**CHAPTER – I**

**Planning Phase**

1. **Introduction**

In order to ease the operation in an organization and minimize the human intervention in implementing the policies and rules in an organization, it is inevitable to adopt online services. Online services provide an access to the system on 24\*7 basis. The present project has identified the problems encountered by the employees of an organization while applying for leave and the tedious process of recommendation and approval of leave applied by an employee by the administrators. In order to avoid such pitfalls, the present project aims at developing an Online Leave Application System (OLAS).

* 1. **Problem Statement**

The intended project is stated as “Development of an Online Leave Application System” meant for an organization having a structure leave policy and well defined rules for availing leave by an employee.

* 1. **Development Process**

Software process is a roadmap comprising a series of predictable steps to create a timely and high quality software product. Firstly it is required to select the appropriate development process model for the project.

* + 1. **Life Cycle Model for the project**

This project is based upon the well known and well established Iterative Waterfall Life Cycle Model. The iterative waterfall model provides a systematic*,* sequential approach to software development that begins with feasibility study and progresses through requirements analysis and specification, design, coding and unit testing, integration and system testing, and deployment. It was being chosen because all requirements were known beforehand. The feedback paths allow for correction of the errors committed during a phase, as and when these are detected in a later phase.

It includes the following phases:

*Feasibility Study*

The main aim of feasibility study is to determine whether it would be financially and technically feasible to develop the product. In this stage of the development it is also checked whether the product will be financially feasible or not and also the availability of required technical expertise in the area of development is also checked.

*Requirements Analysis and Specification*

The aim of the requirements analysis and specification phase is to understand the exact requirements of the customer and to document them properly. This phase consists of two distinct activities, namely

* Requirements gathering and analysis, and
* Requirements specification

The goal of the requirements gathering activity is to collect all relevant information from the customer regarding the product to be developed. This is done to clearly understand the customer requirements so that incompleteness and inconsistencies are removed.

*Design*

The goal of the design phase is to transform the requirements specified in the Software Requirement Specifications (SRS) document into a structure that is suitable for implementation in some programming language. In technical terms, during the design phase the software architecture is derived from the SRS document.

*Coding and Unit Testing*

The purpose of the coding and unit testing phase (sometimes called the implementation phase) of software development is to translate the software design into source code. Each component of the design is implemented as a program module. During this phase, each module is unit tested to determine the correct working of all the individual modules.

*Integration and System Testing*

Integration of different modules is undertaken once they have been coded and unit tested. Integration is normally carried out incrementally over a number of steps. During each integration step, the partially integrated system is tested and a set of previously planned modules are added to it. Finally, when all the modules have been successfully integrated and tested, system testing is carried out.

System testing usually consists of three different kinds of testing activities:

*α – testing:* It is the system testing performed by the development team.

*β–testing:* It is the system testing performed by a friendly set of customers.

*Acceptance testing:* It is the system testing performed by the customer himself after the product delivery to determine whether to accept or reject the delivered product.

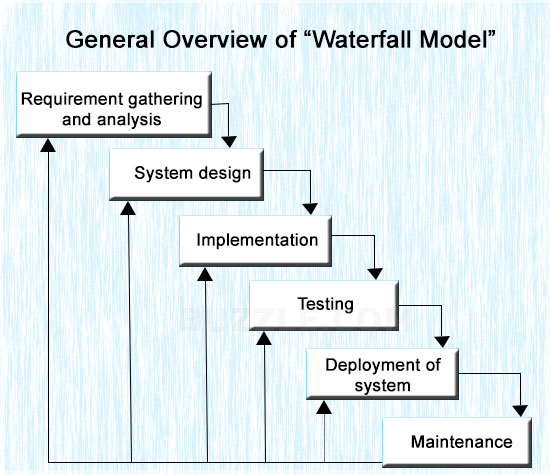
*Maintenance:*

Maintenance involves performing any one or more of the following three kinds of activities:

*Corrective Maintenance:* Correcting errors that were not discovered during the product development phase.

*Perfective Maintenance:* Improving the implementation of the system, and enhancing the functionalities of the system according to the customer’s requirements.

*Adaptive Maintenance:* Porting the software to work in a new environment.



*Figure 1: Iterative Waterfall Model*

* + 1. **Team Structure for the project**

This project involves democratic team structure with THREE members in the team. All the decisions will be made by collective effort of the team members. Since the project will be developed by coordination of all the team members, the responsibility will be equally shared and there will be no burden in any one member for completing the whole project. Moreover, due to mixed experience, varieties in thought process and ideas of the members, the software developed will be more efficient.

Table 1: Team member details

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Member Name** | **Roll no.** |
| 1. | Srishti Priya | 3012212005 |
| 2. | Deepalika Kowar | 3012212019 |
| 3. | Jasreen Kour Bagga | 3012212077 |

**CHAPTER – 2**

**Analysis Phase**

1. **Software Requirement Specification Document**
   1. **Introduction**

This documentation aims at specifying all the requirements for the intended project **“Online Leave Application System”**.

Efforts have been made to gather all the requirements and stipulate it in this document correctly, consistently and removing all ambiguities.

Any assumptions for additional requirements should not be made by any parties involved in implementing, testing and designing while using this product. In case supplementary features are required to be built, then a formal change request must be made and subsequently a new release of this document can be produced.

The following subsections of the Software Requirement Specifications (SRS) provide an overview of the entire SRS.

* + 1. **Purpose**

**Online Leave Application System** is a web based application which provides online services to the employees of an organization to submit leave applications.

The users working in the organization can connect through internet to avail these services. This web based application is more effective and reliable in managing the record of leaves in comparison to the existing system which include a lot of paperwork and human intervention.

* + 1. **Scope**

The **Online Leave Application System** aims athelping the user to address issues from multi-disciplinary angles related to leave management and services. The major benefits of this software are:

* It leads to paperless leave management system
* Leave applications can be tendered, recommended and approved online
* Employees remain tension free of reaching their organization to submit their leave application
* Due to absence of human intervention, the approval process will be fair and uniform
* It is unique application which helps to apply for leaves without any paperwork
* It will include wide variety of modules thus enhancing the capability of the system
* Users can check the status of the leave applied for, know the leave balance, view the notices regarding leave and holidays
* Users become more aware of the leave rules prevailed in their respective organizations
* Administrators get sufficient time to recommend and approve the leave even while away from their offices
* It can be later enhanced into a smart phone application.
  + 1. **Definition, Acronym and Abbreviations used**

**Table 2: Definition, Acronym and Abbreviations**

|  |  |
| --- | --- |
| **GUI** | Graphical User Interface |
| **HTML** | Hyper Text Markup Language |
| **JSP** | Java Server Pages |
| **LB** | Leave Balance |
| **OLA** | Online Leave Application |
| **OS** | Operating System |
| **Servers** | Machines that store all the information and records |
| **SRS** | Software Requirement Specifications |

* + 1. **References**
* Software engineering book by Roger Pressman, Ian Sommerville.
* IEEE Standard 830-1993 IEEE Recommended Practice for Software Requirement Specification.
  + 1. **Overview**

**Review of the Existing System:**

In the current system a leave book is maintained for each employee of the organization and all the requested leaves are entered in it manually. The admin checks for the total leaves and then decides whether to grant a leave or not as per the leave rule prevailed in the organization. This increases the paper work and makes the record maintenance tedious and sole dependency on papers like leave book and leave registers.

**Drawbacks of the existing system:**

* The existing system increases the paper work and makes the record maintenance tedious.
* There is always a risk of damage to the leave books and on damage employee needs to pay a fine to get a new leave book sanctioned.
* Due to human intervention in recommendation and approval process, the admin must be very much alert and aware of the leave rules.
* It is tedious to generate a consolidated report on leaves availed by employees on a monthly basis.
* Trends of availing leave by a particular employee can be easily traced out.

**Proposed System:**

The proposed system of online leave applications automates the existing system. It decreases the paper work and eases the record maintenance by having a database for attendance and leave maintenance. Thus the OLA system reduces the manual effort and paper work as well as saves time in processing the leave applied for.

* 1. **Overall Description**
     1. **Product Perspective**

OLAS is totally self-contained and works efficiently. It provides simple database rather than complex ones and it provides good and easy GUI for new, naive as well as inexperienced users of the computers.

* + 1. **Product Functions**

The product functions will include the following areas:

* Admin and users log in to the application by entering a user name and a password.
* After the successful login the home page appears.

The application is capable enough to store data and perform some editing on them. It will be having user-friendly GUI that will guide the user to easily achieve the same. This application is menu-driven.

Functions can be described as:

* Checking Login
* New Registrations
* Checking available leaves of the user
* Application Submission by the user
* Getting a new application requests from the user
* Granting/Rejecting leaves by the admin
  + 1. **User Characteristics**

Primary users of the system will be employees working in the organization and there will be an admin. Very little technical expertise is needed for reading the outputted data since it is in tabular form.

* Educational level of OLAS- Low
* Experience of OLAS- None
* Technical Expertise- Little
  + 1. **Constraints**

**Software Constraints:** The system will run under windows 7 or higher platforms of Operating Systems.

**Design and Implementation Constraints:** High performance, User friendly, Security based system, Validation of users.

* + 1. **Assumptions and Dependencies**
* Admin and users will have valid username and password for accessing the software.
* Software is dependent on access to internet or intranet.
* All information entered by the user is correct. If any wrong information is found then system will notify an alert.
  1. **Specific Requirements**
     1. **External Interface Requirements**
        1. **User Interface**

The user interface screens are described in Table 3.

**Table 3: OLA User Interfaces Screens**

|  |  |
| --- | --- |
| **Screen Name** | **Description** |
| Login | Log into the system. |
| Registration | New users register their information. |
| User’s Home Page | Display attendance of user, number of leaves, leave balance. Add or update user records. |
| Apply for leave | Displays leave history and apply for leave. |
| Admin’s Home Page | Displays new leave requests from users. |
| Approve or reject leave application | Display leave availability and application form. Add or update records. |

* + - 1. **Hardware Interface**

The system shall run on:

|  |  |  |
| --- | --- | --- |
| * Processor | : | Intel Pentium and Celeron class processors |
| * Processor Speed | : | 1.2 GHz or above |
| * RAM | : | 512 MB |
| * HDD | : | 40 GB |
| * Monitor | : | 15” SVGA |
| * Printer | : | Laser Printer |
| * Mouse | : | Normal |
| * Keyboard | : | Normal |

* + - 1. **Software Interface**
* Net Beans IDE 7.2.1
* Glassfish Server 4.0
* Java 1.6
* HTML 5
* CSS 3
* Microsoft Office Access 2007 Database
* OS Platform such as Windows XP, Windows 7, Windows 8
  + - 1. **Communication Interface**
* This software package should be securely accessible through intranet communication channel (wired or wireless).
* This system supports Google Chrome and Mozilla Firefox web browsers.
  + 1. **Functional Requirements**

The functional requirements of the system should include:

* Maintenance of employee records.
* Provides the information about the leave approval and leave availability.
* Keep employees’ leave record.
* Displays leave history.
* Records leave rules.

The use case report generated is as follows:

*Use Case:* **New Registration** - Any new user using the web based application first time will have to register their details in order to get an access to the application.

*Primary Actor:* User

*Pre-Condition:* Nil

*Basic Flow:*

1. Start the application. User prompted for details.
2. User gives the details.
3. System does validation.
4. Login screen is displayed.

*Alternate Flow:*

1. Registration fails.
2. Prompt the user that the type mismatches with the required details.

*Use Case:* **Login** - Registered users need to log into the application to avail the services.

*Primary Actor:* User/Admin

*Pre-Condition:* The user must be registered.

*Basic Flow:*

1. Start the application. User/Admin prompted for login and password.
2. User/Admin gives the login and password.
3. System does authentication.
4. Main screen is displayed.

*Alternate Flow:*

1. Authorization fails.
2. Prompt the user that he/she typed the wrong username or password.

*Use Case:* **Check Available Leave** - The user is provided with the facility to check his/her leaves.

*Primary Actor:* User

*Pre-Condition:* The user is logged in.

*Basic Flow:*

1. User checks his/her available leaves and their types.
2. User returns back to home page.

*Alternate Flow:*

1. Alert message is displayed if no leaves are available.
2. User returns back to home page.

*Use Case:* **Application Submission** -After checking for available leaves, the user may apply for required leaves accordingly.

*Primary Actor:* User

Pre-Condition: The user has pending leaves.

*Basic Flow:*

1. User fills an application form giving the details for his/her leave request.

*Alternate Flow:*

1. While filling the application form if type mismatches then an alert message is displayed.

*Use Case:* **Get New Application** - The admin gets the new applications requested by the registered users.

*Primary Actor:* Admin

*Pre-Condition:* The admin is logged in and new requests of leaves are received.

*Basic Flow:*

1. The admin gets the new application requests from the users.

*Use Case:* **Grant/Reject** - The admin grants/rejects leave applications as per the organization’s convenience and the prevailing leave policy.

*Primary Actor:* Admin

*Pre-Condition:* The admin is logged in and he/she has received application requests.

*Basic Flow:*

1. Admin checks the details of the employee and decides whether to grant him/her leave or not.
2. This action is redirected into the user’s home page and he/she can view whether his/her leave is sanctioned or not.



Figure 2: Use Case Diagram

* + 1. **Performance Requirements**
* The load time for user interface screen should take no longer than 2 seconds.
* The log-in information shall be verified within 5 seconds.
* Queries shall return result within 5 seconds.
* The system shall consume very little of primary memory.
  + 1. **Design Constraints**
* Design a system that will satisfy functional and non-functional requirements of application program.
* Choose most appropriate structure (Client-server, layered).
* Design may be based on particular structure or style.
  + 1. **Attributes**

**Security**

* Layered structure should be used.
* Most critical assets protected in the innermost layers.
* High level of security validation applied to these layers.
* Provides related protection systems (safely shut down system if failure occurs).

**Maintainability**

* Use fine-grain, self-contained components.
* OLA is being developed in java. Java is an object oriented programming language and is be easy to maintain.

**Portability**

* OLA shall run in any Microsoft Windows environment that contains Java Runtime and Microsoft Access Database.

**Availability**

* System shall be available 24\*7.
* Include redundant components.

**Reliability**

* The system is reliable because the concept of data hiding is implemented. The data accessed by the admin is hidden from other users. Data of individual users is confined to his page.
  + 1. **Other Requirements**
* The system must be completed within the time-frame allotted for development.
* Appropriate funding must be acquired to make required upgrades to the existing systems, buy additional hardware and software and acquire skilled personnel to develop project.
  1. **Data Flow Diagram**

OLA

Users

Admin

Figure 3: Level 0 DFD

Data Store

Check Available Leave

New Registration

Check Login

Application Submission

Gets New

Application Request

Grants/Rejects

Leave

Admin

receives application request

grants or rejects leave

User

registered

member

unregistered

member

apply for leave

check for leaves

Figure 4: Level 1 DFD

**CHAPTER – 3**

**Design phase**

* 1. **Design Document**
     1. **Modules Required**

Table 4: Modules required

|  |  |
| --- | --- |
| **Module Name** | **Description** |
| Login | Log into the system. |
| Registration | New users register their information. |
| User’s Home Page | Display attendance of user, number of leaves, leave balance. Add or update user records. |
| Apply for leave | Displays leave history and apply for leave. |
| Admin’s Home Page | Displays new leave requests from users. |
| Approve or reject leave application | Display leave availability and application form. Add or update records. |

* + 1. **Database Required**
       1. **Registration Database**

**Table 5: Registration Table**

|  |  |  |
| --- | --- | --- |
| **Name** | **Data Type** | **Description** |
| Username | Text | Name of the new user |
| Password | Text | Password of the new user |
| MobileNumber | Number | Mobile Number of the new user |
| Category | Text | Either Admin/ User |

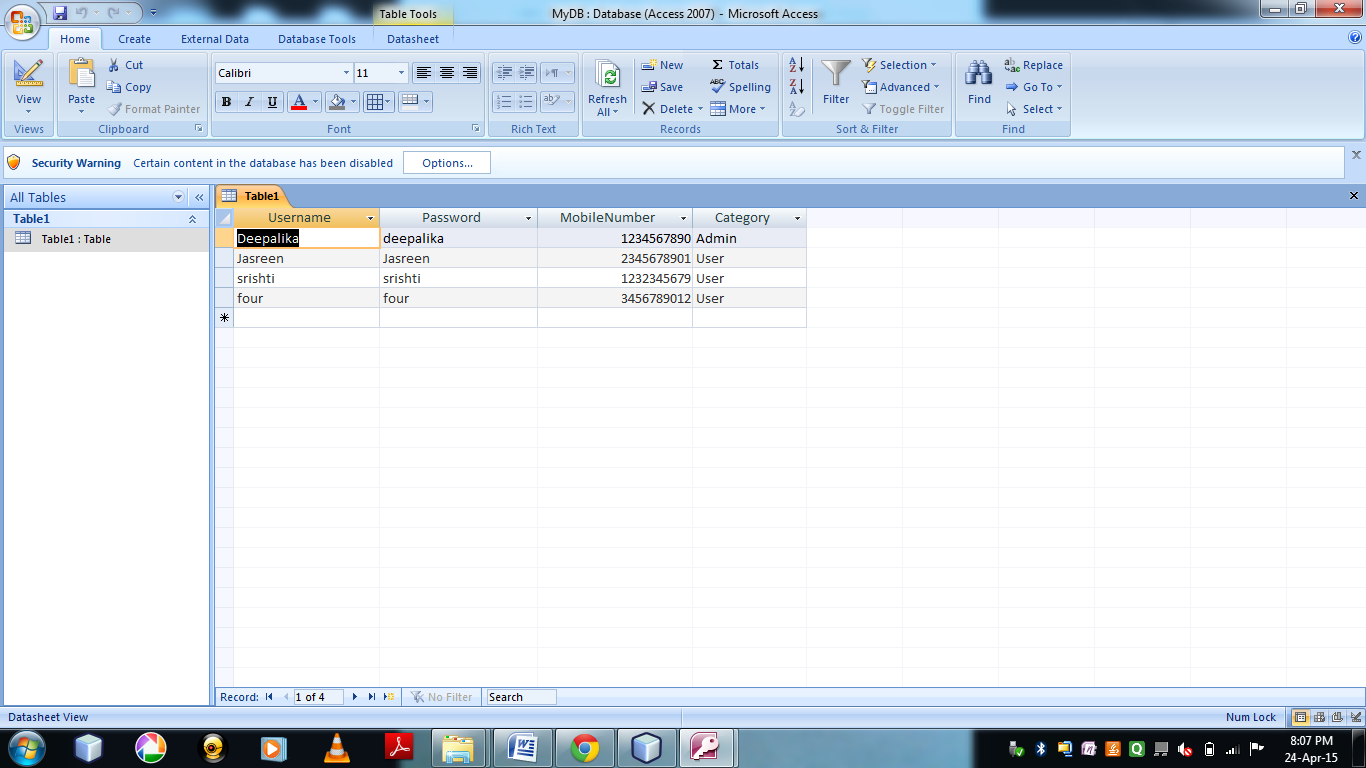


Figure 5: Screenshot of Registration table

* + - 1. **Leave Application Database**

Table 6: Leave Application table

|  |  |  |
| --- | --- | --- |
| **Name** | **Data Type** | **Description** |
| Name | Text | Name of the user |
| LeaveBegins | Date/Time | Beginning Date of leave |
| LeaveEnds | Date/Time | Ending Date of leave |
| TypeOfLeave | Text | Type/ Category of leave |
| ReasonOfLeave | Text | Reason for the leave |
| Message | Text | Message from Admin |

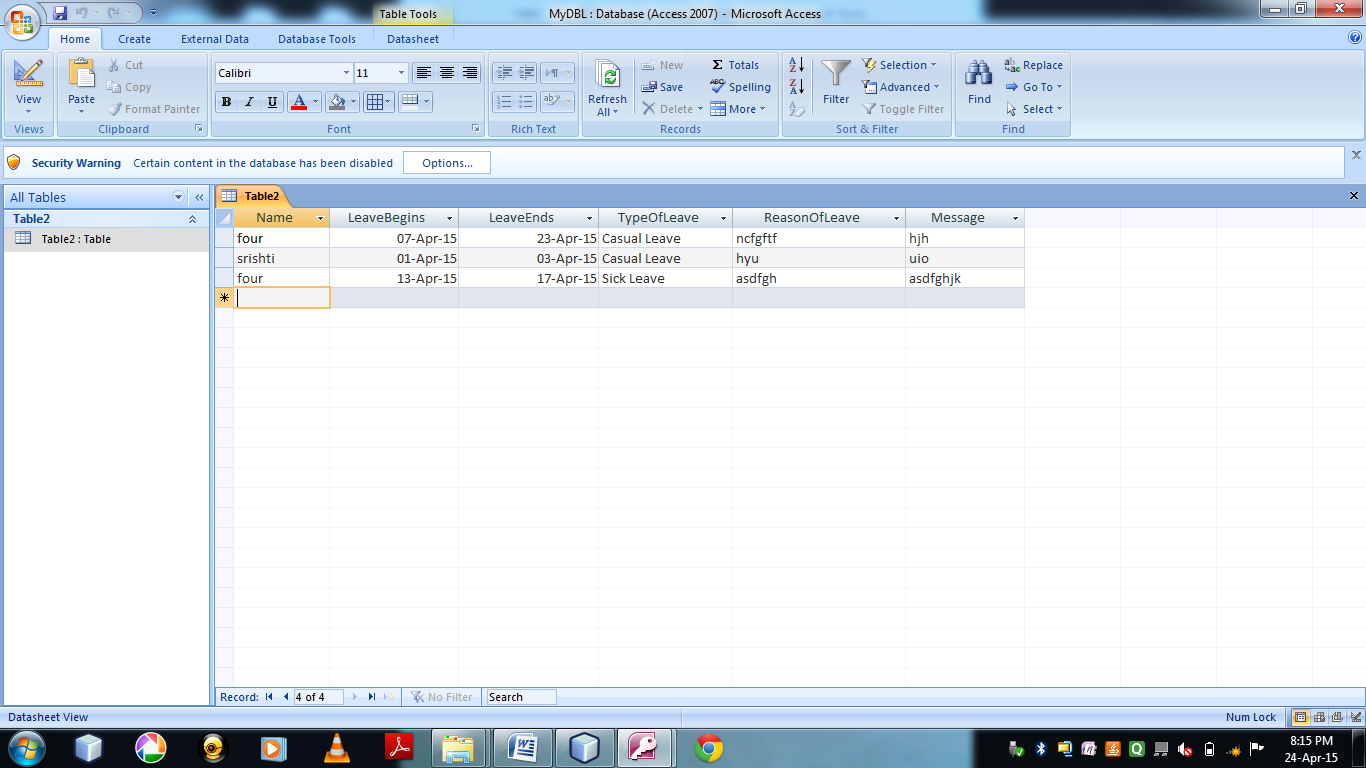
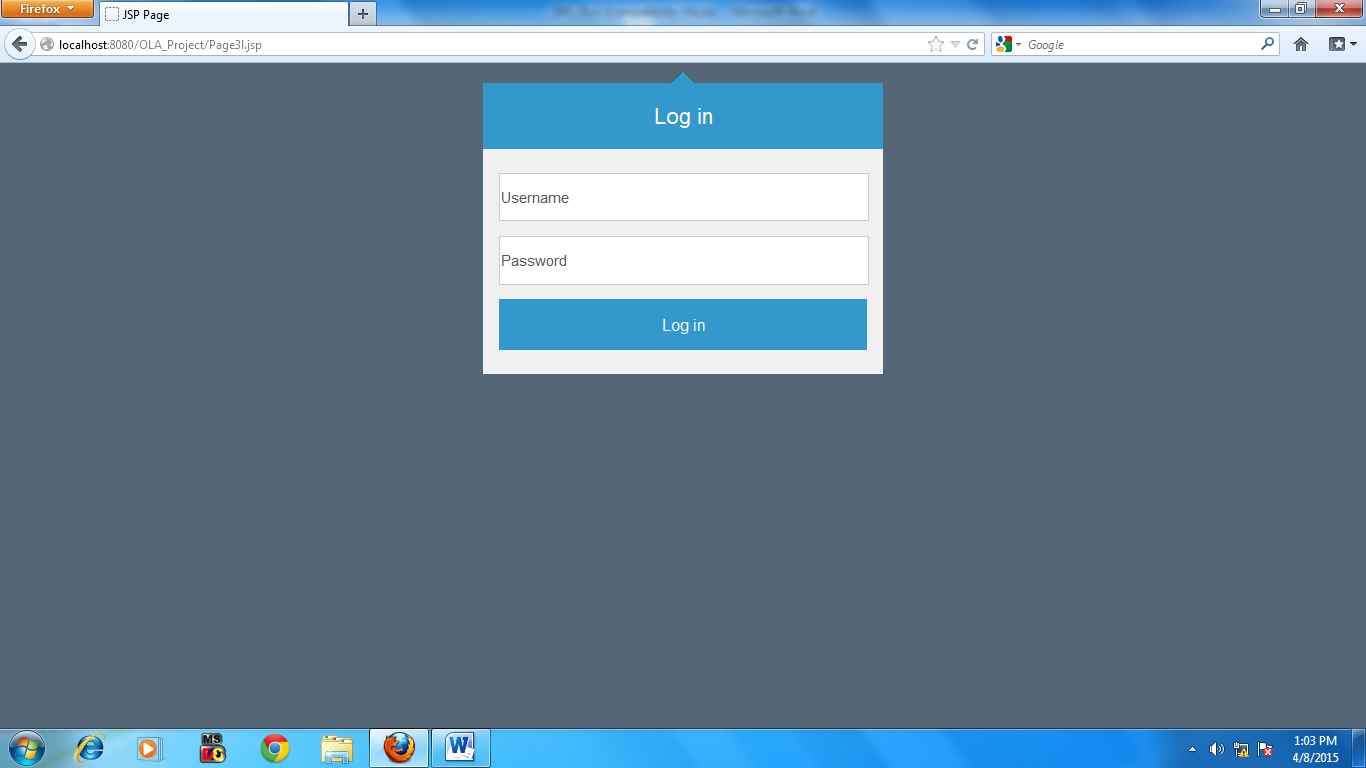
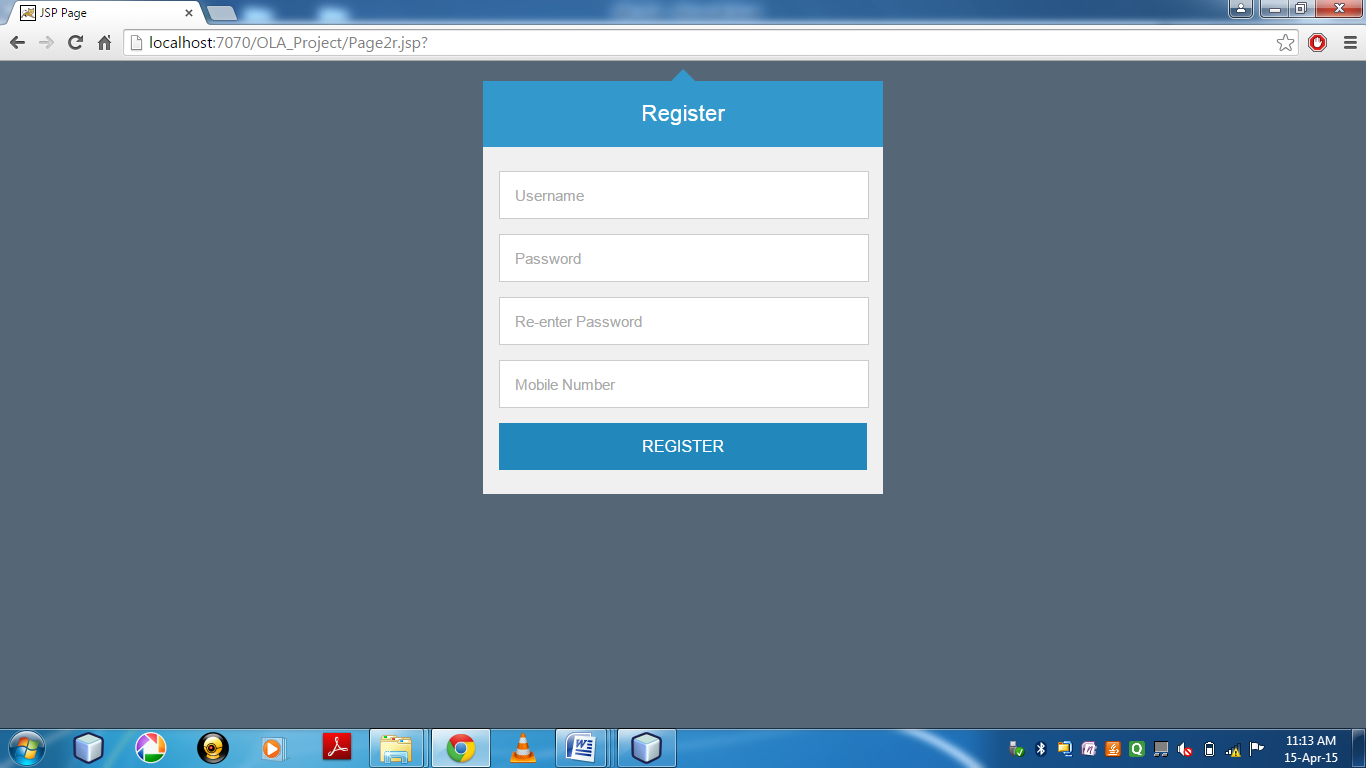
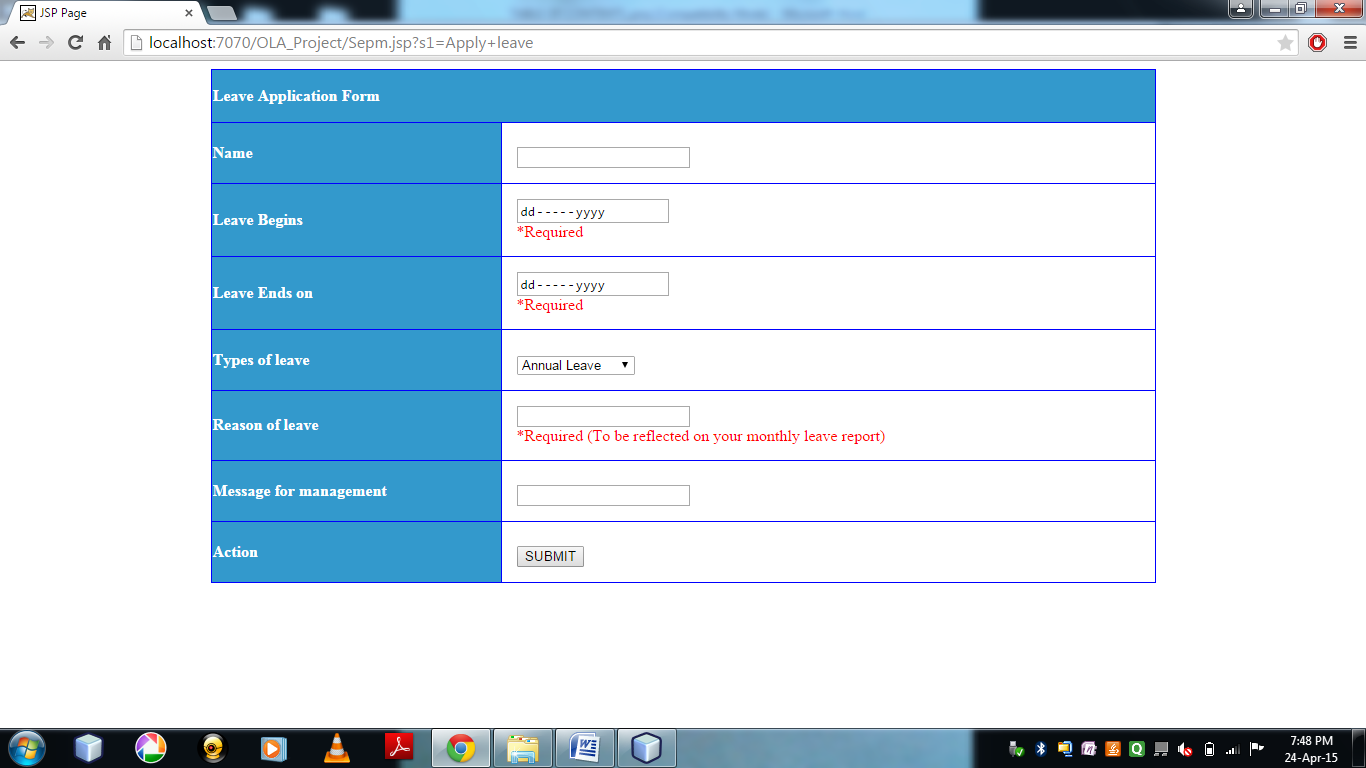
****

Figure 6: Screenshot of Leave Application table

* + 1. **User Interfaces**

Figure 7: Screenshot of Login Page

**Figure 8: Screenshot of Registration Page**

****

**Figure 9: Screenshot of Leave Application Page**

* 1. **Test Plan Document**
     1. **Black Box Testing**

It exercises the input output domain of the program to uncover the error of skill program, function, behavior and performance. It is applied during the later stages of testing. Black Box testing is concerned with the following errors related to:

* Incorrect or missing functions
* Interface errors
* Errors in data structures or external database access
* Behavior or performance errors
* Initialization and termination errors.

Advantages of using this method of testing-

* Tester needs no knowledge of implementation including specific programming languages.
* Tester and programmer are independent of each other.
* Test are done for user’s point of view

In this project of OLA we did not encounter any errors after running Black box testing.

* + 1. **White Box Testing**

This method is performed early in the testing process. Using White box testing method software engineer can derive test cases that:

* Guarantee that all independent paths within a module have been exercised at least once
* Exercise all logical decisions on their true and false sides
* Execute all loops at their boundaries and within their operational bounds
* Exercise internal data structures to ensure their validity

**CHAPTER – 4**

**Conclusion & future scope**

* 1. **Conclusion**

The proposed project on “Online Leave Application System” has been developed and tested. Data and requirement have been collected on a factual basis from an established organization. According to the information and data collected from the source, the requirements of the software have been drawn which invited to develop functional modules. Various modules of the project in use are: Log In, Registration, User’s Home Page, Apply for Leave, Admin’s Home Page, Approve or reject leave application. The screenshots of various modules have been presented. Proper validation of the inputs have been checked and found to be in correct agreement of the requirement of the project. Both White box testing and Black box testing have been carried out with factual data. Bugs detected have been removed and the proposed project is working satisfactorily. Hardware and software specification required for the project have been clearly spelt out. The modules are ready for implementation in any organization. However, before implementation the leave rules and policies are required to be modified according to the need of the concerned organization.

* 1. **Future Scopes**

The **Online Leave Application System** aims athelping the user to address issues from multi-disciplinary angles related to leave management and services. Though sincere efforts have been made to complete the assignment yet there are lot of Scopes to improve upon the base module of the OLA developed through the current project assignment. A few of the scopes are jotted down as below:

* To enhance the reach of the employees and reduction of dependency on local network and hardware available, the present module can be made as an application over smart phone.
* Varieties of tailor made records can be generated as per requirement.
* Notices and circulars can be embedded in the displays.
* Leave rules and Leave policies can be made available to employees on demand.
* Personal messages can be sent to individual.
* SMS services can be integrated to inform the employee regarding the status of the leave applied and leave balance on demand.

**references**

**References**

* Software Engineering – A Practitioner’s Approach, 6th Ed., Roger Pressman, Mc. Graw Hill International Edition,
* Software Engineering Fundamentals, Ali Behforooz & Fredrick J. Hudson, Oxford Publishers.
* Fundamentals of Software Engineering, 3rd Ed., Rajib Mall, PHI
* IEEE Standard 830-1993 IEEE Recommended Practice for Software Requirement Specification.