Our System will help bank to gain competitive advantage over its different products.

The reports you receive are presented in context so you can make informed decisions rather than lose time debating what action to take.

The user using this application can trigger alerts for respective authorities so that decisions can be taken more quickly.

Future Scope

Business Intelligence System is general purpose system. It can be designed for any organization (e.g. Computer firm, accountancy firm, etc). This project is

designed for single state, which covers 8 Districts and 32 Branches. In future, it can modified to accommodate multiple states and can even work for entire organization throughout the Nation. Some of the features that we would like provide in the future are

- Adding the threshold alert system
- Making Generic Business Intelligence System for any Domain
- Enable Market based Analysis
- Template creation
- Local / internationalization.

REFERENCES

- [1]. Importance of Business Intelligence, Mike Price, Jon
- [2]. IBM Cognos 8 Business Intelligence Reporting – Data sheet
- [3]. Learning PHP and MySQL, Second Edition by Michele E. Davis and Jon A. Phillips
- [4]. Beginning PHP5, Apache, and MySQL® Web Development, Published by Wiley Publishing, Inc. 10475 Crosspoint Bouleva
- [5]. Wiley & Sons - The Data Warehouse Toolkit. Second Edition
- [6]. Java Script in easy steps
- [7]. HTML4 in easy steps.
- [8]. <http://php.net/docs.php>
- [9]. <http://dev.mysql.com/doc>
- [10]. <http://code.google.com/apis/charttools/index.html>
- [11]. <http://www.w3schools.com/default.asp>
- [12]. <http://docs.jQuery.com/Tutorials>