**Sanchaar**

**A**

Project Work

Submitted as Major Project in Bachelor of Engineering

Submitted to

## RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA BHOPAL (M.P)

****

Submitted By

**Pankaj Tiwari(0105CS111068)**

**Rajat Mangal(0105cs111086)**

**Satvan Parihaar(0105CS111099)**

Under the Guidance of

Dr. Praveen

## (Department of Computer Science & Engineering)



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

**JUL-DEC 2014**



***CERTIFICATE***

This is to certify that the project entitled **“Sanchaar”** being submitted by **Pankaj Tiwari,Rajat Mangal & Satvan Parihaar** student of **6th**  Semester, in Computer Science & Engineering have done their work as MAJOR PROJECT for Partial fulfillment of the degree from RGPV, Bhopal (M.P.) is a record of bonafide work carried out by them under our supervision.

###### Dr. Praveen Prof. Amit Shrivaastav

###### **Guide Head**

###### **Department of Computer Department of computer**

**Science & Engineering Science & Engineering**

###### Prof. Sanjay Sharma/Prof Monika Agrawal

###### **Project Coordinators**

**Acknowledgement**

**We would like to place on record my deep sense of gratitude to Prof. Amit Shrivastava HOD-Dept. of Computer Science Engineering OIST Bhopal , for his generous guidance, help and useful suggestions.**

**We express our sincere gratitude to Dr. Praveen, Dept. of Computer Science Engineering, OIST Bhopal, for his stimulating guidance, continuous encouragement and supervision throughout the course of present work.**

**We also wish to extend my thanks to Prof. Sanjay Sharma, Prof. Praveen Yadav and other colleagues for attending our seminars and for their insightful comments and constructive suggestions to improve the quality of this project work.**

**We are extremely thankful to Dr. V.K.Sahu, Director OIST Bhopal for providing us infrastructural facilities to work in, without which this work would not have been possible.**

**Pankaj Tiwari (0105CS111068)  
Rajat Mangal(0105CS111086)  
Satvan Parhihaar(0105CS111099)**

**Table of Contents**

* **LIST OF FIGURES……………………………………………………………5**
* **INTRODUCTION……………………………………………………………..7**
* **BACKGROUND AND LITERATURE SURVEY…………………………..8**
* **PROCESS MODEL…………………………………………………………..14**
* **DESIGN……………………………………………………………………….17**
* **TECHNICAL DETAILS…………………………………………………….25**
* **CODING………………………………………………………………………27**
* **TESTING……………………………………………………………………...61**
* **SCREEN LAYOUTS…………………………………………………………67**
* **FUTURE ENHANCEMENTS……………………………………………….77**
* **CONCLUSION……………………………………………………………….79**
* **REFERENCES………………………………………………………………..81**
* **BIBLIOGRAPHY…………………………………………………………….81**

**List Of figures**

**Figure Title Page no.**

**3.1 Waterfall model 15**

**4.1 Use case diagram 18,19**

**4.2 Class Diagram 20**

**4.3 Sequence Diagram 21**

**4.4 Component diagram 22,23**

**4.5 Deployment Diagram 24**

### ABSTRACT

### Sanchaar is an ERP system specifically made to cater the needs of a college or similar academic Institute.It provides various features that will enhance the productivity of daily works .

### It focuses on various attendance,material delivery and feedbacks releted issues in an innovative and time saving way providing institute management full control and ease over the process.

### Sanchaar system is made in such a way that it can extend with the needs of the institute. Its core API and interfaces are smartly designed keeping in mind the ever changing scenario of modern educational institutes.

### Sanchaar system will help institutes to manage their data, keep their services available at fingertips and making everything possible with minimum human effort.Providing everyone the ease and comfort Sanchaar will help oragnisation to achieve its best performance.

### Chapter 1

### Introduction

### 1.1 Problem Statement Oriental Institute of Science and Technology,An Engineering Institute having more than 3000 students base has been facing a lot of problems in managing the students data at this large scale. The existing paper-work based system is too slow to the new changes,sudden inquiry and its very hard and sometimes almost impossible to get any information on-hand for any query. Institute is facing problems in following contexts: (1) Information about any student is not available on the hand if somebody wants to access it. Concerned body needs to go to respective departments and pull-up the bundles of TG-Files to search for student's data. (2) There is no any proof-based and fast communication medium between students and higher level management authorities which leads to confusion between management and students.Most of the time management can not determine what students actually want. (3) There is no any public-portal to share notices/hall-of-fame or anything conveyable to students. Absence of a common platform makes organisation of events unsuccessful and untouched in terms of student responce. (4) Attendance maintainance is really tough in existing system. Inspite of all the efforts, It leads of inaccuracy. A number of cases are brought to notice when student complaint about their attendance being unresonably less in records. (5) Accounts information is not centrally available which leads to confusion and repetetive work at the time of semester registration and exams. Also there is not any notification system for students to know if they have any dues pending. (6) Taking feedbacks is almost impossible these days. Doing it manually and then reading the feedbacks is really messy and time-consuming job. It also leads to inaccurate results of feedback and somtimes bad consequences.

### Institute needs a software system that can manage all their complications easily and management authorities can focus on real-improvement rather than taking decisions on the basis of assumptions. A management system will lead institute towards better management,better understanding and finally many folds better perforance.

**1.2 Aim:**Our aim is to provide a better technological solution to the Institute to manage its existing work and data-load. We will be constructing an ERP system that will manage all the student's data, all the examinations data, attendance and its statistics and a messaging based interaction environment between students and higher management.

### Also our efforts will be to make sure that system is capable of expanding itself in future so that further extensions and modifications can be done in it depending upon the growing needs of the organization.

### 

**CHAPTER 2**

**BACKGROUND AND LITERATURE SURVEY**

**2.1 Study of current system**In the current system we analysed and inspected and finally extracted these 8 problem points. Our solution will address these problems and further improvements into the system will be based on these core problem points and its solution implementation.

(1) Student's data management  
(2) Student's finantial accounts  
(3) Student's attendance  
(4) Student's requests/complaints/feedbacks {no-record}  
(5) Student's communication with higher authorities and its record keeping  
(6) College level feedbacks and its records  
(7) Centralized information portal  
(8) Notices and other media-delivery

We analysed the whole system and we saw that these are the key factors and stack holding pillers in the informational infrastructure of the institute. In order to improve the quality and standard of the system we will have to improve upon the speed,ease accuracy and feasibility of these processes.

We took each of these factors one by one and studied the current process being followed in the institute to achieve these tasks. We found many flaws in the current system and saw scope of improvements in various aspects. In the next excerpts we are giving the detailed study of the current process being folloowed in the system, the problems associated with current procedures and our proposed solution.   
**2.1.1 Student data management  
Current Process**Every semester student registration is done. In student's registration process student fill an offline student's registration form. In this form students fill some old and some new details. The filled student form is undergone a manual process of revalidation/recheck by authority(TG/HOD). As a proof of fee-deposit student needs to show his/her fee reciepts to the concerned registeration authority(TG). At this point there is no any live communication between the TG and account's department is present and registration solely depends upon the fee-receipt paper proof.  
After registration process a big pile of student's registration form is collected and is kept under scruitiny for the whole semester under any specific supervision.  
What is in the form: (information and its content weightage)   
New pictures - 10%  
Contact details - 10%  
existing details - 50%  
feedback about college - 30%  
<<inode diagram-2 : pi-chart using above details >>  
Above mentioned 30% existing details about student (which are not liable to change throughout the cource - viz Name,Father's name, Mother's name, Home address) are refilled and revalidated again. This leads to wastage of time,effort and storage.

**Problems in it**First major problem around this issue is the cost involved in it. There is cost involved in fresh pictures, form reprinting and TG labour.An average teacher getting 30,000/- per month means rs 166/- per hour, is kept busy for hours in maintaining and compiling the stuff which should be totally automatic. This also waste total system time.Repeted information is collected in the system and practically there is no way to get rid of it.Inspite of all the efforts being done to ensure the process being easy, the major concern here is that the student data is not centrally avilable. **Access and availability**data is sole responsibility of TG to protect and maintain under his scrutiny. TG is the main key person involved here to access studen't data.

* Student - can communicate to TG in order to get(or modify) his information.It involved system boiler-plate.
* HOD: if HOD wans to accees the studen't details then he needs to contact the corresponind TG in oder to wt studen't information.The TG provided him the required information in the compiled formats.
* Director - If director of the institute wants to access the studen't details then he needs to contact the respective department's HOD,then HOD contacts with the TG and information is so forwarded.
* External Authority/TPO -> They contact the Director or the respective department and the information flow continues as mentioned above.<<inode diagram-3 : histogram showing information retrival boilerplate (distance vs time) >>

Process,Storage and Retrival all the three important factors are very slow in the current system. We need to address the issue in such a way that information is processed,stored and respectively formatted without any manual effort,on the click of few buttons and all the time available. We also need to ensure that information is secured and authenticated to the latest as possible.  
**Proposed Solution**To address this problem we are proposing an Online registration system for students as in integrated part of our ERP system.This system will help TG’s to get registration done in minimal amount of time with least effort possible.Software will operate on LAN and student will fillup their registration form. Filled up registration form will be collected to their respective TG’s account where TG will one by one accept their registration or reply with correction measures if registration is not accepted. This will remove all the paper work previously involved in the process and as a benefit of the system,data will available at fingertips for any other application.

**2.1.2 Student finantial accounts  
Current Process**In the current system all the finantial details and records of a student are stored in an Tally ERP double entry based system which disposes its data at the end of finantial year. Students do need to carry their own fee receipt as proof of their fee submission. In case of lost receipt,student may even have to resubmit the whole fee as the system is having no proof of record and student lost its proof too.  
Other than record keeping another problem is notification. To get any updated information about their pending dues, student need to manually query to accounts department. Most of the time it happenes that student start facing some trouble because of some pending dues student is not aware of; So unavailability of such notification is a problem.  
**Proposed Solution**  
To address this problem we are proposing a Notification system as in integrated part of our ERP software. This will help accounts department post the notification to students about their pending dues and record keeping. With this system,students will have their proof that their fee is submitted and carry-your-reciept problem will be solved.Under this sytem accounts deprtment officials will have their special account which will make them able to post notifications about student’s liabilities and dues. As soon as their dues get cleared, they will manually update its status to cleared.  
This will allow accounts department to have a log apart from their existing ERP system. Also student’s will be insured and updated about their accounts status to the latest.  
 **2.1.3 Student’s attendance  
Current Process**In the existing system teachers need to carry their attendance register. In each lecture a roll-call is done to take the attendance. Also a separate register is maintained by TG’s in order to maintain daywise attendance. Compilation of attendance report and mantainance of records is alone so much tedious process that its very hard to think about any utilization of this record in any other purpose. Attendance can be a very important analytics information which can help management to take decisions. Because of poor feeding system,its going in waste.  
**Proposed Solution**To address this issue we are proposing an attendace management system as as integrated part of our ERP. Here teachers will be able to feed their attendance in very efficient and innovative way. This will help system keep records of their attendance and generate intuitive data,statistics and charts. These will help management to take better decisions for student welfare.Further to eliminate the paper work at all, we will develop android applications which can be run in a medium phone/tablet. It will act as an attendance register for a teacher. Teacher will create his/her class. Register the attendance and the data will get synchronized with the server in realtime.   
When student will reach their home,they will get report of their today’s attendance. This will ensure the attendance to be error free and instantaneous.

**2.1.4 Student’s request,feedbacks and complaints  
Current Process**In existing system there is no any proper method for students to complaint about something or raise a request about something. Also feedback of services is not taken which leads to misfunctionaing of services and decay in service quality.There is no record keeping of request,complaints and feecbacks. The information which can lead management to take very important and game changing decisions for the institute is going to waste. We need a proper chanellized system which can regulate all the process of complaints,requests and feedback.  
**Proposed Solution**To tackle this problem in an innovative way we are proposing Online Complaint Cell as in integrated part off our ERP system. Students can raise complaints or request on their individual basis and will get a complaint code,much like a simple customer care system. The underlying details of complaint management can be kept hidden from students and requests will be forwarded to respective departments. Student will see his complaint status pending when his request is under consideration. After completion, status will get updated to completed. This will help management to reach to individual problems and analyse respective department’s tendency and capability towards solving the issues. If a department is taking too much time to solve an issue then management can issue a query against that department and increase its resource. This will help management to fix the nail when its needed and where its needed.

**2.1.5 Communication with higher authorities  
Current Process**In the existing system,communication with higher authorities is almost abscent and student have to take irregular visits in order to communicate about his problems,needs and other issues. Complaint management system solved this on issue level but we need another system where students can send message/mail to their collegues,teachers, TG’s and higher administration and also can reply to sent messages.   
**Proposed Solution**We propose a Messaing system as an integrated part of our ERP system. Each and everyone who has some kind of role in the system will have massaging as a basic facility. He/she will have an inbox and a sentbox. Message facility will be similar to any mail sending utility.(Eg. Gmail).On different locations and events user will be given options to send message to concerned person.  
We are also introducing filtering and broadcasting facilities in the messages,so that higher authorites can send messages in broadcast and groups too. Special types of heuristics will make sure that mesaaging remains healthy and pristine for a college environment.

**2.1.6 College level feedbacks and its record keeping  
Current Process**Taking feebacks in current scenario is almost an impossible task. Till not only manual methods were being used for taking any kind of feedbacks. This was not so much accurate and was availability dependant. Taking game changing decisions based on these feedbacks can sometimes lead to disasterous situations. Also this process is so much effort wasting. A person or team have to manually undergo in each and every feedback (paper) and maintain rough data sheets. This limits the possibility of feedbacks to minimum. A potential resource of important analytics data is going waste in papers.  
**Proposed Solution**To overcome this problem we propose a online Feedback system as an integrated part of our ERP System.Administrators will be able to create feedbacks and students will be able to fill feedback forms. The feedback data will get collected over and will help system to generate various types of reports. This will ease the feedback process will make sure its availability independent.  
**2.1.7 Central information portal  
Current Process**In current scenario students,teachers and other authorities do not have any common platform on where the important news and announcements can be spread. Authorities like TPO have to rely on social media to spread the word about anything. This is not acceptable at all. College needs a portal of itself where all the important details can be posted.   
**Proposed Solution**  
To address this issue we introduce an Central Information Portal as in integrated part of our ERP system. Authorities will have the special rights to post on this portal. Students will be able to see it and post views,comments and votes to it.  
Socializing features will be included on it to ensure any back-response expected.(Optional).  
 **2.1.8 Notice & Other Material Delivery  
Current Process**In current scenario teachers need to send mails to students in order to deliver them any notice or any study material. This is a long and manual process. Also in case of any updation needed, the whole process is needed to be repeated for all the students.This can be so much trouble some the teachers.  
**Proposed Solution**To address this problem we will provide a content management system as in integrated part of our ERP system. Under this, all teachres will be provided with their own blogs where they can maintain their profile,can upload meterials and can post articles. This will lead material delivery to a whole new level. Teachers will be able to post updates about their research,studies and any teaching activity they are conducting. We will provide basic social facilities like commenting and upvoting so that a back response can be collected. **2.2 Proposed Solution**Sanchaar ERP system will address the above mentioned 8 issues on the primary basis in its first release. Its core and API will be so made that its functionality can be extended in future depending upon the growing need of the organization. Sanchaar will address the existing issues and helps institution management to tackle them via new and innovative approaches. Our efforts will be to make Sanchaar a next step in the growth of organization.

**CHAPTER 3**

**REQUIREMENT ANALYSIS**

**3.1 Identify the problem**Following problems are identified in the system during the analysis phase:  
(1) Student registration is a very long and tedious process, it often requires a lot of efforts and time and the resultant data is also in papers which can’t be made available at demand if required in instant.  
(2) After each and every semester, the old registration details are discarded and whole registration process is repeated again. It leads to wastage of resources and repeating of already done work.An ideal system should never do same work twice.  
(3) Students should get finantial notifications into their own accounts in form of messaging or separate notifications. It will lead to less risk and better finantial reponce availability from the student side.  
(4) There must be a pain free attendance maintainance system which also does the tedius work of generating reports and statistics automatically without any human intervention. Also system should act as a warehouse and keep the older records safe and intact.  
(5) Student’s request,complaints and feedbacks should be taken and conserved as they are a potential source of analysis and improvement over the current system. This will help management to take better business decision.  
(6) There should be communication medium so that all the users can send messages to one another in the system and there is no need of separate messanging facility.  
(7) There should be central information portal as well as personal blogs so that students do have better reachability to the professors and departments. Online delivery of notices,circulars and other material in a centralized and combined system will ensure the authenticity of information and trust in the system.

**3.2 Scope of the problem**The above mentioned issues are sitting in the backbone of the system. Because of this latency in the system its quality is decaying day by day. Implementing a new solution can be very difficult in the present scenario. System needs to solve these piller problems first to enhance its productivity and response. Once the basic problems are solved, other problems become minor and solvable. Without the presence of a delivery system an institute in just left way behind in time. Students can’t get informaed to any changes and news propagation is slow. Without a response registering system, management will never come to know what students actually want leading to a confusion uttered situation. Absence of an attendance management system will lead to poor judgement in terms of attendance. Attendance can be a potential source of analytics data. A smart system will always take advantage of any analytics possible in its favour to take better decision.  
Out aim to create Sanchaar system is to facilitate all the process in automated and online way. We will collect all the possible data in system and try to apply business intelligence on it.This will help management to take better busieness decision for student welfare. If the software can help management to achieve its goal, we are very confident it will be a successful market-maker.

**3.3 Requirement Specification**

**3.3.1 Functional Requirements**

**1. Web Client:**

It should be accesible through any common web browser following w3c standards.

**2. Authentication:**

Different type of users should be assigned different authentication levels according to the task and rights assigned to them.A proper rights management should be maintained.

**3. Communication system:**

Communication system should be there which can help different users communicate in

text based media i.e. message between their working hierarchy.

**4. Attendance Management system:**   
should be able to record attendance on daily basis without fail. It should accept data in generic ways so that in future this functionality can be accessed via embeddable devices.

**5.Content Management:**system should be able to handle large amount od data and processing load that its users will impose on it during a busy day.System should be able to manage its content and keep them safe.

**6**. **Extendability**System should be so made that it can be extended and tailored according to the changing needs of the organization.

**3.3.2 Non Functional Requirements**

**1. High Performance:**  
System should be able to handle hight amount of loa on normal and peak times,it should be able to deliver high performance and high data availability.

**2. High availability:**   
System should be able to remain up and runnig always no matter how much load its bearing. It should be able to protect itself from any error in case of high demand and fatigue.  
**3. Secure:**  
System should be highly secure as in terms its dealing with private data of an institute. Data is always crucial and it should be protected at any cost.  
**4. Regular backup facility:**System should be able to backup all the data at regular intervals. In case of any disaster it should be able to restore its backup without delay providing user a continuous undisturbed experience.  
**5.Modular Architecture:**  
Change is the key to growth of an organization and this system should be made modular so that it can adapt to that changing demands. Modules could be added to it and could be edited on defined rules.

**3.4 Identify Users  
3.4.1 Student**Student is the most crucial part of an organization. It is generating data on accounts,registration and social basis. The organisation is liable to provide its performance to its prime customers i.e. students. Students are the main stackholder in the current system. We are programming all the facilities in order to provide seemless experience to the student falling under current system.  
**3.4.2 Teachers**Teacher are the second most data generating people in the system. With posting capabilities provided in the form of blogging gand notice posting, teachers will generate lots of data. Our aim is to provide teachers a seemless experience in terms of facilties of blogging, manage their content and attendance management of their students.  
**3.4.3 TG**TG are responsible to manage data of students and generating daily and monthly reports according to their attendance and academic performance. TG will be accepting students registration and will be sole in-charge of students data assets.  
**3.4.4 HOD**  
HOD is mainly concerned with viewing performace and attrndance reports of students and any complaint submitted in case. We put HOD in higher authorities which will be proided with analytics and report viewing facilities.  
**3.4.5 Other Authorities**  
Director,TPO and other management personal come under other authorizes. They are provided with general features of report viewing and notice posting. In case of any customization, exisint modules will be tailored to meet particular requirements.

**3.5 Feasibility Study  
3.5.1 Time feasibility**We are planning to divide the whole system work into 24 sprints(3 sprints per module). Each sprint will be producing a working module which we can directly deploy on the servers. Although we preent any use of incomplete system unless until the finall load testing of complete system is not done.We will be able to complete the implementation phase in next 6 months of time.  
**3.5.2 Technical feasibility**  
All the technologies required to implement this system are currently existing in the market. There is no need to do special research and development to invent the implementation platforms.We will be usingfollowing technologies of web development to implement our solution:  
**(1) PHP**   
PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. As of January 2013, PHP was installed on more than 240 million websites (39% of those sampled) and 2.1 million web servers. Originally created by Rasmus Lerdorf in 1994, the reference implementation of PHP (powered by the Zend Engine) is now produced by The PHP Group. While PHP originally stood for Personal Home Page, it now stands for PHP: Hypertext Preprocessor, which is a recursive backronym.  
**(2) Codeigniter**CodeIgniter is an open source rapid development web application framework, for use in building dynamic web sites with PHP. The first public version of CodeIgniter was released on February 28, 2006, and the latest stable version 2.2.0 was released June 5, 2014.

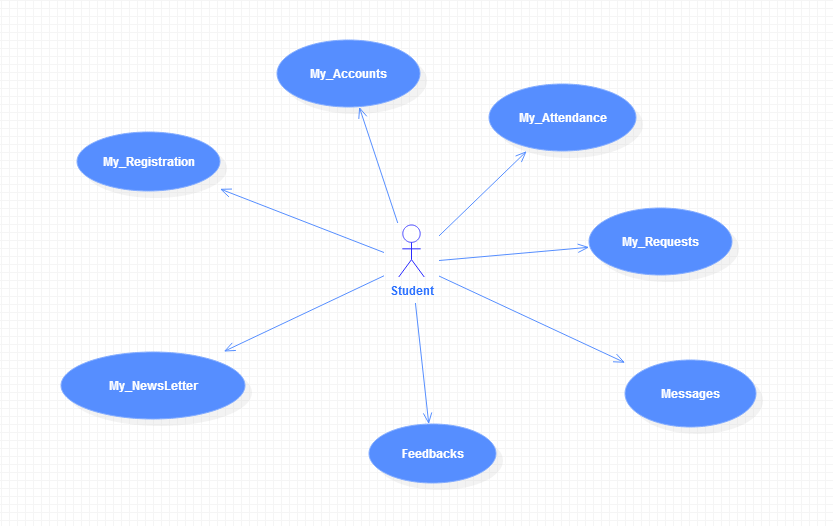
CodeIgniter is loosely based on the popular Model-View-Controller development pattern. While view and controller classes are a necessary part of development under CodeIgniter, models are optional.

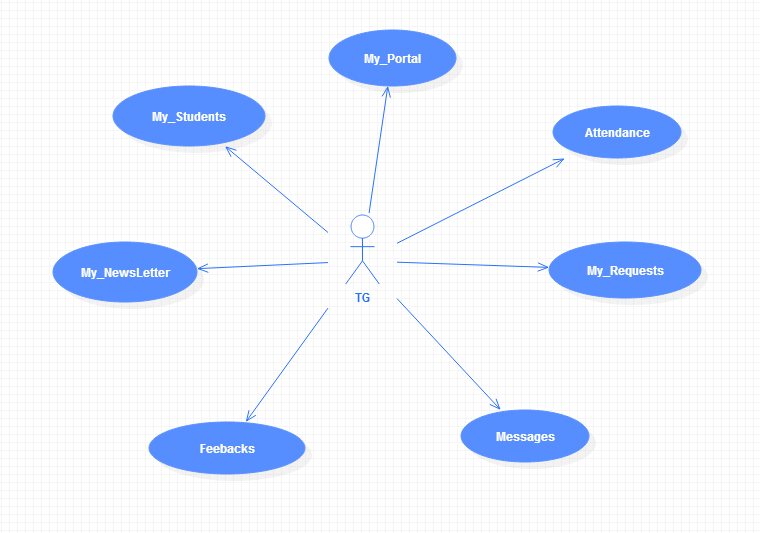
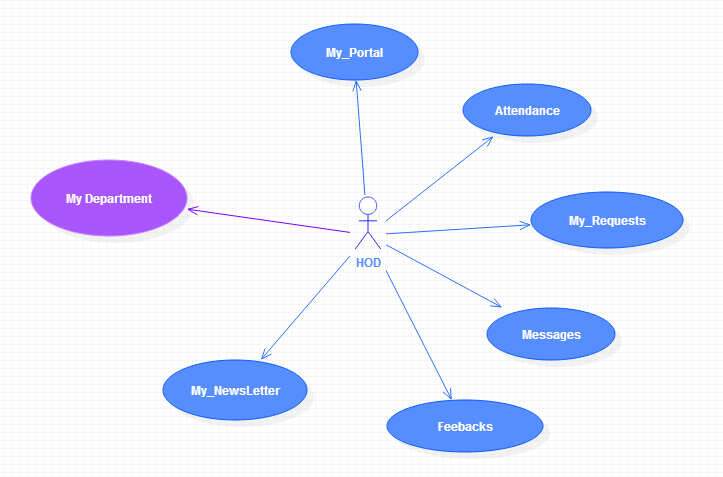
CodeIgniter is most often noted for its speed when compared to other PHP frameworks. In a critical take on PHP frameworks in general, PHP creator Rasmus Lerdorf spoke at frOSCon in August 2008, noting that he liked CodeIgniter "because it is faster, lighter and the least like a framework.".

**(3) MySQL**  
MySQL is an open source RDBMS that relies on SQL for processing the data in the database. MySQL provides APIs for the languages C, C++, Eiffel, Java, Perl, PHP and Python. In addition, OLE DB and ODBC providers exist for MySQL data connection in the Microsoft environment. A MySQL .NET Native Provider is also available, which allows native MySQL to .NET access without the need for OLE DB. MySQL is most commonly used for Web applications and for embedded applications and has become a popular alternative to proprietary database systems because of its speed and reliability. MySQL can run on UNIX, Windows and Mac OS. MySQL is developed, supported and marketed by MySQL AB. The database is available for free under the terms of the GNU General Public License (GPL) or for a fee to those who do not wish to be bound by the terms of the GPL.

**(4) Bootstrap**Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.Bootstrap makes front-end web development faster and easier. It's made for folks of all skill levels, devices of all shapes, and projects of all sizesBootstrap easily and efficiently scales your websites and applications with a single code base, from phones to tablets to desktops with CSS media queries.Bootstrap is open source. It's hosted, developed, and maintained on GitHub.Millions of amazing sites across the web are being built with Bootstrap.  
**3.5.3 Economic feasibility**This project will be economically possible because its completely built under open source technologies. Using open source tehnologies reduces cost of project as open source technologies are free of cost and are competitively used in developing high performance solutions. We are using agile development methologies which will make the building of this big software easy and systematic.   
**3.5.4 Legal feasibility**This project is being developed using Open Source technologies which makes us legally free of any licensing cost. Adantage of using open source technologies is that we can focus more on technical aspect of our project without worrying about the legal aspects of using the technology.  
**3.5.5 Operational feasibility**We will be developing this system on personal laptops(4gb RAM,Intel 2.5 GHz) machines.

The system will be deployed on a RedHAT 7 Linux Server being operated in institute. It contains 8gb RAM and 16 cores of 2.5 GHz processors. The whole serer is backed up with 1 TB of SATA 1.5 Stoarage disks. Regular backup of whole database is taken manually each satarday. This configuration is more than enough to serve 1000 users at a time for a web application. We are confident that our system will run healthy on such configuration.   
**3.5.6 Social feasibility**As a proof of trust this system will be deployed on fake data for some time and automated tests will be performed on it. If system runs fine and is able to pass all the benchmarks and criteria defined by our evaluation team then it will be fit to deploy in the institute premices.

**3.6 Object Oriented Analysis Modeling  
3.6.1 Scenario based modeling  
Use case diagram : Student**  


**Use case diagram : TG** **Use case diagram : HOD**

**2.1 Software Requirement Specification:-**

**2.1.1 Introduction:-**

The purpose of designing this software is to provide teacher with all the data and facts related to students sitting at one place.It enables the faculty a better understanding of different students of different semester and branch .

* + 1. **functional requirement:-**

Functional Requirements:

**1. Installable desktop client application:**

Desktop application that can communicate with the remote server and act as desktop

middleware between data entry point and server.This application should contain most of the view related code that is needed to render the response sent from the server in

appropriate way.

**2. Web Client:**

In case of unavailability of desktop application web client will act as rescue.It should be

accesible through any common web browser following w3c standards.

**3. Authentication:**

Different type of users should be assigned different authentication levels according to the task and rights assigned to them.A proper rights management should be maintained.

**4. Communication system:**

Communication system should be there which can help different users communicate in

text based media i.e. message between their working hierarchy.

5. Business logics should be implemented in **the core business APIs.**

* + 1. **Non functional requirement:-**

1.OpenSAP should act as a business application development framework which provided different set of features to make business class applications.

2. It should be leak proof and secure.

3. The Code and Components should be open source and free and there should be no any

licensing cost involved in any direct or indirect sense.

4. Should be scalable on high load or demands.

5. System should be modular with respect to the business needs.

6. Performance

**2.1.4 interface:-**

**2.1.4.1 User Interfaces:-**

-home page which provides all function.

-user interface should be easy to interact

|  |  |
| --- | --- |
| **2.1.4.2 Hardware Interfaces** |  |
| **Monitor screen:-**The software shall display information to the user via monitor.  **Mouse:-**The software shall interact with the movement of the mouse.  **Keyboard:-**The software shall interact with the keystrokes of the keyboard. |

**2.1.4.3 Software interface**

Software interface is developed in JAVA and SQL language.interface of this SOFTWARE developed according to the latest tags of JAVA and SQL.

**2.2 Feasibility Report :-**

**2.2.1 Innovativeness and Usefulness :-**

Technical feasibility centers on the existing manual system of the testmanagement process and to what extent it can support the system.According to feasibility analysis procedure the technical feasibility of the system is analyzed and the technical requirements such as software facilities,input are identified.

**2.2.2Market Potential and Competitive advantages:-**

Economic analysis is most frequently used for evaluation of the effectiveness of the

system. This part of feasibility study gives the top management the economic

justification for the new system. This system has a great market potential because,

if the organization implements this system, it need not require any additional

hardware resources as well as it will be saving lot of time.

# DEVELOPMENT METHOD:

The following methods and approaches are used to develop this project.

## MySQL Server

A database management, or DBMS, gives the user access to their data and helps them transform the data into information. These systems allow users to create, update and extract information from their database.

A database is a structured collection of data. Data refers to the characteristics of people, things and events. SQL Server stores each data item in its own fields. In SQL Server, the fields relating to a particular person, thing or event are bundled together to form a single complete unit of data, called a record (it can also be referred to as raw or an occurrence). Each record is made up of a number of fields. No two fields in a record can have the same field name.

## SQL Server Tables

SQL Server stores records relating to each other in a table. Different tables are created for the various groups of information. Related tables are grouped together to form a database.

## Primary Key

Every table in SQL Server has a field or a combination of fields that uniquely identifies each record in the table. The Unique identifier is called the Primary Key, or simply the Key. The primary key provides the means to distinguish one record from all other in a table. It allows the user and the database system to identify, locate and refer to one particular record in the database.

* **Foreign Key**

When a field is one table matches the primary key of another field is referred to as a foreign key. A foreign key is a field or a group of fields in one table whose values match those of the primary key of another table.

* **Referential Integrity**

Not only does SQL Server allow you to link multiple tables, it also maintains consistency between them. Ensuring that the data among related tables is correctly matched is referred to as maintaining referential integrity.

* **Relational Database**

Sometimes all the information of interest to a business operation can be stored in one table. SQL Server makes it very easy to link the data in multiple tables. Matching an employee to the department in which they work is one example. This is what makes SQL Server a relational database management system, or RDBMS. It stores data in two or more tables and enables you to define relationships between the table and enables you to define relationships between the tables.

* **Data Abstraction**

A major purpose of a database system is to provide users with an abstract view of the data. This system hides certain details of how the data is stored and maintained. Data abstraction is divided into three levels.

* + **Physical level:** This is the lowest level of abstraction at which one describes how the data are actually stored.
  + **Conceptual Level:**  At this level of database abstraction all the attributed and what data are actually stored is described and entries and relationship among them.
  + **View level:**  This is the highest level of abstraction at which one describes only part of the database.

**Advantages of RDBMS**

1. Redundancy can be avoided
2. Inconsistency can be eliminated
3. Data can be Shared
4. Standards can be enforced
5. Security restrictions ca be applied
6. Integrity can be maintained
7. Conflicting requirements can be balanced
8. Data independence can be achieved.

**Disadvantages of RDBMS**

A significant disadvantage of the RDBMS system is cost. In addition to the cost of purchasing of developing the software, the hardware has to be upgraded to allow for the extensive programs and the workspace required for their execution and storage. While centralization reduces duplication, the lack of duplication requires that the database be adequately backed up so that in case of failure the data can be recovered.

## JAVA 1.6

The most important characteristic of Java is that it was designed from the outset to be machine independent. We can run Java programs unchanged on any machine and operating system combination that supports Java. Java programs are intrinsically more portable than programs written in other languages. An application written in Java will only require a single set of source code statements, regardless of the number of different computer platforms on which

it is run, so it is very useful for internet application.

* Platform independence - Java programs can be run on many platforms without modification. This portability is assured by using a Virtual machine1. When a Java program is compiled byte-code is created rather than a standard executable file. Effectively, this is machine code for a virtual machine, which is then interpreted by the Java interpreter. The byte-code can be run on any platform which has a suitable interpreter.
* Security - since Java has always been designed with distributed applications in mind, security has been incorporated right from the start, and if anything this has been seen to be too restrictive.

**Java and the JDK**

The Java Development Kit contains all the necessary tools for the development of Java applications and applets, including a compiler, interpreter, class libraries, applet viewer and debugger. The current version of the JDK, 1.4.2, is freely available from Sun's web she- for machines running Windows 95, Windows NT, Solaris SPARC and Solaris x86, and a version for the Macintosh is expected later this year (version 1,0 is currently available). Ports to other machines may also be available from other sources. It should be noted that browsers may not yet support applets created using features in the latest JDK.

**Characteristics of Java**

* Simple
* Secure
* Portable
* Object-oriented
* Robust
* Multithreaded
* Architecture-neutral
* Interpreted
* High performance
* Distributed
* Dynamic

**J2EE**

The Java-2 Enterprise Edition (J2EE™) provides a component-based approach to the design, development, assembly, and deployment of enterprise applications. The J2EE platform offers a multitiered distributed application model, reusable components, a unified security model, flexible transaction control, and web services support through integrated data interchange on Extensible Markup Language (XML)-based open standards and protocols.

**Components of J2EE**

* **SERVLET**

Java servlets are small, platform-independent Java programs that can be used to extend the functionality of a Web server in a variety of ways. Servlets are to the server what applets are to the client—small Java programs compiled to bytecode that can be loaded dynamically and that extend the capabilities of the host. It is a server side programming language.

* **JSP**

Java Server Pages is the extension of servlet to simplify the programming and coding of servlet. It includes HTML tags to make coding easy. Ultimately it convert into the servlet at the time of calling.

**CHAPTER 3**

**PROCESS MODEL**

**3.1 Proposed Process Model:-**

**In this project Waterfall Model is followed.**



**Fig 3.1**

**This model contains 6 phases:-**

* **Feasibility and requirement analysis :-**

Requirement gathering activity involves the analysis of the problem and

collection of the relevant information relating to the product. The main aim of

this activity is to determine whether it would be financially and technically

feasible to develop the product.

* **Requirement gathering:-**

The goal of this phase is to understand the exact requirements of the customer

and to document them properly.

* **System Design :-**

The goal of this phase is to transform the requirement specification into a

structure that is suitable for implementation in some programming language.

* **Testing:-**

In this phase all the modules of the software are tested.

* **Deployment of System:-**

In this all the modules are integrated and then deployed.

* **Operation and Maintenance:-**

Release of software inaugurates the operation and life cycle phase of the

operation.

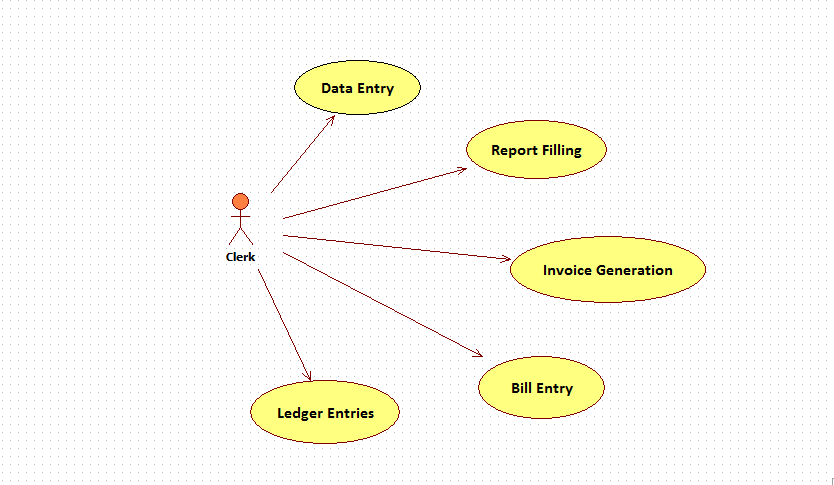
**The phases always occur in this order and do not overlap.**

**CHAPTER 4**

**DESIGN**

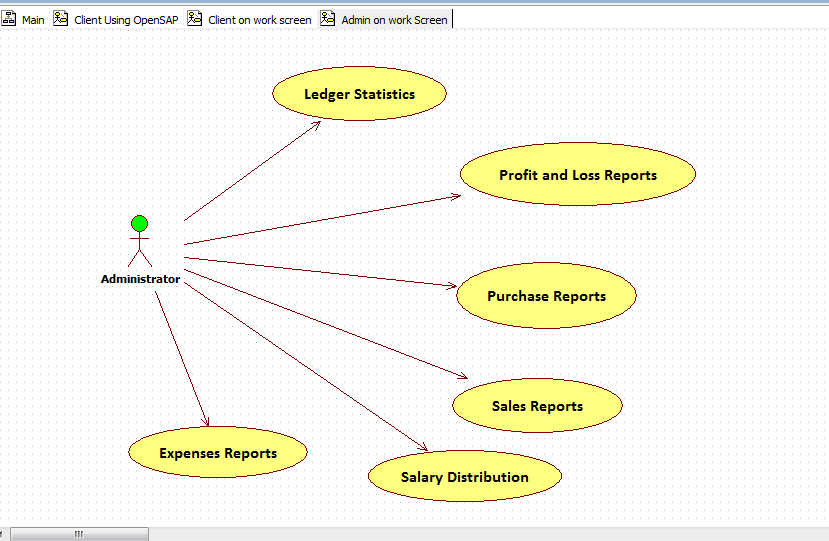
**4.1 Use case diagram**:- A use case diagram can portray the different types of users of a system and the various ways that they interact with the system.

**Fig 4.1**

****

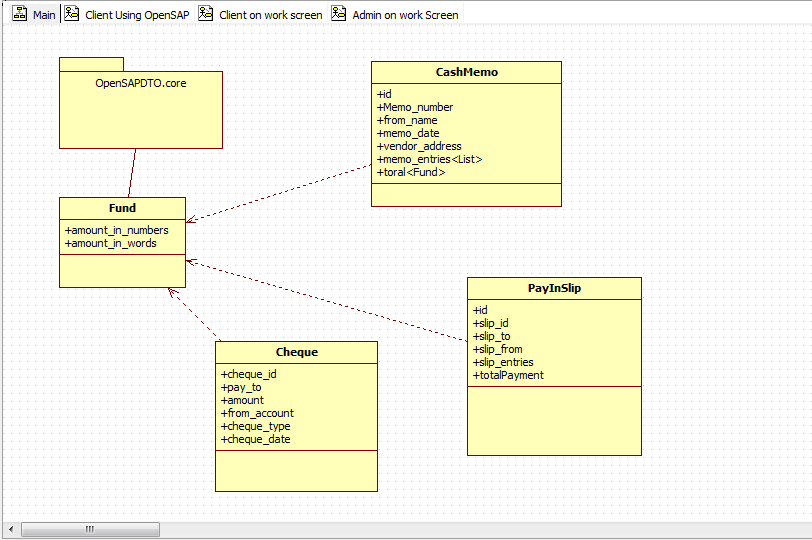
Use case diagram for a cleark level person who interact with different modules.

**Fig 4.2**

****

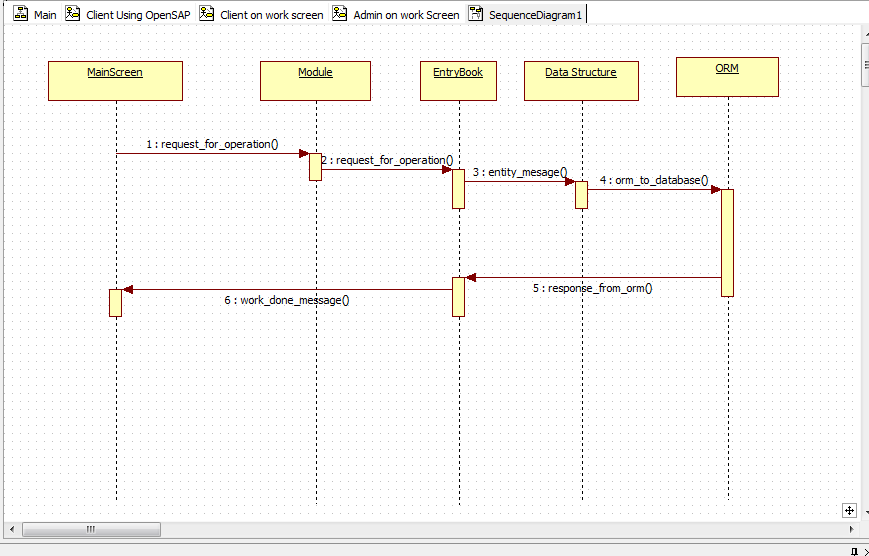
Use case diagram for the Administrator person who used different modules according to his admin panel.

**4.2Class diagram**

****

**4.3 Sequence Diagram:-** A sequence diagram interaction diagram that shows how processes operate with one another and in what order.

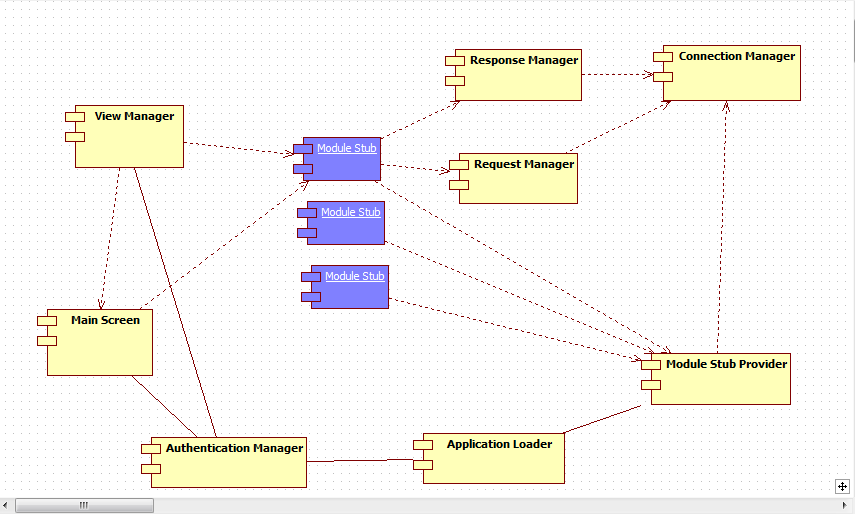
SEQUENCE DIAGRAM OF HOME PAGE:-



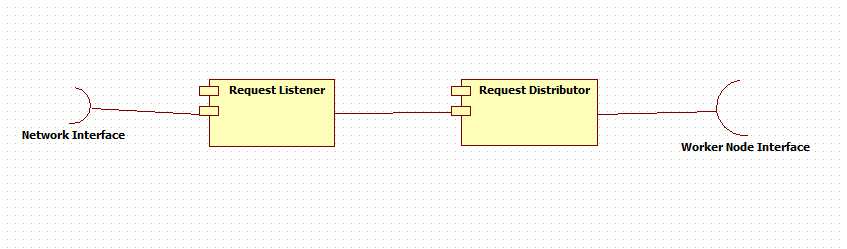
Following diagram shows how module is loaded and its working operationwise in the system.

**4.4 Component Diagram**:- A component diagram depicts how components are wired together to form larger components and or software system. They are used to illustrate the structure of arbitrarily complex systems.

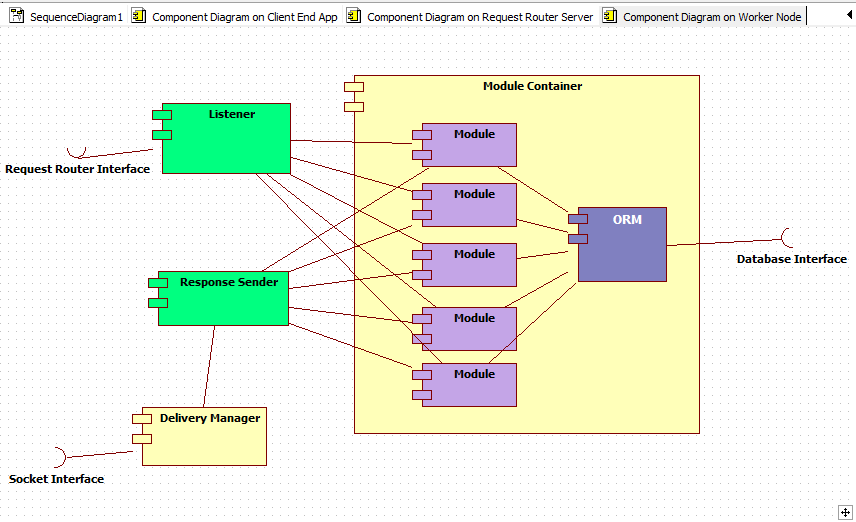
**Fig 4.4.1 Desktop App**



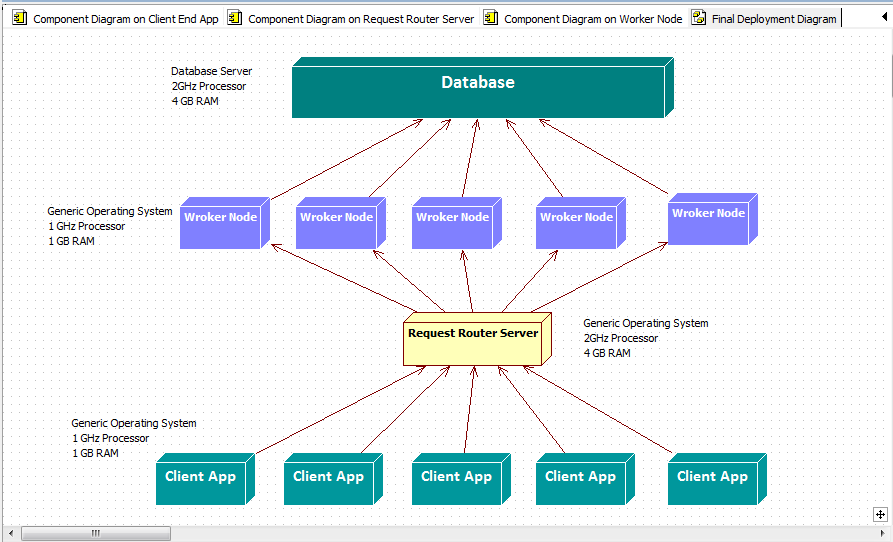
**Fig 4.4.2 Request Routing**

****

**Fig 4.4.3 In Server**

****

**4.5 Deployment Diagram:- A deployment diagram in the**[**Unified Modeling Language**](http://en.wikipedia.org/wiki/Unified_Modeling_Language)**models the physical deployment of**[**artifacts**](http://en.wikipedia.org/wiki/Artifact_(UML))**on**[**nodes**](http://en.wikipedia.org/wiki/Node_(UML))**.**

****

**CHAPTER 5**

**TECHNICAL DETAILS**

**5.1 Software Specification:-**

**Server:**

* Front end Web: HTML,CSS,JavaScript Desktop App : Java,JavaFX
* Back end JSP,Servlet,mysql
* Database My SQL.
* Server MySQL Server,Apache Tomcat 7.0.1
* Connectivity jdbc-hibernate connectivity over jdbc

**Client:**

* Java Virtual Machine Should be installed
* A web browser should be installed to access web-application.
  1. **Hardware Specification:-**

**Client Side**

* 512 MB RAM (minimum).
* 1 GB Hard disk Space(minimum).
* Network Connection

**Server Side**

* 4 GB RAM (minimum).
* 1 GB Hard disk Space(minimum).
* Network Connection

**CHAPTER 7**

**TESTING**

* Software testing is the process of executing a program with intension of finding errors in the code.
* It is a process of evolution of system or its parts by manual or automatic means to verify that it is satisfying specified or requirements or not.
* To purpose of system testing is to check and find out the errors or faults as early as possible so losses due to it can be saved.
* Testing is the fundamental process of software success.
* Testing is not a distinct phase in system development life cycle but should be applicable throughout all phases i.e. design development and maintenance phase.
* Testing is used to show incorrectness and considered to success when an error is detected.

Different levels of testing are used in the test process; each level of testing aims to test different aspects of the system:-

The first level is unit testing. In this testing, individual components are tested to ensure that they operate correctly.

The second level is integration testing. It is a systematic technique for constructing the program structure. In this testing, many tested modules are combined into the subsystems which are then tested. The good here is to see if the modules can be integrated properly.

Third level is integrationtesting. System testing is actually a series of different tests whose primary purpose is to fully exercise computer based system. These tests fall outside scope of software process and are not conducted solely by software engineers.

**7.1 Testing Used:-**

* White Box Testing
* Black box Testing

**7.1.1 White box testing:-** White-box testing tests internal structures or workings of a program, as opposed to the functionality exposed to the end-user. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs.

While white-box testing can be applied at the unit, integration and system levels of the software testing process, it is usually done at the unit level. It can test paths within a unit, paths between units during integration, and between subsystems during a system–level test. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements.

**Techniques used in white-box testing include:**

* API testing (application programming interface):- testing of the application using public and private APIs.
* [Code coverage](http://en.wikipedia.org/wiki/Code_coverage):-creating tests to satisfy some criteria of code coverage (e.g., the test designer can create tests to cause all statements in the program to be executed at least once).
* [Fault injection](http://en.wikipedia.org/wiki/Fault_injection) method:- intentionally introducing faults to gauge the efficacy of testing strategies.
* [Mutation testing](http://en.wikipedia.org/wiki/Mutation_testing) methods.
* [Static testing](http://en.wikipedia.org/wiki/Static_testing) method.

**7.1.2 Black-Box Testing**:- Black-box testing treats the software as a "black box", examining functionality without any knowledge of internal implementation. The testers are only aware of what the software is supposed to do, not how it does it. Black-box testing methods include: [equivalence partitioning](http://en.wikipedia.org/wiki/Equivalence_partitioning), [boundary value analysis](http://en.wikipedia.org/wiki/Boundary_value_analysis), [all-pairs testing](http://en.wikipedia.org/wiki/All-pairs_testing), [state transition tables](http://en.wikipedia.org/wiki/State_transition_table), [decision table](http://en.wikipedia.org/wiki/Decision_table) testing, [fuzz testing](http://en.wikipedia.org/wiki/Fuzz_testing), [model-based testing](http://en.wikipedia.org/wiki/Model-based_testing), [use case](http://en.wikipedia.org/wiki/Use_case) testing, [exploratory testing](http://en.wikipedia.org/wiki/Exploratory_testing) and specification-based testing.

### 7.1.3Alpha testing:- Alpha testing is simulated or actual operational testing by potential users/customers or an independent test team at the developers' site. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing.

**7.1.4 Beta testing:-** Beta testing comes after alpha testing and can be considered a form of external [user acceptance testing](http://en.wikipedia.org/wiki/User_acceptance_testing). Versions of the software, known as [beta versions](http://en.wikipedia.org/wiki/Beta_version), are released to a limited audience outside of the programming team. The software is released to groups of people so that further testing can ensure the product has few faults or [bugs](http://en.wikipedia.org/wiki/Computer_bug). Sometimes, beta versions are made available to the open public to increase the [feedback](http://en.wikipedia.org/wiki/Feedback#In_organizations) field to a maximal number of future users.

**7.2 Test Cases & Results:-**

**Cheque Entity Class**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S. No** | **Operation** | **Correct Parameters Passed** | **Incorrect Parameters passed** | **Result** | **Expected Result** |
| 1 | Add a cheque | **Pay\_to:** A/11231089  **Amount:** Fund(1000,”One Thousand”)  **From\_account:**  A/12121335  **Cheque\_type:**  ChequeType.PAYEE  (others default) | Pay\_to:  <not specified>  Null string passed **“ ”** | OpenSAPInvalidDataException:  **Message**: Target Account not specified | Error should be thrown. |
| 2 | Update a cheque | **Pay\_to:** A/11231089  **Amount:** Fund(1000,”One Thousand”)  **From\_account:**  A/12121335  **Cheque\_type:**  ChequeType.PAYEE  (others default) | **From\_account:**  <not specified>  Null string passed **“ ”** | OpenSAPInvalidDataException:  **Message**: Target Account not specified | Error should be thrown at incorrect object. |
| 3 | Delete a cheque | **Cheque\_id:**  String cheque\_id: valid A/c112121212 | **Cheque\_id**  **<**invalid input>  Null String | OpenSAPInvalidDataException:  **Message**: Target cheque\_id not specified | Error should be thrown . |
| 4 | Fetch Cheque List | **From\_account:**  A/12121335 | **From\_account:**  **<**invalid input>  Null String | OpenSAPInvalidDataException:  **Message**: Target account not specified | Error should be thrown . |

**Result: Pass**

**PaySlip Entity Class**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Operation** | **Correct Parameters Passed** | **InCorrect Parameters Passed** | **Result** | **Expected Result** |
| 1 | Add a PaySlip | **Slip\_id:**  A/C123234  **Slip\_to:**  A/C1234232 | <null string> passed | OpenSAPInvalidDataException  Message: Account not specified | Error should be thrown at incorrect input |
| 2 | Update a PaySlip | **Slip\_id:**  A/C123234  **Slip\_to:**  A/C1234232 | <null string> passed | OpenSAPInvalidDataException  Message: Account not specified | Error should be thrown at incorrect input |
| 3 | Delete a PaySlip | **Slip\_id:**  A/C123234 | <null string> passed | OpenSAPInvalidDataException  Message: Account not specified | Error should be thrown at incorrect input |
| 4 | Fetch PaySlip List | **Slip\_from:**  A/C123234 | <null string> passed | OpenSAPInvalidDataException  Message: Account not specified | Error should be thrown at incorrect input |

**Result: Pass**

**CashMemo Entity Class**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No** | **Operation** | **Correct Parameters Passed** | **InCorrect Parameters Passed** | **Result** | **Inference** |
| 1 | Add a CashMemo | **Memo\_id:**  A/C123234  **memo\_to:**  A/C1234232 | <null string> passed | OpenSAPInvalidDataException  Message: Account not specified | Error should be thrown at incorrect input |
| 2 | Update a CashMemo | **Memo\_id:**  A/C123234  **memo\_to:**  A/C1234232 | <null string> passed | OpenSAPInvalidDataException  Message: Account not specified | Error should be thrown at incorrect input |
| 3 | Delete a CashMemo | **Memo\_id:**  A/C123234  **memo\_to:**  A/C1234232 | <null string> passed | OpenSAPInvalidDataException  Message: Account not specified | Error should be thrown at incorrect input |
| 4 | Fetch CashMemo List | **Memo\_id:**  A/C123234  **memo\_to:**  A/C1234232 | <null string> passed | OpenSAPInvalidDataException  Message: Account not specified | Error should be thrown at incorrect input |

**Result: Pass**

**CHAPTER 8**

**SCREEN LAYOUTS**

**LOAD BALANCING SYSTEM**

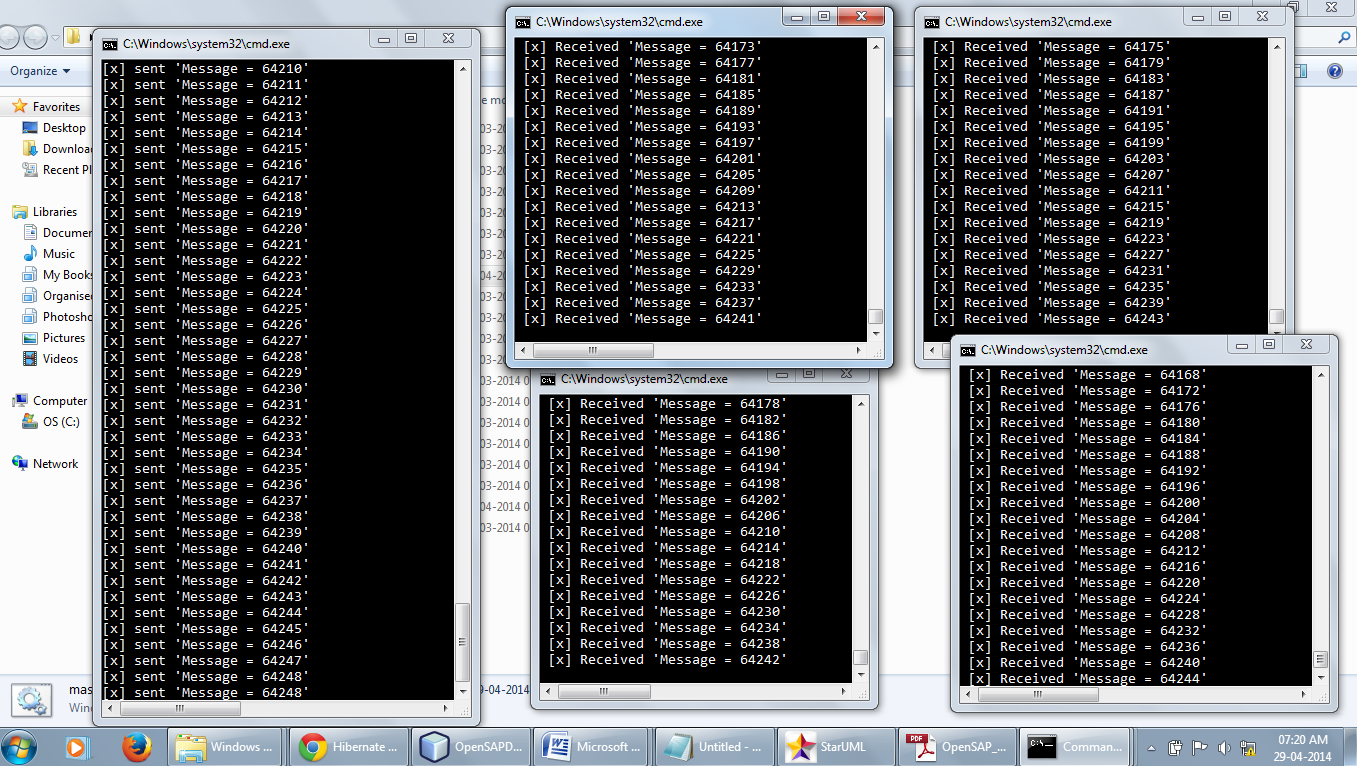
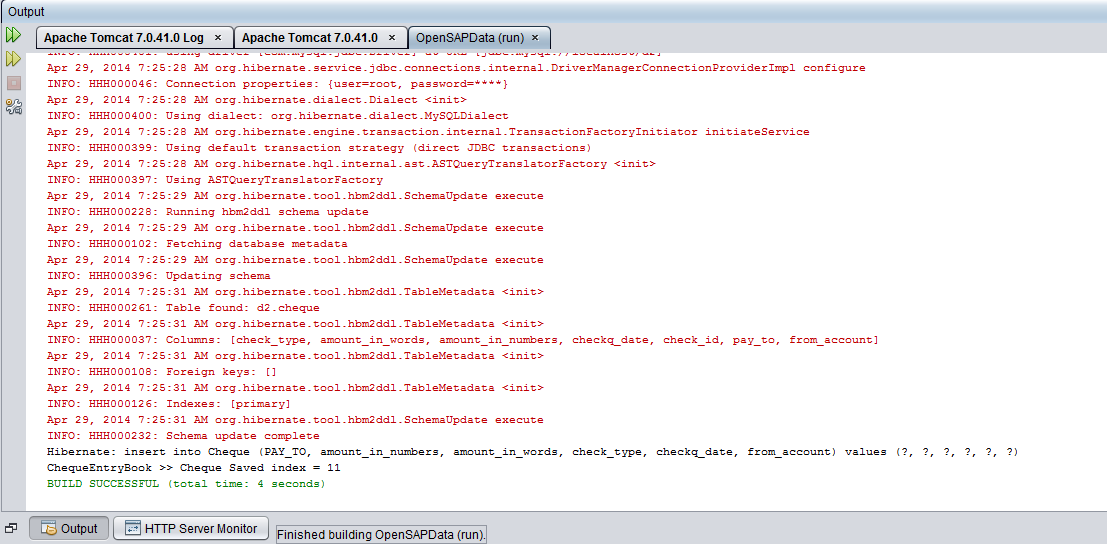
****

Fig: The Load Balacing Reqquest router Server demo where request objects being generated by a master process are evenly distributed to the worker process.

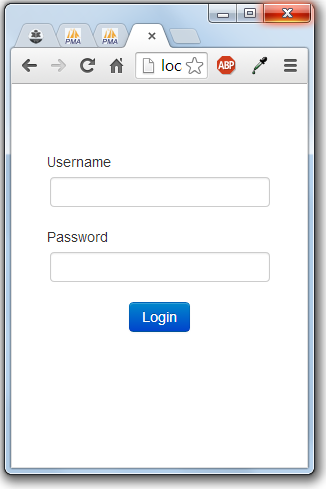
**Working of the DATA Structures:**



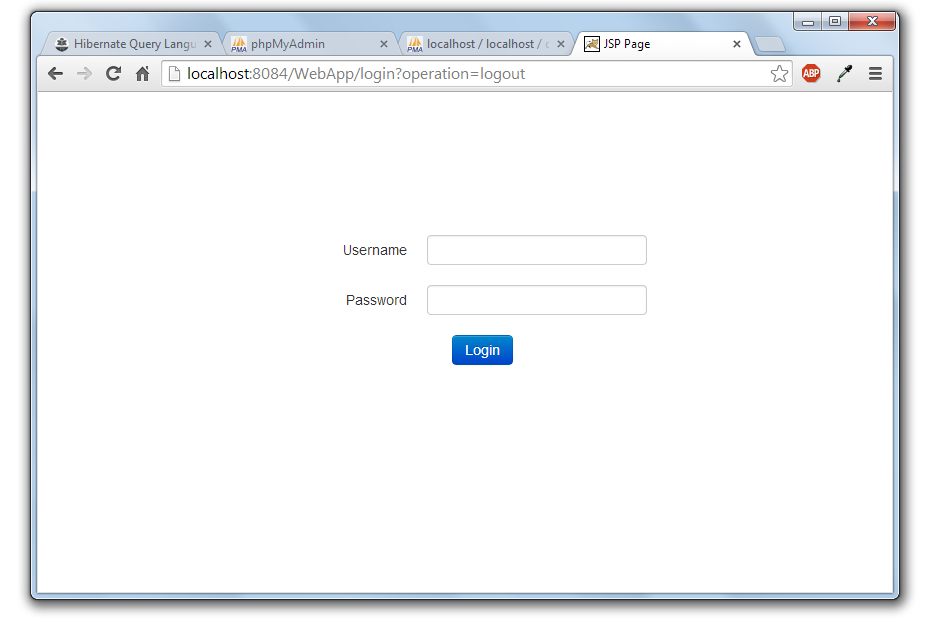
**LOGIN SCREEN:**

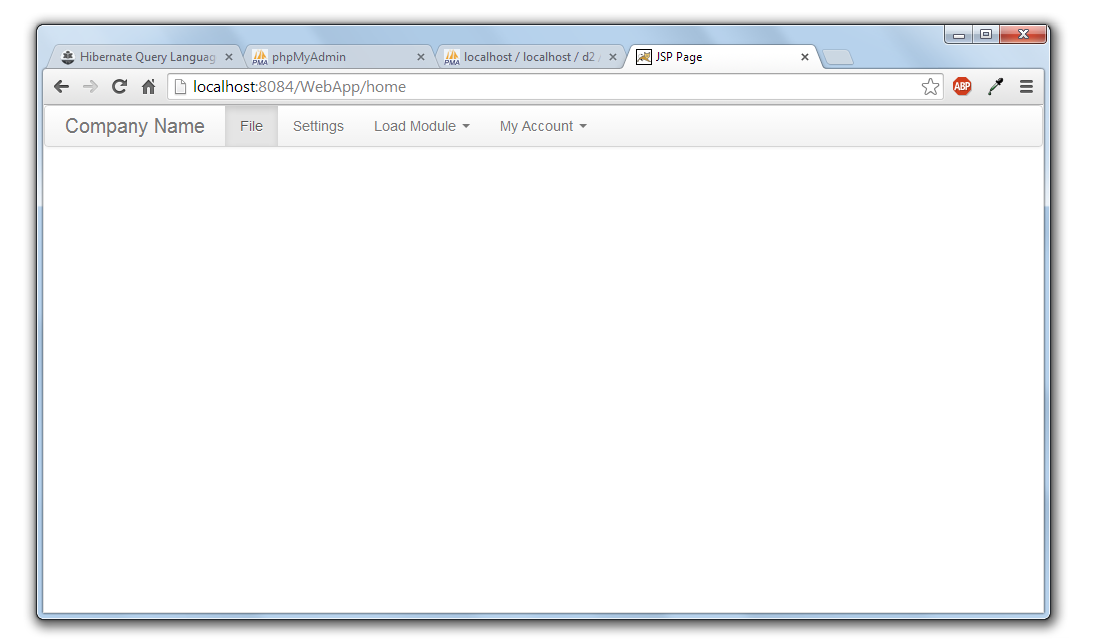
**Screen-1 (Initial)**

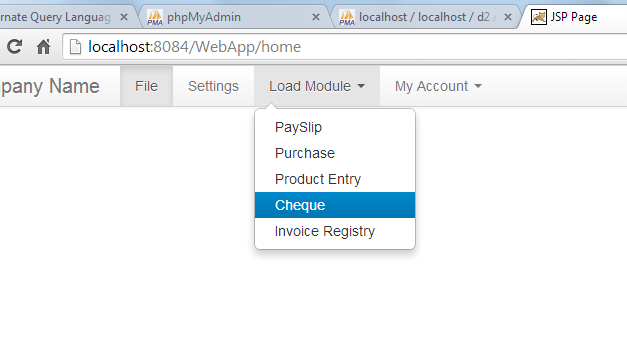
For low screen resolution device web browser



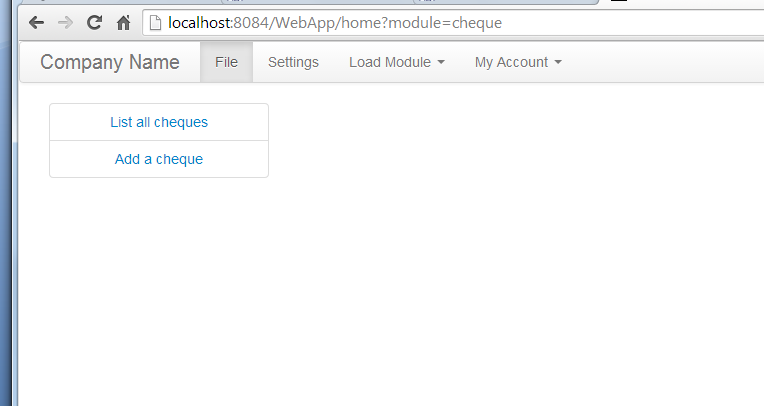
For wide screen browsers:

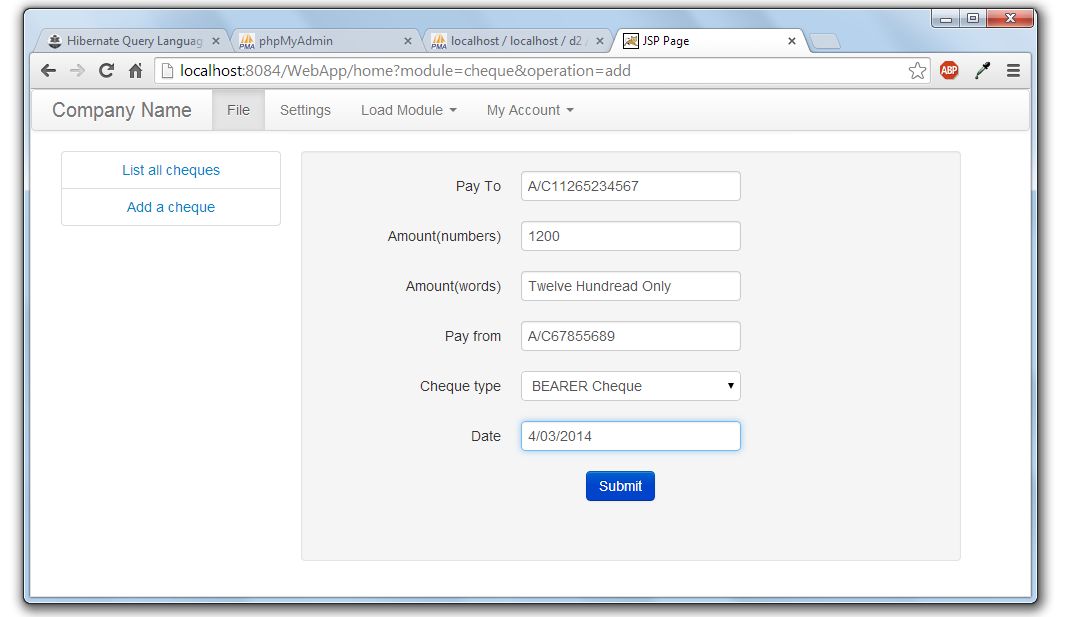


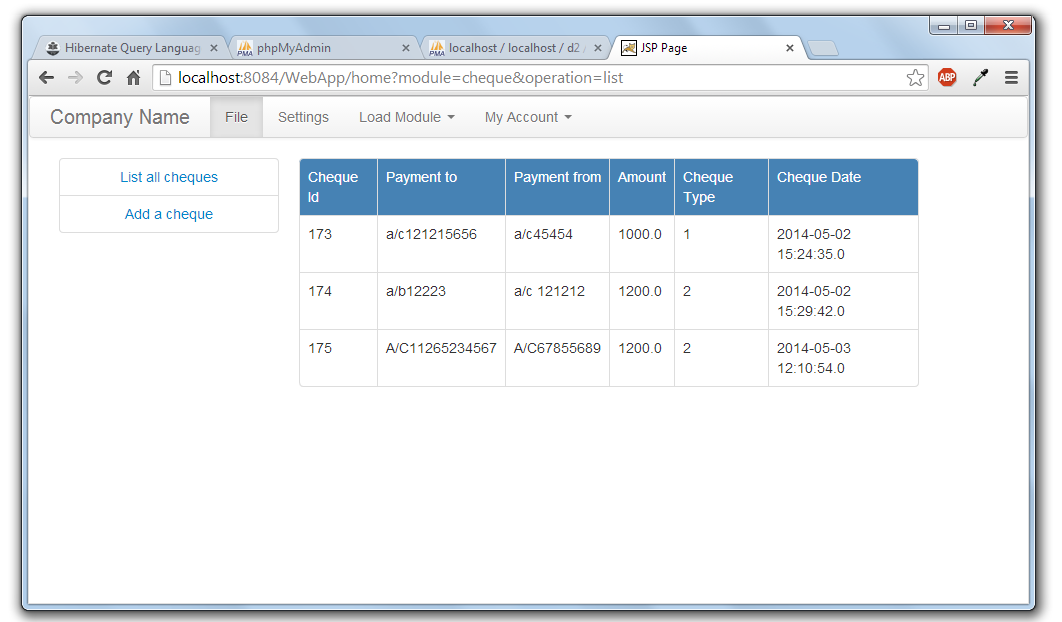
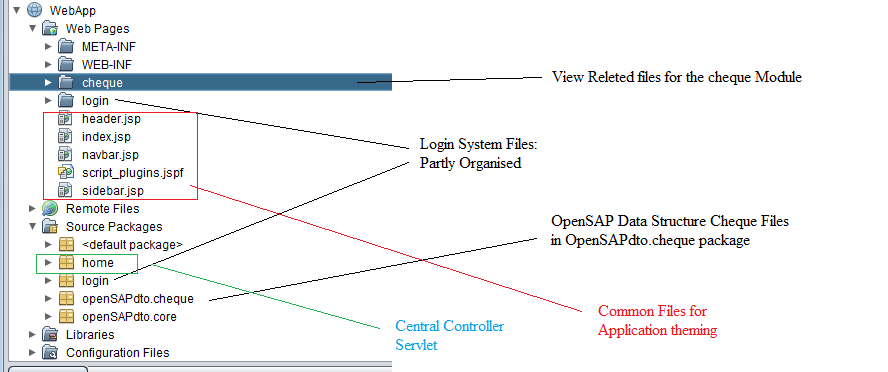
Home Screen After Login   


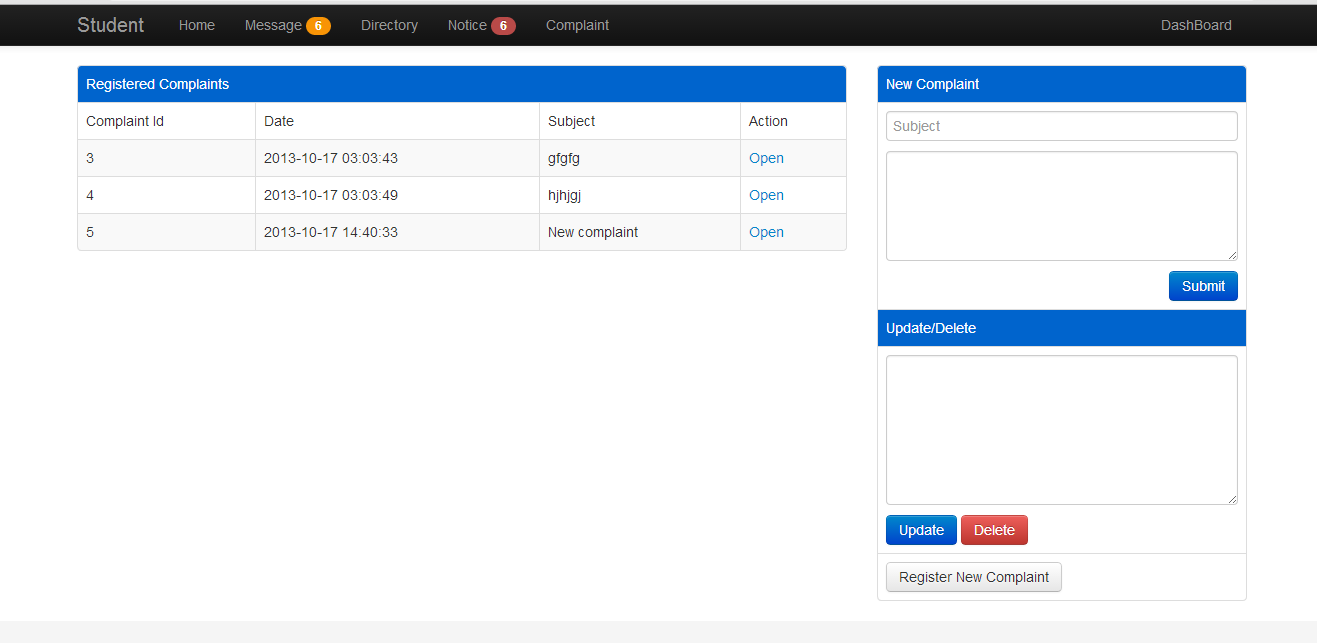
Selecting Cheque Module  


Cheque Module Loaded with list of operations:

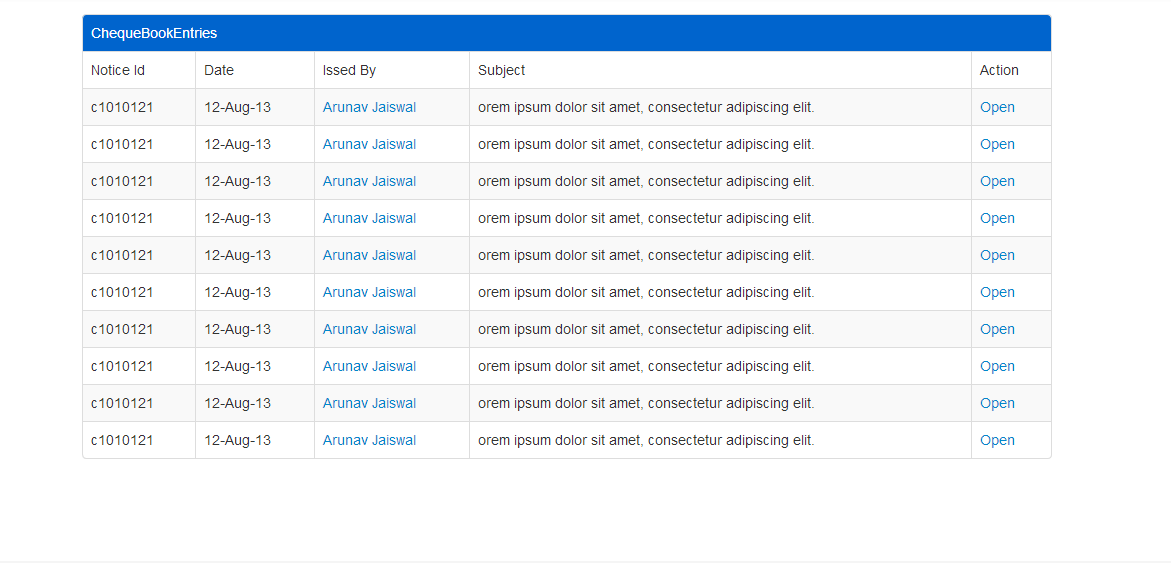


Adding A cheque  


Listing all cheques:  
  
File organisation in Project:  


**A complaint system built with the similar architecture (php based)**

**Fig:** A complaint panel showing the capabilities of complaint module



**Fig:** Table showing chequeBook Entries

**CHAPTER 9**

**FUTURE ENHANCEMENTS**

OpenSAP is a collection of different facilities and helper systems put together for ease of business application development. Motivated from the SAP systems it helps developers to skip working on the infrastructure of the application system and focus on the business goal first.  
The modular architecture of the OpenSAP helps more when it comes on tailoring the application for any specific requirement. Different modules are loosely coupled in the glue of connection components that help abstract the interactivity and most of the time the interaction will be between uniform interaction interfaces. Following are the key areas which will be enhanced in the future and forthcoming releases of OpenSAP:

1. **Module Development**: According to the standard business context different modules can be created which will be used to add up the capability of the system according to the needs.
2. **Enhancements in the Views** : Themes will be developed for the View Manager for view rendering which will help the user to work with more visual interest and will increase the overall productivity of him as well as the organisation.
3. **Web Interface for foreign computers**: Computers which are located away from the business organisation or the personal systems of the Administrators or other people should not be bloated with the OpenSAP desktop client if they want to work outside the working environment. Rather a lightweight web interface will be made which will load as a web application.
4. **OpenSAP SDK and IDE**: Making the modules and manually deploying can sometimes be cumbersome for new developers on the OpenSAP system so in next releases we are planning to prepare our own IDE for the smooth and systematic development of the modules and core enhancement.SDK will provide the API’s for various facilities and IDE will be a way to code and deploy the business code.

**CHAPTER 10**

**CONCLUSION**

OpenSAP starting as the bunch of different application development facilities will soon emerge as a complete application development stack and being Open Source it can prove to be a cheaper alternative for High priced SAP systems which are no doubt useful and successful but still the cost factor associated with them makes it difficult for Low budget organisations, students and researchers to involve in its development and use.

The project is made open-source with the hope that developers from the different parts of the world will take interest and will be able to collaborate in different parts of the OpenSAP without any issue of copyright and ownership.

The Open Source projects are always made keeping in mind the utility and the cost pain associated with the establishment of those utilities.With making this software an open source utility my aim in mind is the development and evolvement of the product without any glitch associated with the licensing -problems.Till now with completion of this report I have made the lengendry web framework which is working quite similar to what the application will be working in near future.Application does have some generic login system that works fine on different levels of rights.It have a module registry that registers different modules and the corresponding operations those modules provide.The developer who will develop the application on this platform will use the ready made Data transfer objects to store the information in the database and if needed he can make his own requirement based DTO by tailoring the existing once.Then He will have to prepare the views corresponding to the different action that follow in the application.The application framework removes the integration headache of developer by providing a tight mechanism that works in flow,the developer have to prepare just the fit-in components for the application workflow.

This type of application framework increases the developer’s productivity to many folds since the developer need not to worry about the integration and code-boilerplate and focus mainly on the business logic of the application.

**References:**(1) Java collection framework studied at tutorialspoint.com  
<http://www.tutorialspoint.com/java/java_list_interface.htm>

(2)HTML,CSS and JavaScript   
<http://www.w3schools.com/>  
(3)JavaBrains Tutorials for Hibernate and JSP:  
<http://javabrains.koushik.org/>

**BIBLIOGRAPHY**

The following books were very helpful during the completion of project:

* **Software Engineering**

By Roger S.Pressman.

* **The Complete Reference Java 7th Edition**

By Herbert Schildt

* **Java How to Program**  
   By Deitel and Deitel
* **SAM Teaches you SAP in 24 Hours 4th Edition**

By George W.Anderson