FireBolt A Financial Management System

By

Mushtaq Ali (2009-NUST-BIT-311)

Ata Ur Rehman (2009-NUST-BIT-307)

Project documentation submitted in partial fulfillment of the requirements for the degree of

Bachelors in Information Technology (BIT)

National School of Electrical Engineering & Computer Sciences

National University of Science & Technology

Islamabad, Pakistan

2013

CERTIFICATE

It is certified that the contents and form of Project entitled **FireBolt A Financial Management System** Submitted by **Mushatq Ali** and **Ata Ur Rehman** was done under our supervision and it is satisfactory for the requirement of the degree.

Advisor: _____

Muhammad Bilal

Co-Advisor: _____

Bilal Ali

DEDICATION

Our project is dedicated to Almighty Allah, parents, faculty and friends.

ACKNOWLEDGEMENTS

We are thankful to our advisor and Co-Advisor, Muhammad Bilal, Bilal Ali for providing help and guidance for completion of this project. With their motivation, support and timely feedback, we were able to complete the project and achieved the objectives of our final year project.

We are also thankful to Mr.Kahif Sharif and Mr.Majid Maqbool for their precious feedback on the eve of final year project expos, their feedback helped us in better understanding and implementing the requirement of project. We are also thankful to Mr. Naeem Qasim for giving us suggestions to resolve issues in the project.

ABSTRACT

Today every organization requires an efficient and effective application that can manage financial transaction. Among the various sub-systems of a financial application is General Ledger which is also known as GL that acts as a central repository for all financial transactions. General ledger is the summary of all transactions occurring at any company. GL includes features like chart of accounts management, financial year and periods support, and multiple-currency, multi-language support and comprehensive financial report writer.

TABLE OF CONTENTS

TABLE OF FIGURES

Abbreviations

ADF	Application Development Framework
XML	Extensible Markup Language
XFB	Next Generation Financial Application
PBP	Project base package
JAAS	Java Authentication And Authorization Service
SOA	Service oriented Architecture
IDE	Integrated Development Environment
JSF	Java Server Faces

Chapter 1

INTRODUCTION

In this chapter we will discuss the importance of a software based solution for managing financial transactions for an organization, some existing systems, project scope and objectives, technology used and problem statement.

1.1 PREFACE

Today almost every organization has their finance department. Finance department have different responsibilities like keeping trace of account payable, account receivable, assets, inventory, sales and purchases. Need of the hour is that there should be a system which should do all these functions with efficiency and effectiveness. The system should help executives in decision managing, this thing will help them in growing and this also makes them enable to know that in which department improvements are needed, for this purpose reports are needed on the organization choice (monthly, weekly, yearly). For this purpose organizations want to have a comprehensive report writer in proposed solution. Larger organizations deal in different kind of currencies, so there is need that should be a system who could be there who can help them in multi-currency scenario. In some cases people from different global areas work at a place and they need a system which should work in their own language so multi-locale is also needed for financial applications. So an application which should keep track of all financial transactions happening in an organization is needed.

1.2 IMPORTANCE

The system is of vital importance for all organizations who manage their financial data by manual entry or by paper work. All of their record will be available to them in single application. The provision for bank payments and receipts, cash payments and receipts and journal vouchers all will be available to them. The software will also include a reporting component that will help the users to run important financial reports like vouchers, trial balance, balance sheet, tax

payables, bank/cash reconciliation statements, income statements, and many more. The system has extra features like security, scalability and many more due to which it would be more useful and helpful to organization.

1.3 PROBLEM

In order to provide better financial recording and reporting the transactions, organizations need such a system which has features like scalability, availability, security, low cost and easy to use.

1.3.1 No Database in organizations

Still a lot of organizations in Pakistan (including Government Organizations) use register for keeping track of their transaction. Even they don't use computers for keeping track of their transactional reports. However this culture is changing but the change is very minor for example they are using spread sheets for preserving financial transactions. The database tools for saving the data are not present there. So it becomes very difficult to keep financial reporting (which is very large in many cases) without any database. So a Paper-less system which is scalable and is easy to use is required for such organizations so that they effectively maintain transaction history.

1.3.2 Lack of Scalable application

There are system present which are used for keeping history of financial transactions but problem with them is that they are not scalable. Scalability is measured in terms of system performance when load is increased beyond a threshold. The system should work normally in case of increase load or it should not crash but the dilemma of the hour is that applications crashes or their performance becomes so degraded that it becomes unacceptable for the users and when users become dissatisfied then applications fail in such cases.

1.3.3 Lack of Security in GL Systems

The security of financial application is of vital importance. But the factor is that existing financial applications don't have much security in them. The transaction history which is highly confidential also becomes available to other users and problem of exposed information is present. Issues of Authentication, Authorization, and Role management are present there in current applications. So a system which caters all above mentioned problems is needed.

1.3.4 GL system with less cost

There are lot of GL system are present but they are very costly for example Oracle financials which is best creating reports and supports financial tracking effectively and efficiently but have cost in millions of the Dollars and it is simply unaffordable for middle level organizations. Similarly other solutions exist on low cost but they have their own issues of performance and low customer satisfaction.

1.3.5 One Application with All functionalities

An application which performs all the tasks of a general ledger is needed. Currently there are many applications present which provides one are two functionalities but did not give the whole solution. For example application for making balance sheet are present also applications for currency conversion are also present similarly application for each single activity of GL is present but a single application which provides user with all those functionalities is not present.

1.4 PROPOSED SOLUTION

As mentioned above the problems which exist for financial management systems, XFB will provide solution to all those problems, as it has all features required for managing financial data of any organization.

The following are the benefits for any organization which will use our system.

- XFB will finish paper work. No one has to keep the record in papers as it becomes tidy and time consuming and also data can be lost easily.
- XFB is a low cost application. In country like Pakistan where it is difficult for people to buy the expensive software, our system suits them. It is cost efficient and fewer resources are required for its maintenance.
- No other software will be required to do things. So it will be easy to train the user of system. The problem of learning and configuring multiple systems will be solved.
- Our application is secure and scalable. With increase in number of users there will be no problem. Every user is authenticated. As financial data should be secured from unauthenticated exposure so it is important to secure the data so that a user who has not privileges should not be able to access the data.

1.5 PROJECT SUCCESS CRITERIA

As the project is intended to solve the problem of all organizations who keep track of their fininacial data, so success of project will be based on level of acceptence of solution, If system is easy to use and understandable to users and also it performs all the required opertions in efficent and effective manner. The security of system will be a key featue to decide whether project is success or not as fininacial data is confidential so it should be secure.

1.6 PROJECT OVERVIEW

After observing different organizations which include organization of different type like life insurance, Universities, Banks etc… we are giving them a software based solution which will help them in managing their financial data.

In existing solutions, the user have to use different software who perform different tasks for him, for example inventory management, financial reporting, account payable, account receivable etc…

Now, we are giving them a system which will give precise results by integrating all the sub systems, which previously were maintained and managed separately by organizations.

By doing so we can manage the transactions performed by different components and also we can manage the security of application on a single place. There are many other advantages like testing of the application will be easy, every component or sub system can be tested independently. Scalability can be maintained easily. Multi-language support will provide more easiness for people of different global areas.

1.7 PROJECT SPECIFICATION

When talking about specification of our application, this application is made using web 2.0 so this will definitely run in stsndard browser which supports the web 2.0 componenets. So that no comatatbility problems occur. For this prefered browser is latest version of Mozila Firefox, moreover it can run on internet explorer and Google chrome too. Our application will be deployed on a server and client will be able to access it through the URL.

First user will login to our application through correct credentials (username, password). The user authentication is done on the standard authentication and authorization library named as Java authentication and authorization service (JAAS).

On completion of successful login the user will be redirected to main menu page. On main menu page there will be a lot of options to user with respect to it privileges or roles defined. The menus or options available to a particular group are defined and every user belongs to a particular group of users hence the menus corresponding to that particular group from which current user is logged in are shown.

There will be a super user or admin who will add or remove users from a group. He will have all the privilligas like create, upadate, delete, grant privillages or revoke etc… for adding a user to other group admin will be requested to make chnages and he will do it.

User can perform action on system and it will result in database transcation. For examplke user can add new bank recepeit, currency type, segment type etc…

The project is broad and include many interesting features like financial report writer which will provide reporting facility. On the basis of these reports an organization can check where they are standing now. They can check in which month there account receivables are greater or in in which month they need more inventory, all these things leads towards decision support system.

As project is lagre we are making some components of the applicationn which include :

- Main menus setup
- Books
- Calednders
- Currencies
- Segments
- Qualifiers
- Segment values
- Code combinations
    - Financial report writer is also part of the application but currently we are focusing on above mentioned parts.

1.8 Scope

The following are requirement for system

- Meta Framework
- Main menus setup
- Books
- Calednders
- Currencies
- Segments
- Qualifiers
- Segment values
- Application should be developed using ADF and JDeveloper

1.9 OBJECTIVES

The objectives of application are as below

- System should be useful and useable both
- System performance should never degrade whether load increased or not
- System should be robust and effective
- Less training would be required for a new user to use it
- System UI should be user friendly
- System should be up-to-date in terms of database values
- System should be secure so that no data is shown to others and a person having particular privileges can do specific operations assigned to him/her

1.10 PRODUCT COMPONENTS

Our application has following components:

- Meta Framework
- ADF Security
- Menus and Options based on User privileges
- Global menus
    a) Add / Delete / Update/ Create Menu
    b) Preview Menus
    c) Re-order Menus
    d) Hierarchy in tree
- GL component
    a) Add / Delete / Update/ Create Component
    b) Pages based upon dynamic regions
    c) Re-order components
    d) Visibility through icons
    e) Report printing through print button
    f) Commit and Rollback buttons to make changes permanent or discarding them
    g) Master detailed views for entries in currencies and some other pages

1.11 TOOLS AND TECHNOLOGY USED

1.11.1 Oracle JDeveloper 11g Release 2

Oracle JDeveloper 11g R2 is a free IDE provided by Oracle Corporation, it simplifies development of Java, XML, JavaScript, and service oriented Architecture (SOA) and Java EE (J2EE) applications. JDeveloper offers complete end-to-end development to Oracle Fusion Middleware and Oracle Fusion Applications with support for the full development life cycle.

1.11.1.1 Key Features

Some of the core features of JDeveloper are as:

- Java SE 6 Support
- Integrates the complete development environment
- Deployment and management of application
- Code editor and navigator
- Its studio edition provides support of ADF framework
    - ADF is a Framework built on the top of JSF
- Some other features are described as below:
    - Xml support
    - Web 2.0 development support
    - Web Services
    - EJB, JSF and Toplink support

1.11.1.2 Advantages:

The advantages of using Oracle JDeveloper R2 are as:

- Recommended by oracle community for ADF development
- Drag and drag ADF components with easiness
- Fast development time and better code fragmentation
- Free IDE so can be used for development by just downloading it from oracle website
- Like other development tools JDeveloper also covers complete software development lifecycle which include designing, coding, testing, and deploying the application.
    - For designing JDeveloper provides the support of UML and database diagram composing options. For coding it provides standard MVC way to code e.g. web content will be in different package, classes will be in separate packages. For testing the application it provides ANT support. For deploying the application it allows to integrate with Weblogic server and other servers too.

When talking about resources for learning and support for tool being used by us then there are large numbers of resources available some are as

- **Resources for learning and support:**

http://www.oracle.com/technetwork/developertools/adf/learnmore/index101235

ADF Community has provided step by step tutorials for guidance of new users and to support the developers in best way, code snippets along with working

examples of the major components are available freely; proper documentation of each and every component is given. There are many other tutorials available on the website of oracle, which are timely updated to provide support to developers.

- **Question Answer forum (OTN):**

https://forums.oracle.com/forums/ is there to support the developers in case of some problem. Developer can post his problem on website and the answer is provided by some other developer or some book other, the system of answering is very good as correct answer contains points so every developer tries to give the correct answer to get maximum points.

As help is web based, so all questions remain posted on website and most of the time a question is already answered and developer find it by little bit searching.

1.11.2 Oracle Weblogic Server

It is one of the industry's best application server used for building EE applications and also used for deploying the enterprise applications. It lowers cost of building, maintain and deploying applications, it increases the scalability which is a vital need of our application.

1.11.2.1 Key Features

Key features of Weblogic server are given below:

- Transformation for both XML and non XML data is possible
- Integration control for file, email etc…
- Proper documentation and help is available
- Better performance than other servers
- Developer can easily perform different tasks on it

1.11.2.2 Advantages

Advantages of Weblogic server are given below:

- Simplified mechanism for security

- Security can be integrated easily through security realms

- Load balancing, availability and scalability can be better achieved
- Simplified mechanism for development, deployment operation and administration
- Best for application made in ADF and recommended by Oracle community

1.11.3 SQL Developer Data Modeler

1.11.3.1 Features and Advantages

Advantages and benefits of data modeler are given below:

- Easy to use visual tool
- Glossary can be made in this
- Can Generates DDL
- Simplifies the database creation process
- Changes are easy to made in this

1.11.4 Oracle Express Edition (XE) 11g R2

1.11.4.1 Features and Advantages

Features and advantages of Oracle XE are given below:

- Strong documentation and strong community for help named as APEX forum
- It is scalable and it can be deployed anywhere on laptop or servers which are stand alone
- Deployment process is very simplified
- Server side validation, processing and querying is available
- It is web based, which makes it more easier to use

1.12 Summary

As described above our project contains Oracle technologies for development, designing and deployment of our application. Oracle technologies are more reliable to use and performance of these applications is well known to everyone. So we are using Oracle technologies for our application.

Chapter 2

LITERATURE REVIEW

2.1 INTRODUCTION

In this chapter we are going to tell some already existing financial software available in market and then we will discuss their advantages and disadvantages.

2.2 TOP COMPETITOR (Financial Applications)

There are lot of system available for keeping track of financial reports but these system have limitations in them and they have many flaws in it, we discussing some of them here.

2.2.1 ARIA

ARIA is an inventory managing software build in PHP this software is open source and available on the website of ARIA however this software have many problems in it for example security is not present in it, system in not scalable and it contains bugs in it too and provides a very limited solution.

2.2.2 PeopleSoft Financial Application

Peoplesoft General ledger is an other tool used for maintaing the financial history this contain many modules in it including account payable ,account receivable , asset management, expense management ,General ledger management etc. besides this Peoplesoft GL system have very robust security for example it have row level, query security, network security, database security etc. the problem with this system is that loading the data is very costly operation for this system. So many servers are needed for operating such system which becomes very costly and small organizations can not offered such systems.

Following is the snapshot of one of the image of software mentioned.

Figure 2-1

2.2.3 Oracle Financials

Oracle financials is also a financial application build by oracle. This application is very extensive one and provides solution to almost every financial solution including GL. It contains more than 200 entities in it and have very good security mechansims implemented in it. The problem with this system is that with such huge number of functionalities is not easy to maintain and some of the functionalities might be useless for clients and also its cost is in million of dollars so small or middle organizations can not afford this system. More over employees training for using this system is extensive.

2.2.4 Oracle Forms Based Application

Some application exist which use Oracle forms for financial reporting. As oracle forms are slow and system made by Oracle forms are not user friendly because these contains little options for user interface desigining. Bulit-in security security mechanisms are not present in these. Moreover paradigm is shifting from Oracle

forms to other technologies like JDeveloper etc. which contains so many options for making user interfaces. Systems built in new technologies have easy maintenance , scalable and built-in security mechanisms are present in these.

2.3 Conclusion

Financial Application are widely used so these should to support the multi-language and multi currency support in it because many people prefer to use this system in their own languages. And by providing support of these functionalties less training of the users would be required. Now days very few systems provide funtionality of multi language and multi currency by providing these functionalities users satisfaction level will increase.

Fusion technologies are very famous all around the world now days. These technologies helps in building applications on the standards. By building systems on standards it becomes easy to make change in system. Moreover security, UI design and task flows are present to help deeloper. For building UI ,Rich client and Oracle faces are used commonly and these have many components to make user interface more attractive and interactivity to system is made simple and better.

Survey tells that in Pakistan there is need of such system which can provide financial reporting functionalities. As Pakistan is a backword country here speed sheets and small softwares are used for financial reporting. They need system which can help them and provide comprehensive solution for problems.

2.4 RELATED TOOLS AND TECHNIQUES

Fusion technologies are very famous all around the world, devlopment is easy in these technologies is very easy. In comparsion to ADF there are many other framwork like JSF, Sturts, Spring along with hibernate could be used for making this application. Likewise there are other development tools other JDevloper like Eclipse and Netbeans IDE exists which could be used for developing the application.

2.4.1 Struts Framework

Strut is a J2EE based framwork which enables rapid development and testing of application.

For building our application Strut along with hiberanate was also an option. Here are some features of Strut framework.

Features

- Custom class based controlling components
- Population of Java Beans through HTML form elements and their validation
- XML based configuration for all underlying components
- Full support for internationalization through provision of resource bundles for all the HTML elements
- Tag libraries for easy integration with your existing JSP's and
  An elegant way of handling form related errors

2.4.2 Netbeans and Eclipse IDEs

Netbean and Eclipse are alternative IDEs which can be used for the Development of ADF systems. The problem with them is that they don't have simple drag and drop of web2.0 components like JDeveloper. In JDeveloper the default setting is such that code is separated in model and controller while in Eclipse or Netbean you have to make separate folders by yourself so due to this easiness JDeveloepr is preferred.

2.5 Why ADF preferred over other Frameworks??

ADF is preferred over other frameworks because of many reasons and some of them are written below

- Due to rich faces components the system look and feel becomes awesome
- Access to database tables through entity objects is very simple
- Page navigation handling is simpler and can be done easily, it contain more navigation options than JSF

- Built on the top of technology stack

2.6 Why JDeveloper preferred over Eclipse & Netbeans??

JDeveloper was preferred over Eclipse and Netbeans due to following reasons

- Easy drag and drop of components available
- Packaging seems simple and code remain well organized
- Better control over system and changes can be made easily
- Supports for development of ADF Mobile, ADF faces, ADF Swing etc…

Chapter 3

METHEDOLOGY

3.1 Introduction

In this chapter we are going to explain the overall architecture of our system, activities done and we will tell how our system is implemented.

3.2 Process

The process of using system is step by step and we will be explaining them as following

Step 1: The user will login to our system by inserting hi login credentials assigned to him/her.

Step 2: After completion of login process user will be forwarded to main menu page where different operations can be performed by him.

Step 3: The user can add the new currency type by clicking new button and can also delete or update already presented currencies. Similarly he/she can add new bank receipt or delete existing bank receipt.

Step 4: After performing some operation user will commit the changes and then these changes will be performed in database. When user will refresh the page the changes will be shown to user.

Step 5: After performing one action user can do whatever he want, he can move to other page and then can perform some task of his choice.

Step 6: After the all these tasks he can logout and he will be redirected to login page.

3.3 SYSTEM ARCHITECTURE

We have followed a layered architecture in developing the system. We have developed the Meta framework on the top of ADF framework to support our application. And then we have developed individual use cases for the application one by one.

3.3.1 Meta framework

The Meta framework is used to develop the overall application. Meta framework is the standard way of developing of a enterprise application. The meta framework approach provides a control over the of the system.

The Meta framework is built on the top of ADF. Thus combines the power of ADF and adds new functionalities thus achieve higher results which makes it worth more. It will be discussed later in the coming chapters in more detail.

3.3.2 ADF Architecture of Application

Oracle Application Development Framework follows a standard MVC architecture. ADF has four layers namely ADFv (ADF View), ADFbc (ADF Business Component) , ADFm (ADF Model) and ADFc (ADF Controller). All of these layers work together and pass information with each other.

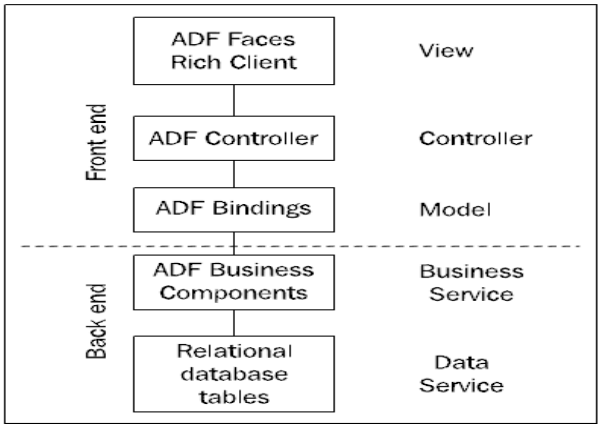


Figure 3-1

3.4 Firebolt Architecture

As mentioned earlier we are using a layered architecture to build the system. It is the simpler approach to divide the application into other layers. It makes it easier to understand application flow.

Architecture of the application refers to application components and their relationships. It is the flow among system modules.

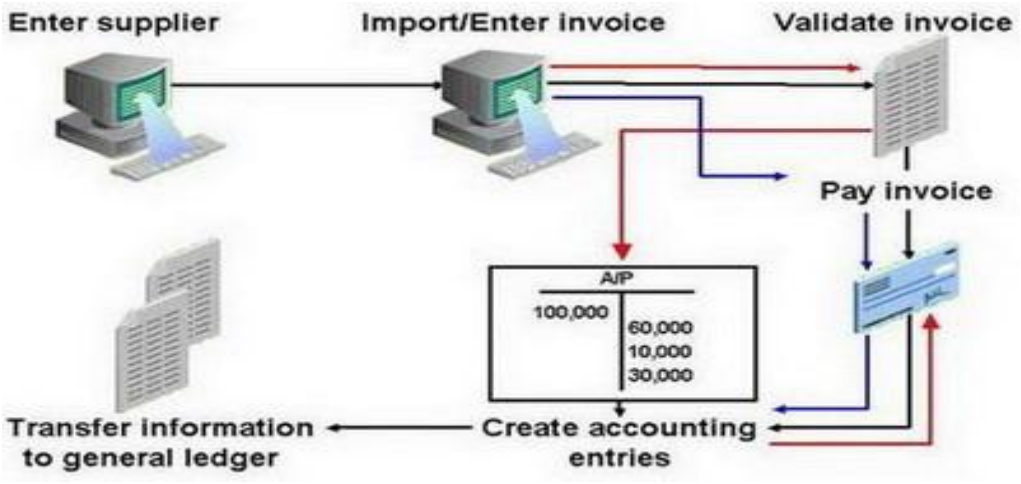


Figure 3-2

3.5 GL Architecture

General Ledger (GL) is the most important part of our system. It is the main point where all entries are gathered. The architecture of GL is as under:

Figure 3-3

3.6 Project Structuring

The system is structured into different layers and these layers are built differently like all the common UI of the application is placed at one place, all common code is handled at one single point.

Below are the different worksapces that implement the different layers discussed.

- XfbMaster
- XfbCommonModel
- XfbCommonUI
- XfbCommonCode
- Sub- Systems

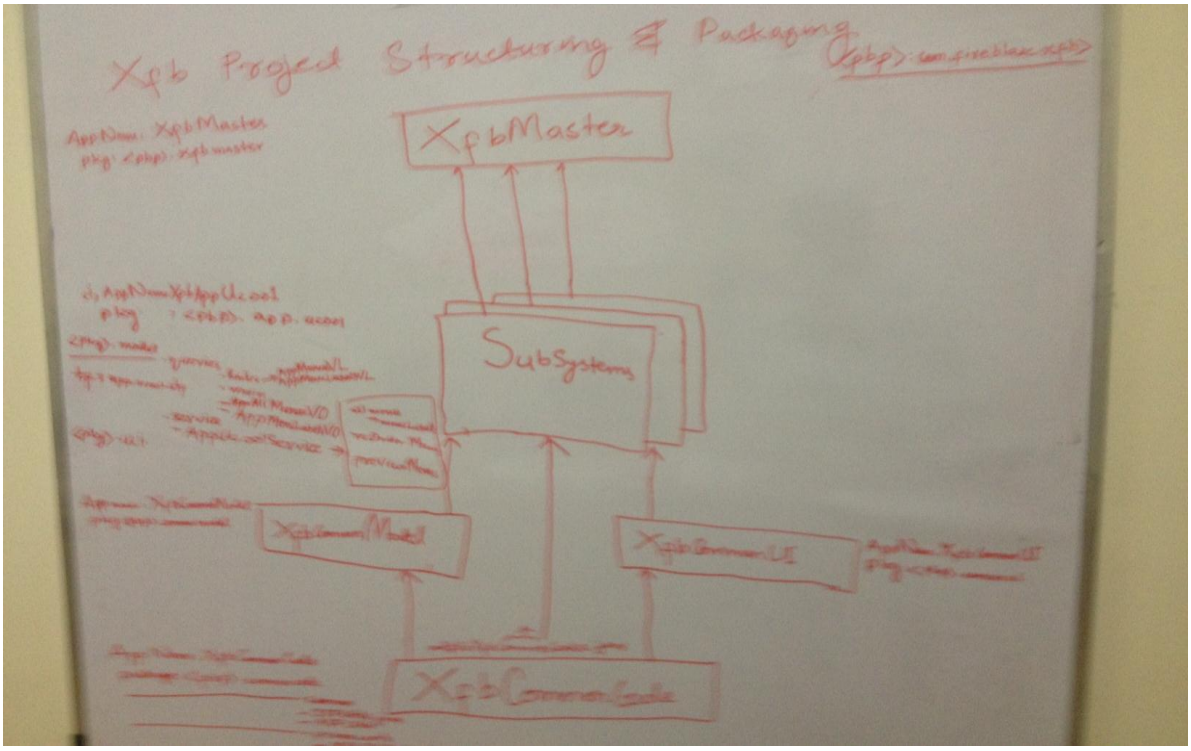


Figure 3-4

The project structuring helps in making the development easier. All modules are built independently thus achieving higher results. These modules then are deployed to a jar file and in last imported to the master workspace, the front module, that deals with client, the master makes use of all imported modules.

Note: Each workspace should be deployed as an ADF library JAR. Workspaces that depend on objects from other workspaces will then import the latest ADF Library JAR released by that workspace. The subsystem workspaces will include the ADF libraries released by the three common workspaces, and the master workspace will include all the subsystem workspaces.

CommonCode workspace is where you place your framework extension classes and any utility classes you develop. It will have framework buffer classes, util classes (AdfUtil,JsfUtil) etc. These classes will be used by other work spaces.

CommonUI workspace is where you keep all the common elements that define the visual identity of your application like skins, page templates, and page flow templates.

CommonModel is where you keep all of your entity objects for the whole application. This will include Entities, Views and application module that will be used across the application.

Subsystem workspace are use case or user story like financials, account payable, receivable, control panel etc.

Master Workspace is where the build and configuration manager puts everything together. This workspace depends on all the other workspaces and contains no code of its own.

Note: com.fireblaze.xfb is the project base package (pbp) for the application where *com* refers to the company, *fireblaze* is the company name, *xfb* is the name of the application which means Firebolt X.

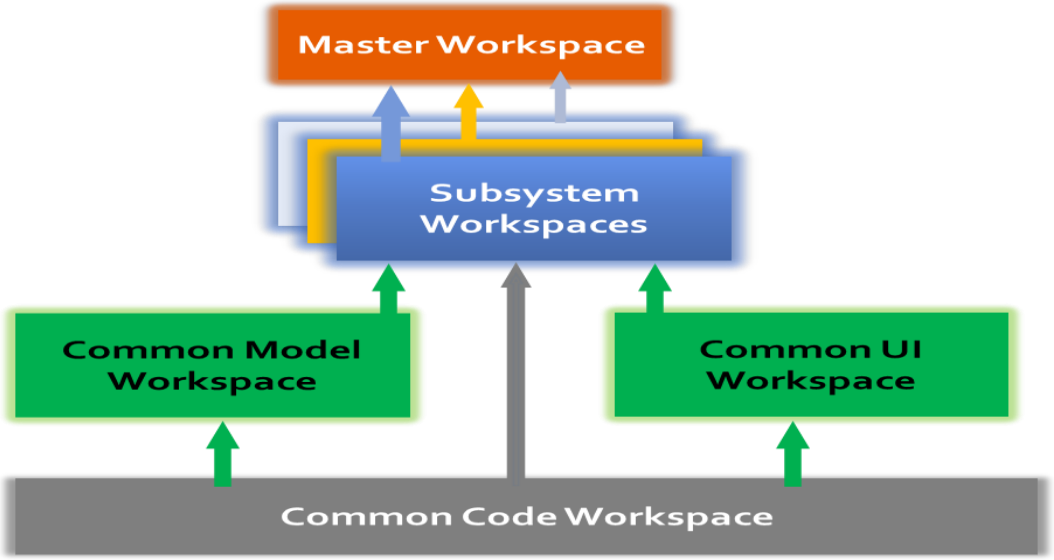


Figure 3-5

Chapter 4

IMPLEMENTATION

In this chapter execution of different system modules, use cases and scenarios is discussed. The chapter describes the implementation of the system overall.

The system mainly compromises of different modules which are described here.

4.1 Meta Application Framework

Developing a framework is the standard way of development of an enterprise application. The approach provides a sophisticated control over the making of the system.

The Meta framework is built on the top of Oracle Application Development Framework (ADF). Thus combines the power of ADF and adds new functionalities thus achieve higher results which makes it worth more.

Also the Meta framework made for enterprise applications helps in standardization of the application. The application is made into different components thus now we only need to change at one place. We now just need to update at one single point and the rest will be updated. It is a great feature to have.

A framework on the top of ADF to develop same kind of applications helps in minimizing the time to develop next application thus saving a lot of development time in making enterprise applications.

It also helps in reusability of the system. As application requiring same functionality no longer would need to write all this code and work again they just need to import it. And reuse it in the system, thus achieving the same functionality.

The Meta framework is not helpful in only reusability but it is also helpful in many ways as it supports following features:

i. User management

ii. Menu management

iii. Security management

iv. Function management

v. Bookmark management

vi. Language management

The Meta framework makes it easier for the developer to develop the system comfortably. It is saves a lot of development time and provides reusability too.

The Meta framework ERD is used to make meta framework. The ERD takes care of all the features the framework needs. The ERD helps in creating user, roles, adding user to particular role; creating objects, pages, takflows, adding objects to pages, adding pages to particular taskflows, giving permissions to roles on different object on different taskflows. It also adds user in organization. It also creates menus and menu labels. It also supports multi language so different labels for different languages of one menu will provide internationalization (i86n).

The ERD helps in making the Meta framework for the application. The comprehensive ERD helps in understanding the Meta framework and the system. It helps in clear understanding of the application what is it going to do.

Below is the ERD which is used to make the meta framework for the application.

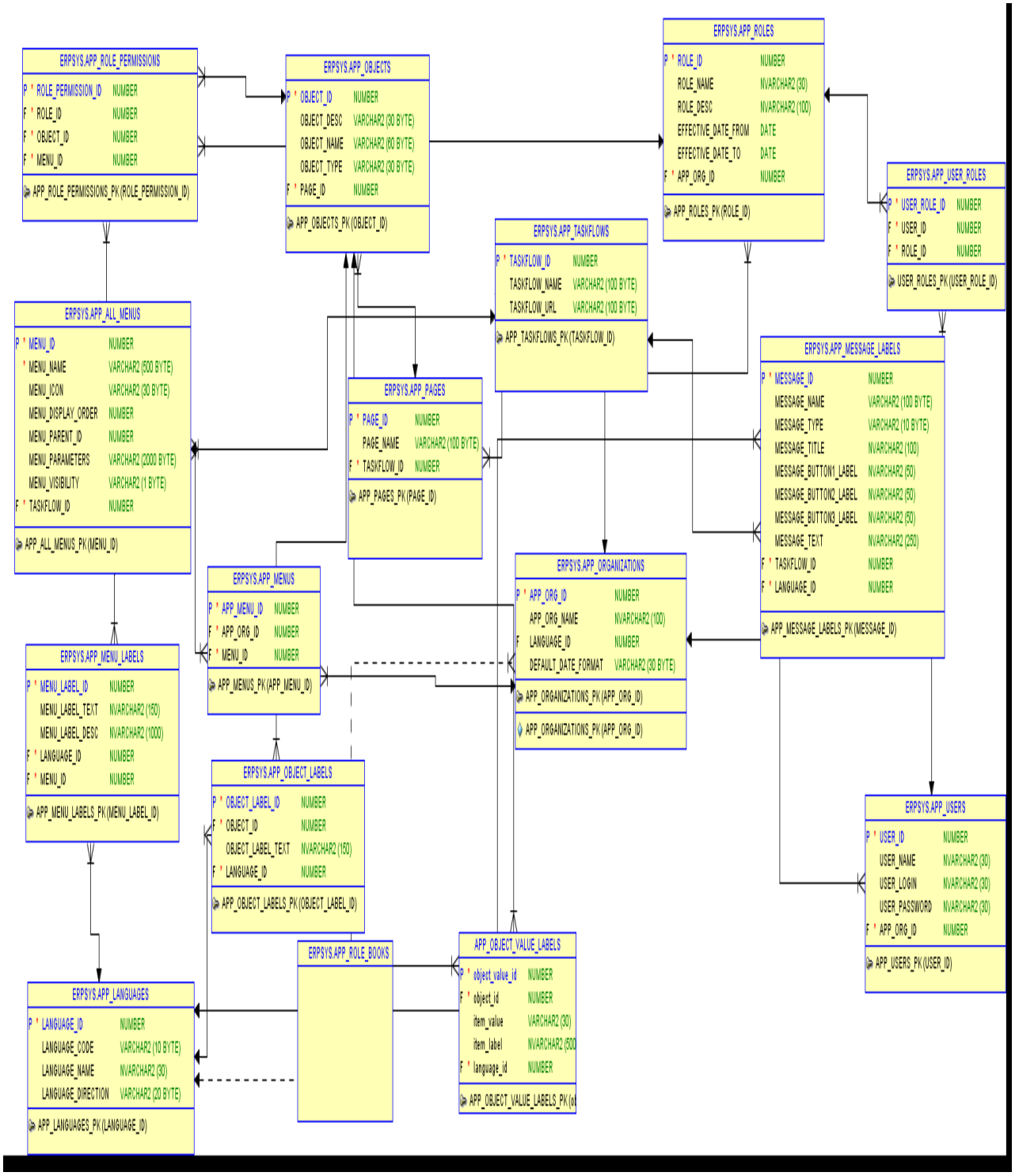


Figure 4-1

4.2 Login Module

The login is the essential part of any application where you want to distinguish between users and show users their particulars based on their username.

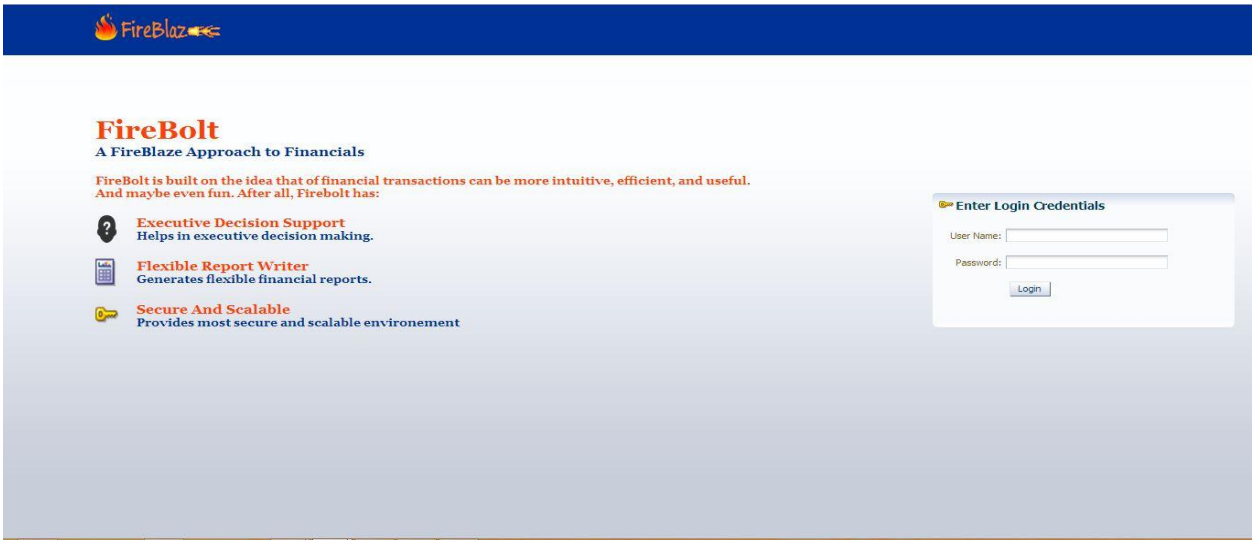


Figure 4-2

The login module of the firebolt is made to secure the system thus only authenticated users can log into the system and access it.

The login process depends on Web Logic Server (WLS) which is responsible for authenticating users. The WLS is configured with the system to authenticate users trying to get into the system.

The users first enter their usernames and password for authentication which goes to WLS for authentication, where appSysSecurityProvider, a SQLAuthenticationProvider, validates user based on provided credentials , it then returns the result back to AuthenticationHandler bean which checks if the attempt was successful or not.

```
<sec:authentication-provider xsi:type="wls:sql-authenticatorType">
  <sec:name>myCustomDBAuthProvider</sec:name>
  <sec:control-flag>SUFFICIENT</sec:control-flag>
  <wls:data-source-name>frameworkDS</wls:data-source-name>
  <wls:plaintext-passwords-enabled>true</wls:plaintext-passwords-enabled>
  <wls:sql-get-users-password>SELECT PASSWORD FROM USERS WHERE USER_ID = ?</wls:sql-get-users-password>
  <wls:sql-user-exists>SELECT USER_ID FROM USERS WHERE USER_ID= ?</wls:sql-user-exists>
  <wls:sql-list-member-groups>SELECT GROUP_ID FROM USERGROUPS WHERE USER_ID = ?</wls:sql-list-member-groups>
  <wls:sql-list-users>SELECT USER_ID FROM USERS WHERE USER_ID LIKE ?</wls:sql-list-users>
  <wls:sql-get-user-description>SELECT DESCRIPTION FROM USERS WHERE USER_ID = ?</wls:sql-get-user-description>
  <wls:sql-list-groups>SELECT GROUP_ID FROM GROUPS WHERE GROUP_ID  LIKE ?</wls:sql-list-groups>
  <wls:sql-group-exists>SELECT GROUP_ID  FROM GROUPS WHERE GROUP_ID = ?</wls:sql-group-exists>
  <wls:sql-is-member>SELECT USER_ID FROM USERGROUPS WHERE GROUP_ID = ? AND USER_ID = ?</wls:sql-is-member>
  <wls:sql-get-group-description>SELECT DESCRIPTION FROM GROUPS WHERE GROUP_ID = ?</wls:sql-get-group-description>
  <wls:password-style>PLAINTEXT</wls:password-style>
  <wls:sql-create-user>INSERT INTO USERS VALUES ( ? , ? , ? , ?, ?)</wls:sql-create-user>
  <wls:sql-remove-user>DELETE FROM USERS WHERE USER_ID = ?</wls:sql-remove-user>
  <wls:sql-remove-group-memberships>DELETE FROM USERGROUPS WHERE USER_ID = ? OR GROUP_ID = ?</wls:sql-remove-group-memberships>
  <wls:sql-set-user-description>UPDATE USERS SET DESCRIPTION  = ? WHERE USER_ID = ?</wls:sql-set-user-description>
  <wls:sql-set-user-password>UPDATE USERS SET PASSWORD = ? WHERE USER_ID = ?</wls:sql-set-user-password>
  <wls:sql-create-group>INSERT INTO GROUPS VALUES ( ? , ? , ?)</wls:sql-create-group>
  <wls:sql-set-group-description>UPDATE GROUPS SET DESCRIPTION = ? WHERE GROUP_ID =  ?</wls:sql-set-group-description>
  <wls:sql-add-member-to-group>INSERT INTO USERGROUPS VALUES( ?, ?)</wls:sql-add-member-to-group>
  <wls:sql-remove-member-from-group>DELETE FROM USERGROUPS WHERE GROUP_ID = ? AND USER_ID = ?</wls:sql-remove-member-from-group>
  <wls:sql-remove-group>DELETE FROM GROUPS WHERE GROUP_ID  = ?</wls:sql-remove-group>
  <wls:sql-remove-group-member>DELETE FROM USERGROUPS WHERE GROUP_ID = ?</wls:sql-remove-group-member>
  <wls:sql-list-group-members>SELECT USER_ID FROM USERGROUPS WHERE GROUP_ID = ? AND USER_ID LIKE ?</wls:sql-list-group-members>
</sec:authentication-provider>
```

Figure 4-3

If it was a wrong attempt it prompts on the login page to tell wrong username or password was entered and if login was successful it redirects user to home page.

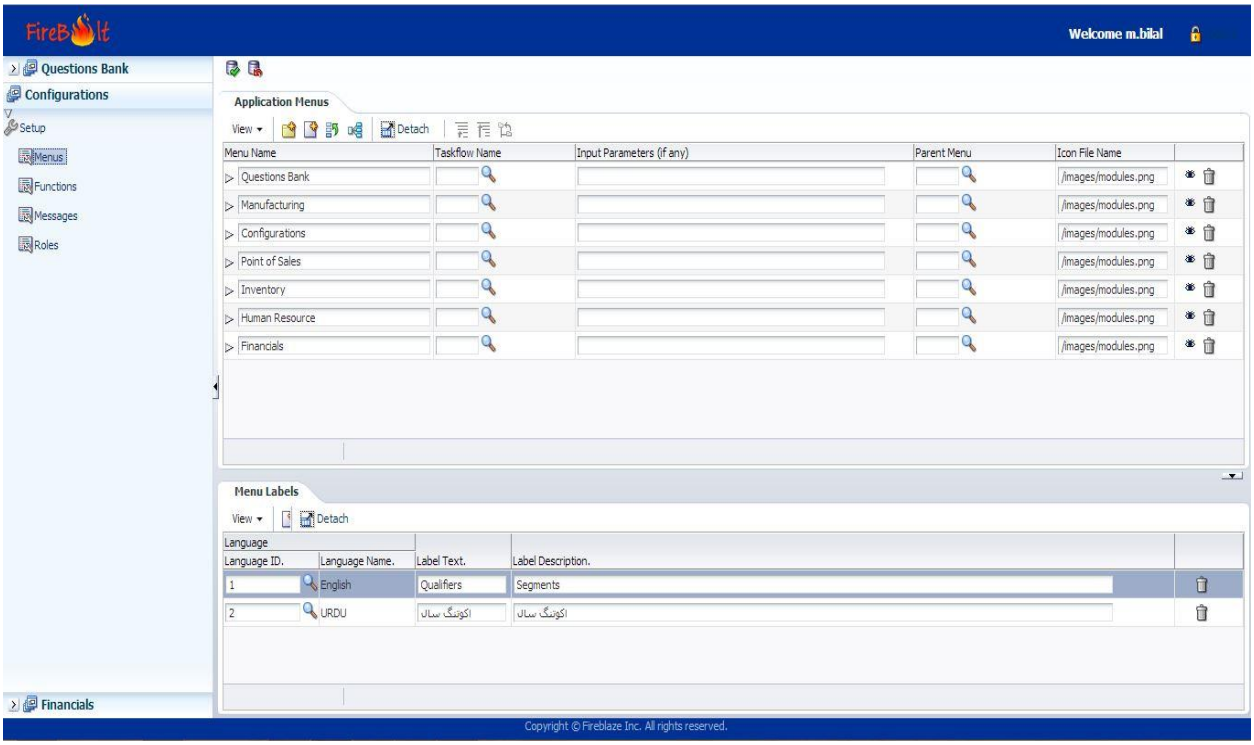


Figure 4-4

4.3 Role Management

In this module, management of user and roles is done. A user is added to a particular role and it can also be removed from it. We can create a new role, we can delete it also.

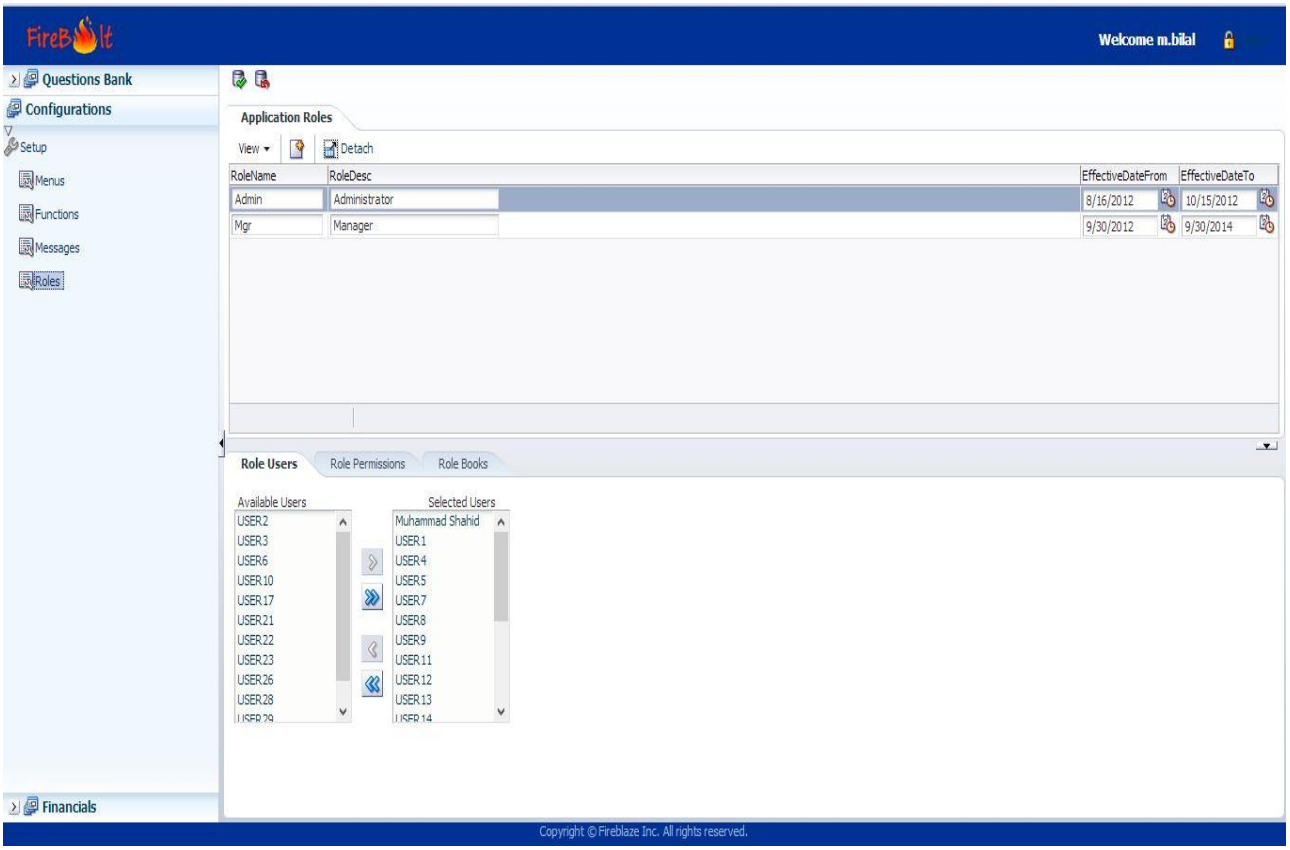


Figure 4-5

Furthermore, permissions of a role are also setup here. We grant one role set of permissions on objects and thus on this basis particular objects are visible/executable by the users in that role. We can also set the effective from and of the roles too.

Also the management of role books is done in this module of the system. Overall this setup module is the key in setting role/user permissions for the application. It is this part that secures a normal user from accessing admin kind of control. That is it assigns specific details that belong to the specific user.

4.4 Menu Management

Menu management is the important aspect of the enterprise application. It is this component of the system that manages all application menus that will be available to users.

In this module, menus are managed. We can create root menus, child menus belonging to particular menu. We can add taskflow URL to the menus here, specify parameters (if there are any), we can assign icons to the menus or change them,

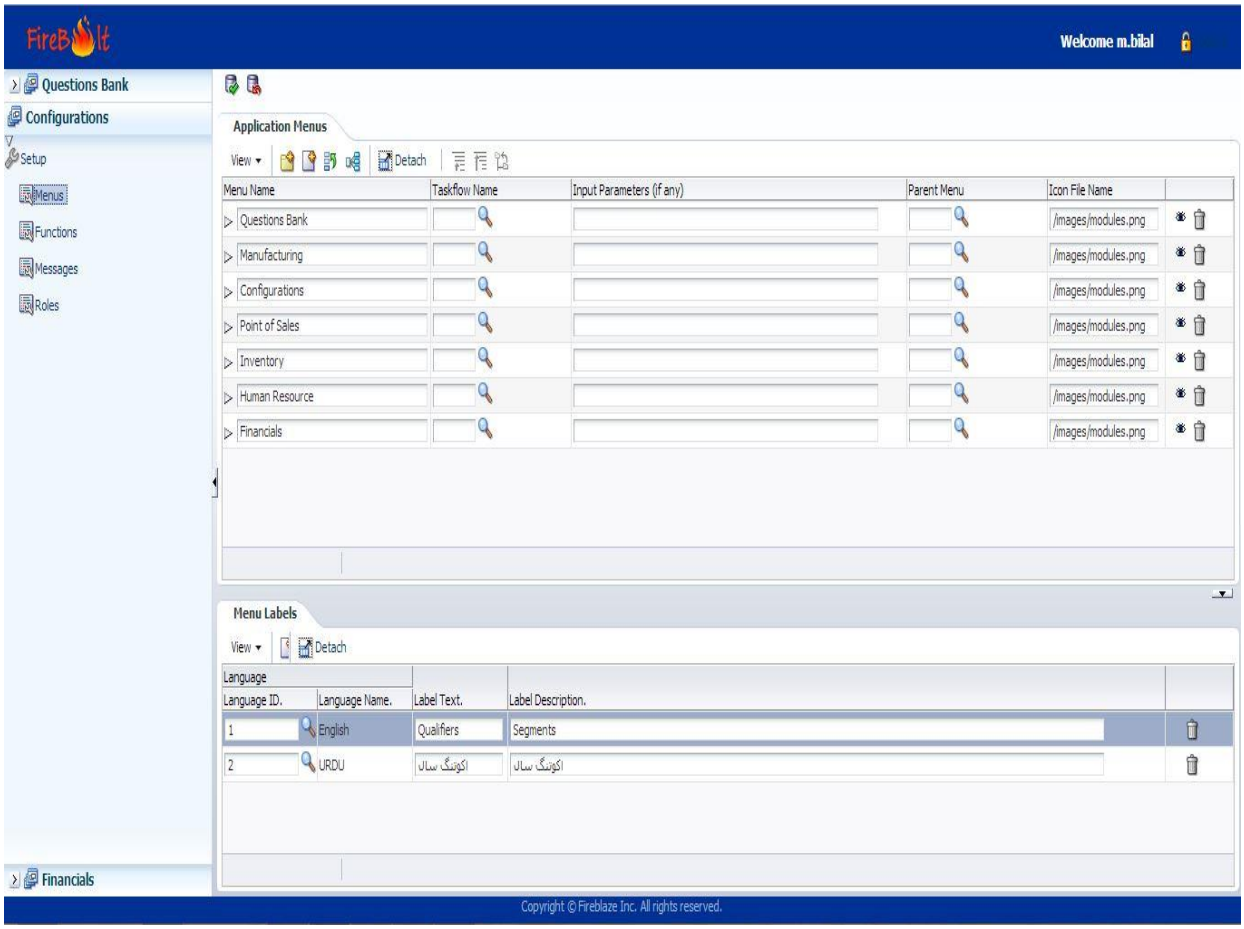


Figure 4-6

We can also set if the menu will be visible of not. Also we can delete any root menu or child menus. We can also rename application menus here and change their child nodes.

Menu labels are also managed here. We can assign more than one menu labels to one menu in different languages like one in English and other label in Urdu, thus achieving internationalization of the application. We can also set description of the label.

We can create multiple labels and delete them too. We can also set language in which the new label is going to be set which is selected by a list of value that lists all the languages one organization supports or has.

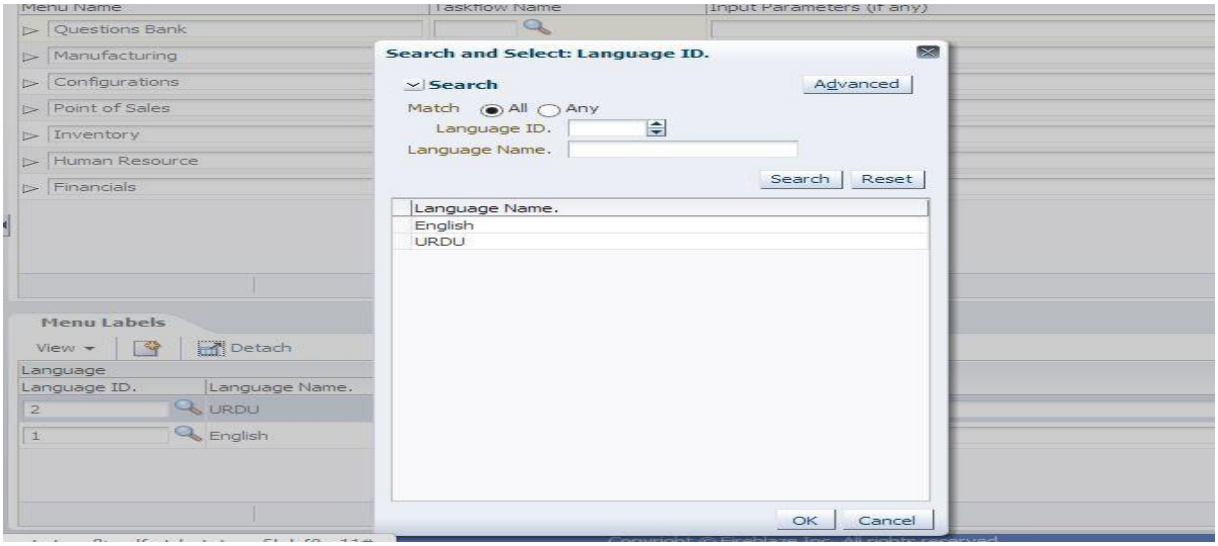
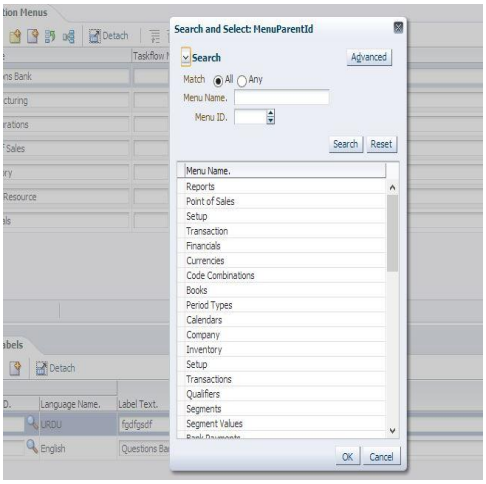
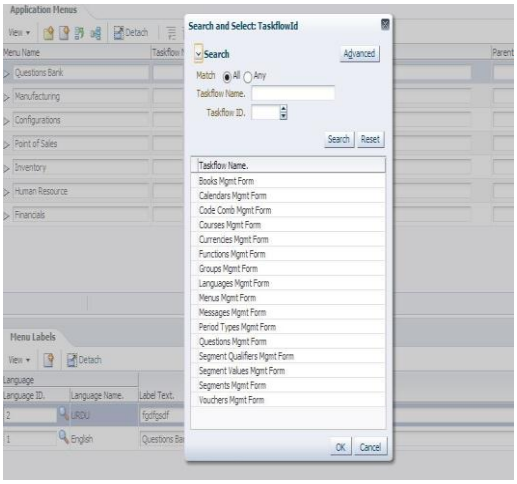


Figure 4-7

The module has also other list of values also. Among them one is list of values for setting parent of the current selected menu. Thus current menu becomes the child of the menu which is set here. Another list of values (lov) is the lov for taskflow which sets taskflow to the currently selected menu.





Thus current menu is assigned to this taskflow that means the menu URL will be set by this taskflow (function). Thus clicking on this application menu assigned taskflow URL will be opened.

Re-ordering of the menus is also the major requirement of the menu management. So here we can also re-order menus. We can set order of menu to top, end, previous or next.

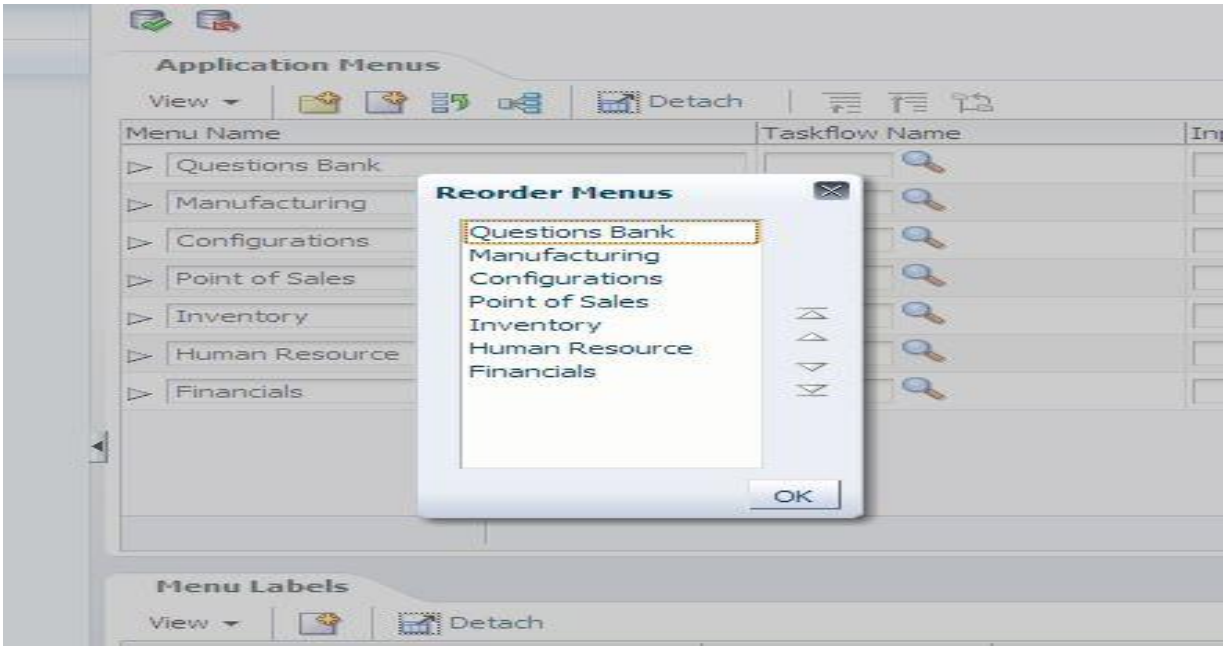


Figure 4-8

Another requirement that is mostly required when it comes to management of menus is previewing the all application menus in different languages.

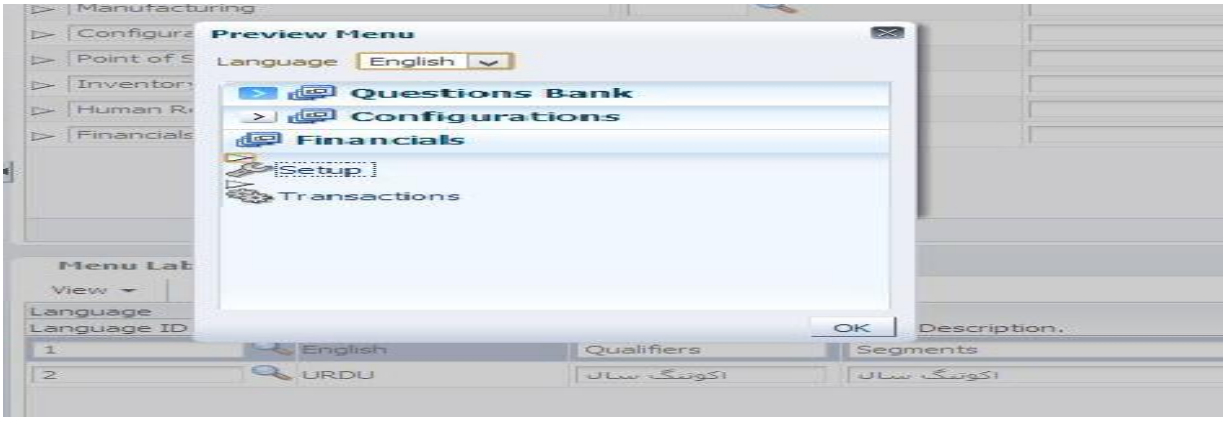


Figure 4-9

We can preview all application menus in different languages by changing the language of the application. Thus all menus will be displayed in that language menu labels.

4.5 Functions

In this module of the application, functions or taskflows are managed. It is this module that handles the management of taskflows that are assigned to application menus and used in role permissions

We can create new functions / taskflows; update their taskflow name and taskflow URL. We can also delete any taskflow in this module.

4.6 Currencies

Currency management is the important part of the financial application. This module that deals with, management of currencies and exchange rates.

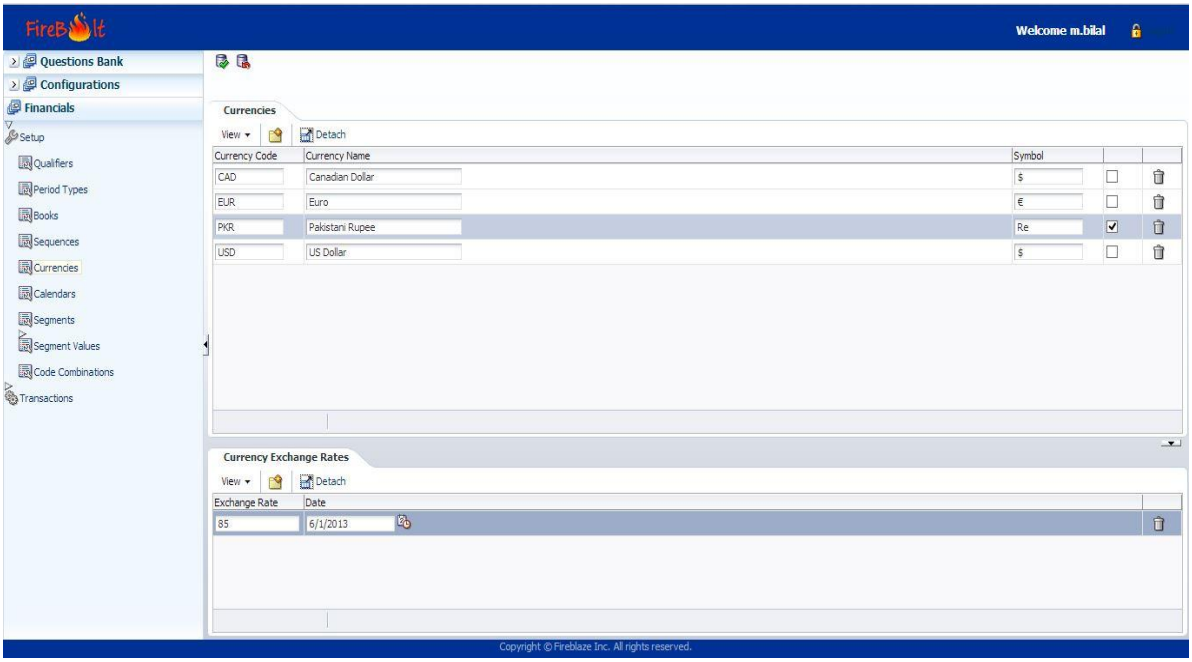


Figure 4-10

In this module multi-currency feature of the application is achieved by letting users create multiple currency but setting one base currency for the organization. We can update currency name, symbol, base or no base currency. We can also delete any currency previously created.

The exchange rates of the currencies are also handled in this module. The master detail form does it all. When a currency is selected in master form, the detail form shows its currency exchange rates. In currency exchange tab, we can create a currency exchange entry, update previous rate and update the effective date. We can also delete any previous exchange rate entry.

4.7 Books

In this module books management of the financial application is done. It is this component of the system that deals with handling of book types and their

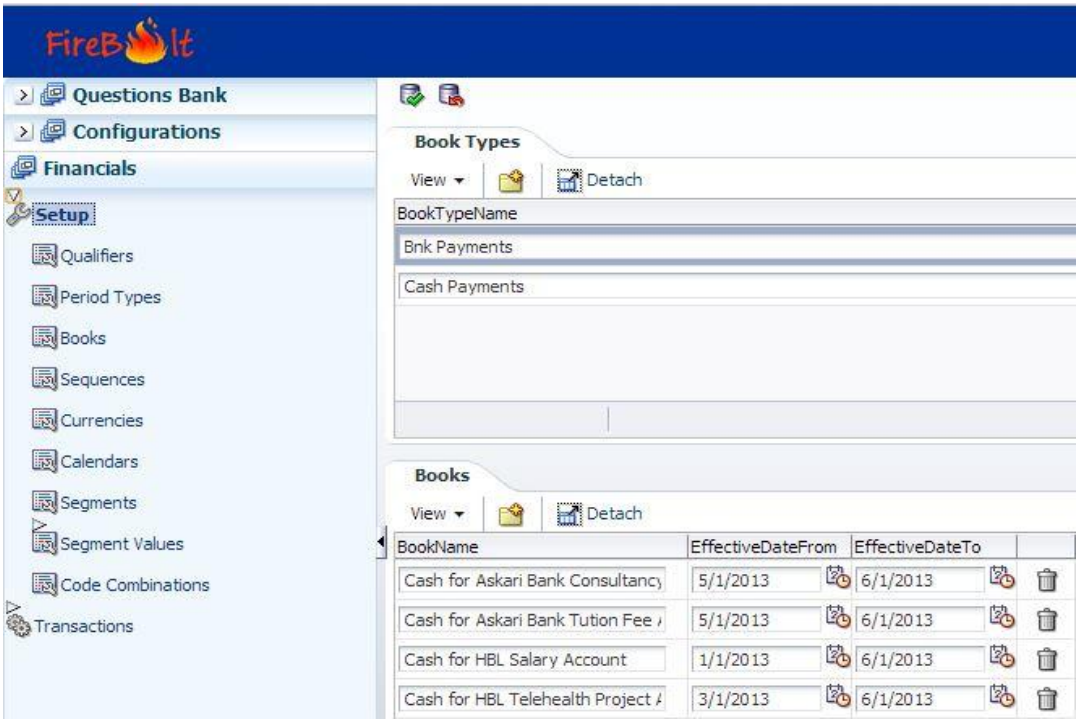


Figure 4-11

Books. The master detail form helps in managing book types and books. When in master form, a book type is selected; detail form shows its relative books.

In this module, we can create new book type like Bank Payments. We can also delete any previous book type.

Furthermore, we can also create books in book types. We can update book type name, the effective date from and effective date to. We can also delete a book.

4.8 Period Types

Period types are the important module of the application which deals with the management of financial periods. It held a lot of importance as it is the point where all financial periods are created as well as managed. Because financial periods are of great importance so this module kept great importance.

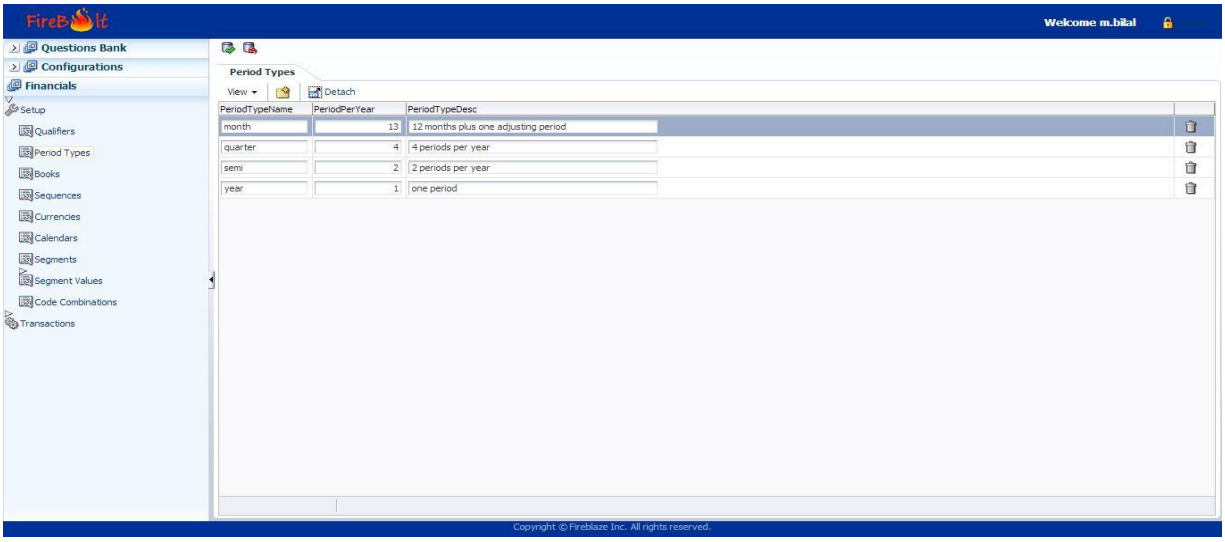


Figure 4-12

In this module, we can create a new financial period and we can delete previous periods too. Furthermore, we can also update the current financial period, its name, periods per year and its description.

4.9 Calendar

This module deals with the financial year and financial periods management.

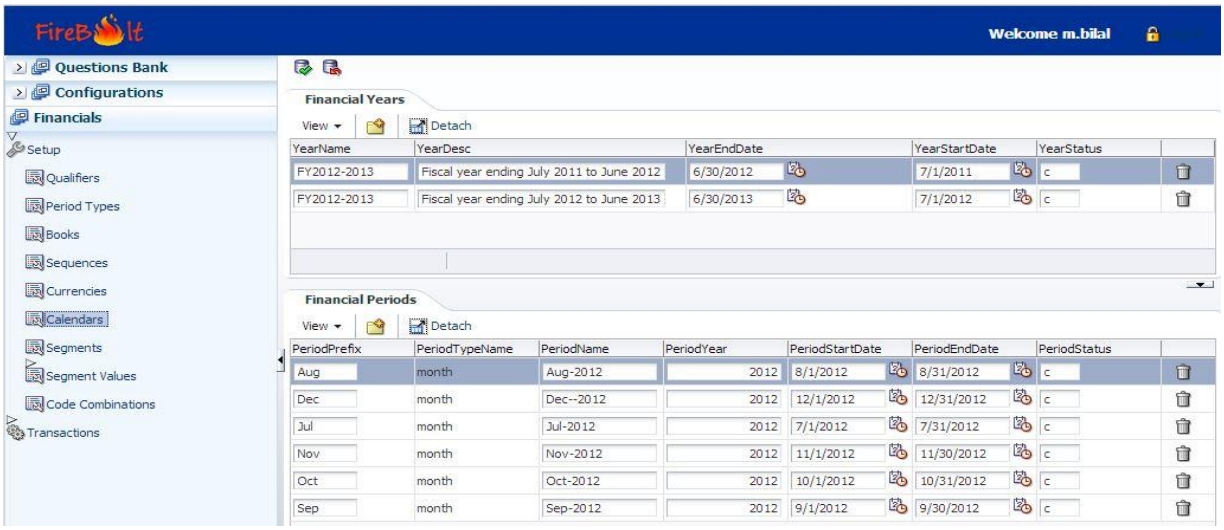


Figure 4-13

In this module there is a master detail form, when we select a financial year in the master form, the detailed form shows its relative financial periods and its details.

Also in this form we can create new financial years and we can also delete previously made years. We can also update any financial year, its name, its current status, its description, its start and end date.

Furthermore, we can also create financial periods in the detailed form where we can also delete existing periods. We can change the details of the financial periods also. We can change its name, its prefix, year, start and end date and status also.

4.10 Segment Qualifiers

In this module we take care of segment qualifiers and its management.

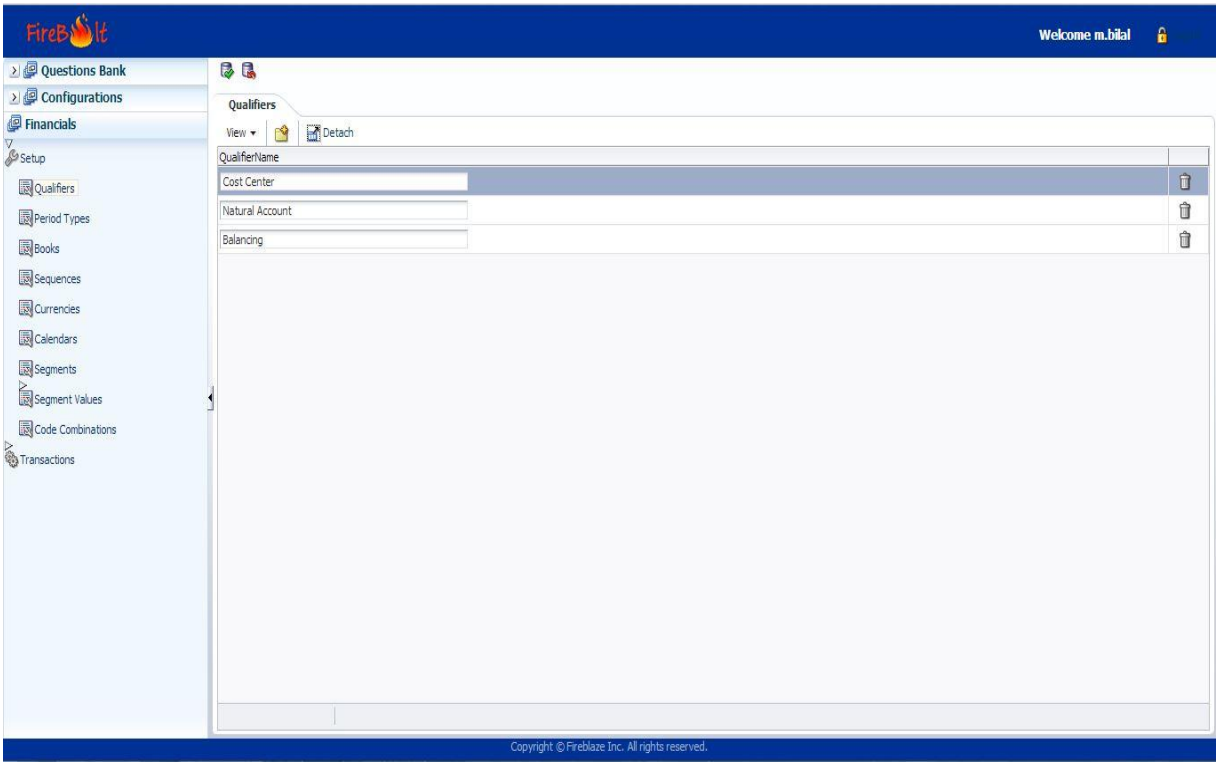


Figure 4-14

In this component, we can create qualifiers and we can delete them. These qualifiers will be available in segment management form in the list of values. There

are only three possible values for segment qualifiers namely *Cost Center*, *Natural Account* & *Balancing.*

4.11 Segments

This module is the most important part of the GL component of the system. In this module all the management of segments and its detail is done.

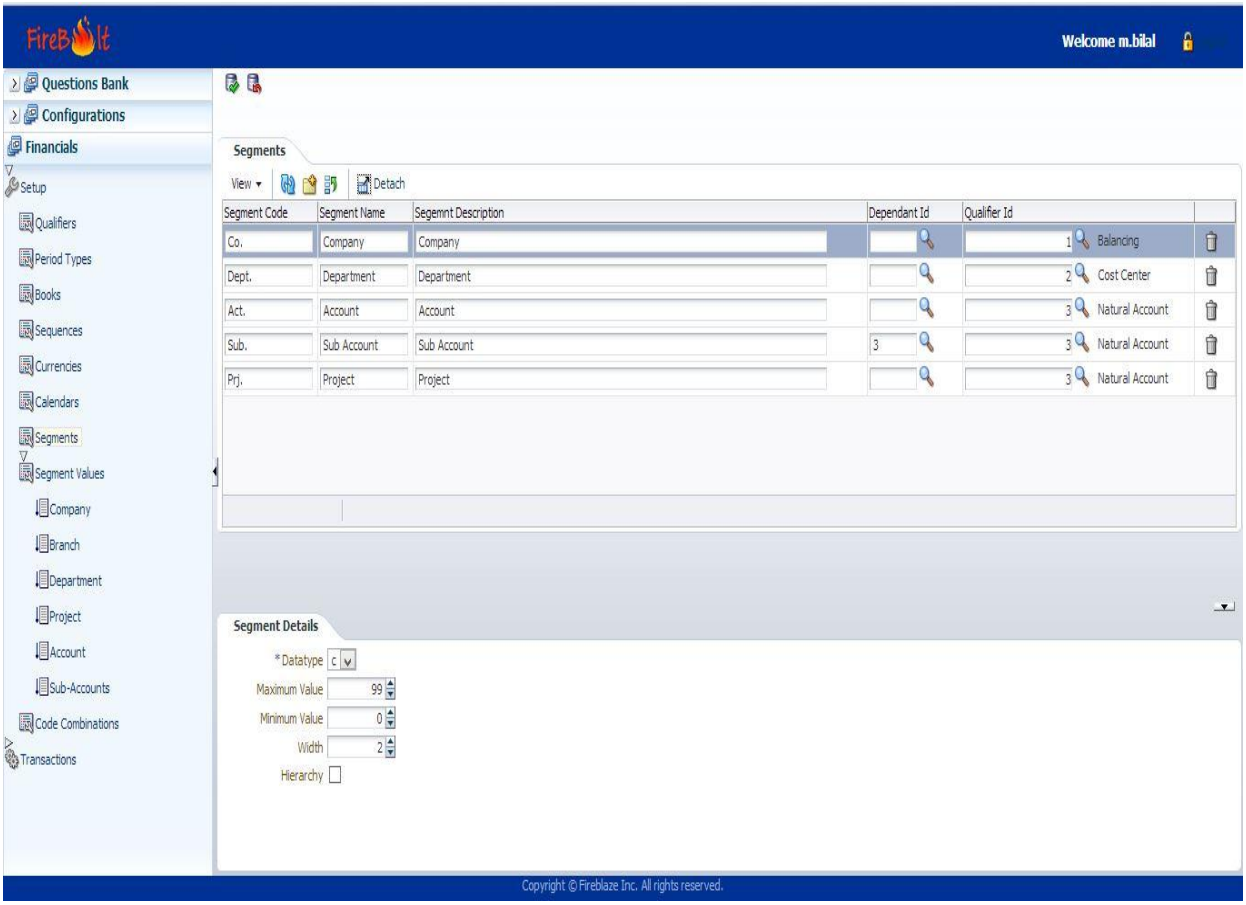


Figure 4-15

In this part of the application segments can be created and we can also delete them. There are two forms; first form presents the important information while the second only provides the extra details of the selected segment.

We can also update the information of segments, its name, its code, description, qualifier, dependent segment and datatype, max, min value, and width. There are two lovs in this module; one is for listing dependent segment id and other

is to select segment qualifiers that were made in qualifiers form described on previous page.

4.12 Segment Values

In this module, segments that were created in segments form are managed. It is very exciting and interesting to know that there is only one form that implements this use case.

Figure 4-16

Segments created in segments form now appear as a new form in the segments values form. Only one form is there to cater all these segments only values are changed at runtime depending upon which segment is clicked. For example if company is clicked then data of company is shown, if branch is clicked then branch's data is appeared.

Chapter 5

DISCUSSIONS

XFB is a web based application made in Oracle XE and Application development framework and the server used was Weblogic all of relation design was made in Oracle Data Modeler.

ADF along with Oracle XE is best thing for Java EE developers for development of web applications. The life cycle and process of development is very time taking and tough. Full command on ADF is necessary to build full end to end system. The system made in ADF has great look and feel and easily understandable for new users.

Chapter 6

CONCLUSION

XFB is component based application, it contains several modules and every module can be tested separately some of the modules are as

- Segments
- Segment values
- Currencies
- Bank receipt

Every module has its own complexity and can perform different truncations by form of that particular module. The transaction type is create, update and delete.

In master detailed forms there is different process for every operation and different level of transaction module made available.

The data shown to every module is similar as data present in database, data synchronization is key part of our system.

Chapter 7

RECOMMENDATIONS

We found Fusion web applications are very robust, scalable and have user friendly interfaces. If you are making an enterprise level application then ADF is one of the best options for making that.

ADF takes a longer time in learning, so if your application is small or medium level than you should prefer some other technology but for large level applications it is highly recommended.

ADF contains 250 rich faces component which help in setting best possible user interface. These components contain amazing built in functionalities that would be time consuming to develop using JSF or other framework.

Chapter 8

REFERENCES

8.1 Learn ADF

http://www.oracle.com/technetwork/developer-tools/jdev/overview/index-100269.html

8.2 Online Help System

https://forums.oracle.com/

www.oracle.com

8.3 Learn Financial Applications

http://en.wikipedia.org/wiki/Accounting_software

www.en.wikipedia.org/wiki/General_ledger

http://financialaffairs.georgetown.edu/procedures/Introduction_to_Peoplesoft.html#Overview

www.aria.com

Chapter 9

APPENDICES

Here is the image of Oracle JDeveloper 11g 11.1.2.3.0 which we used for developing our application.

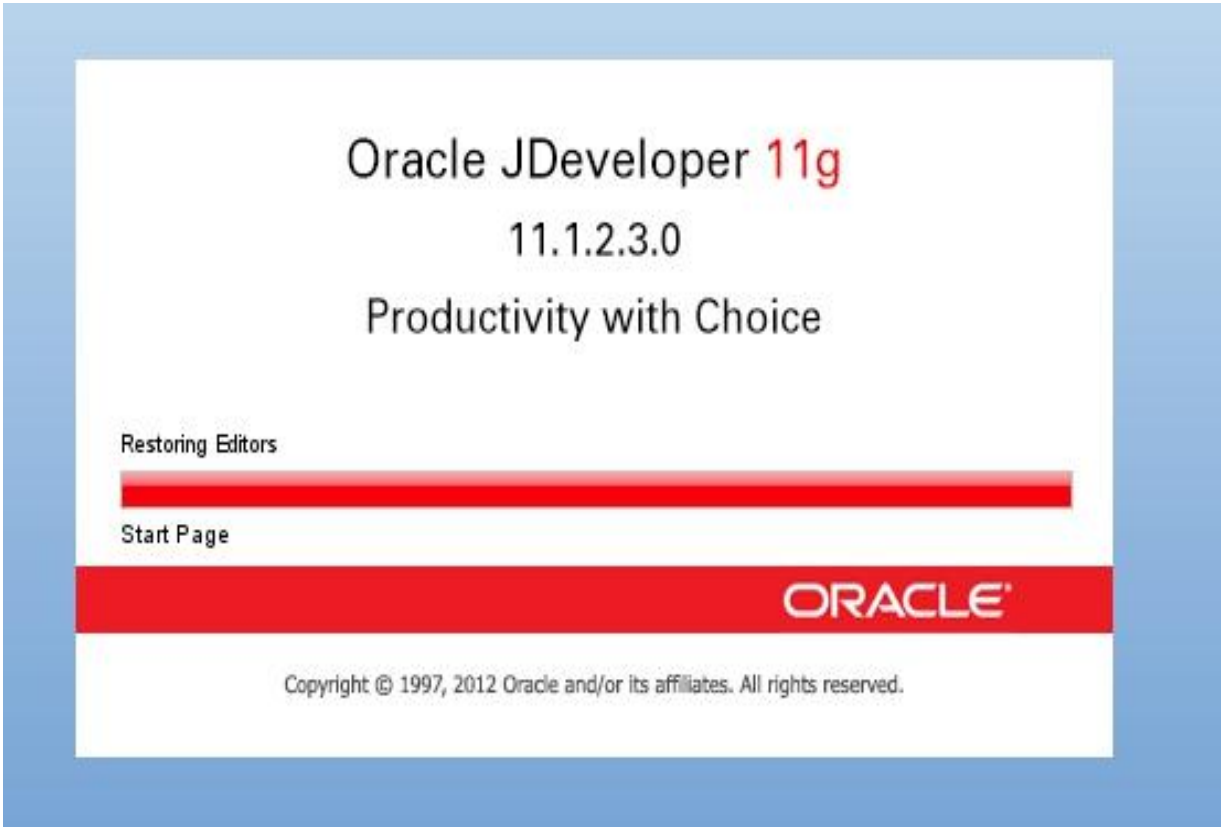


Figure 9-1