LEAVE RECOMMENDATION AND APPROVAL SYSTEM

BY

JEHANZAIB KHAN

MUHAMMAD UZAIR

JUNAID NAZIR

Submitted to the Faculty of Electrical Engineering Department
National University of Sciences and Technology, Islamabad
in partial fulfillment for the requirements of a B.E Degree in
Computer Software Engineering
May 2017

CERTIFICATE OF APPROVAL

This is to officially state that the thesis work contained in this report

"Leave Recommendation and Approval System"

Is carried out by:

Jehanzaib Khan, Muhammad Uzair, Junaid Nazir

under my supervision and that in my judgment, it is fully ample, in scope and excellence, for the degree of Bachelor of Computer Software Engineering from National University of Sciences and Technology (NUST).

Approved By:

Signature: _____

Supervisor: ***Asst Prof. Bilal Rauf***

MCS, Rawalpindi

ABSTRACT

Leave Recommendation and Approval System (LRAS) is proposed as a platform for enabling the students and faculty of Military College of Signals to efficiently apply and process their leave applications.

Every day many faculty members and students have to apply for leave for different reasons. Its hectic job for the authorities and the applicants as well to follow a manual system that is currently in practice in MCS. The applicant first has to visit the administration to write their name in the registers or get an application printed, which are then carried by the staff members to each authority that is included in the hierarchy for recommendation and approval. The authority maybe absent or sometimes the staff member may take longer than usual to get the applications approved. The applicant may have to wait for days, not knowing about the status of his/her application. The system is inconvenient and inefficient.

The LRAS will automate this entire system making it more efficient and convenient for applicant, staff members and authorities. The system will be transparent as the user will be able to keep a track of the application status. The authorities will be notified on their account about any application they have to recommend or sign. If the authority is not present due to any reason, the rights will be shifted to the higher authority. The authorities will not have to go through papers to keep a track of all the faculty or student members on leave. An overall leave record will be maintained for each user showing their leave history. A text message will be sent to the authority if the leave is kept unattended for 48 hours. The system will reduce the workload of leave approval in MCS to none.

DECLARATION OF ORIGINALITY

We hereby declare that the work contained in this report and the intellectual content of this report are the product of our work. This thesis report has not been formerly published in any structure nor does it include any verbatim of the published resources which could be treated as violation of the international copyright decree. We also affirm that we do recognize the terms ‘plagiarism’ and ‘copyright’ and that in case of any copyright infringement or plagiarism established in this thesis, we will be held fully accountable of the consequences of any such violation.

DEDICATED WITH LOVE AND RESPECT TO MY PARENTS, TEACHERS AND FRIENDS

ACKNOWLEDGEMENTS

All glory goes to Almighty ALLAH who led us to this extent. May all glory, honor and Adoration be unto thy Name.

Our special thanks go to our supervisor **Asst Prof Bilal Rauf** for his guide line, in completing this project successfully. We wish to add our acknowledgment to the people from MCS, where it all started, especially Dr. Ahmed Muqeem Sheri, Madam Ayesha Naseer, Dr. Hammad Afzal, Dr. Naima Altaf and Dr. Tauseef Ahmed Rana.

A deep gratitude towards **Dr. Adnan Ahmed Khan** (Head of Computer Software Department) for his guidance and facilitation for the Project.

At last we are most obliged to our parents, their support contributed immensely to the success of this project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

1 Introduction

Leave Recommendation and Approval System (LRAS) is a system that will automate the leave recommendation and approval process, which in MCS currently, is carried out manually. The system will be designed for students and faculty. Each user will have his/her account where he/she can apply for leave. The system will offer the nature of leave and based on that data. Hierarchy for the approval of that application will be generated by the system based on the policies and the SOPs as described by MCS. The respective authority will get a notification about the leave application.

The system currently in practice is manual, which is extremely time consuming and hectic. It may take days to get an application approved in the current system, which is very inefficient. Other problems include the absence of the concerned authority. Moreover, it's difficult to keep a record of all the absent personnel and we have to go through stacks of papers in order to find out what we require. LRAS will completely automate the entire process, from signing the application to keeping a record of it.

1.1 Intended Audience and Reading Suggestions

The thesis report of Mobile App for Leave Recommendation and Approval System is meant for all the stake holders.

1. **Project Supervisor:** It will help to supervise the project and guide the team in a better way.
2. **Development Team:** It will help the developer to develop the product and to trace back the functional requirements.
3. **Testing Team:** It will help the testers to understand the constraints.
4. **UG Project Evaluation Team:** Evaluation committee which will evaluate the progress of UG Projects.
5. **Students:** Any student who require any reference or help can read this report.

1.2 Motivation

The leave process in MCS follows a strict hierarchy and is very time consuming. The applicant has to visit each office included in the hierarchy which makes it very hectic and dissatisfactory. This complex leave process system makes the system inefficient and requires an extra effort in maintenance of all the leave records. The authorities also have to pay extra mind to the matter in these cases.

The motivation for our project was to automate the system entirely and bring it on online in order to save time, energy and resources. To make process much more efficient, reliable and transparent, was our goal.

1.3 Project Scope

Each user on the Leave Recommendation and Approval System will have his/her account i.e. faculty or student. The system will offer the nature of leave to the applicant i.e. C/LVE, P/LVE, EX-Pak LVE, Study LVE, Sick LVE. The applicant will choose the type of leave from the given options. The template for the chosen application will appear and the user will fill it out. After the submission of the application the proper hierarchy for getting an application signed, will be generated. The first recommendatory authority who has to recommend the application will get the notification for recommendation and so the hierarchy will be followed. The user will be able to keep a track of the application status. The authorities in the chain of command will be able to leave comments on the leave while exercising their rights.

1.4 Project Vision

For	the students and faculty who wish to apply for leave and in much quicker and efficient way.
What	The Leave Recommendation and Approval System (LRAS) provides an interface to the user with the system generated templates, and while allowing the respective authority to get to know about it and respond to each application
The	Leave Recommendation and Approval System
Is	The project that is mainly categorized as a system for managing leave applications, and is planned to assist both the applicant and the leave granting authority in swift resolution of application approvals
Those	Applicants do not have to apply for leave manually
Unlike	In the existing practice of physical visit to authorities' Office or get the applications printed
Our Product	By using LRAS, user can not only save time by swift application forwarding procedure and onward resolution followed by the feedback; but also bring their applications into the notice of the higher officials of MCS

Table 1 Project Vision

1.5 Project Objective

1.5.1 Primary Objectives

1. Mobile/Computer platform for lodging all leave applications.
2. Make the leave process system much more quick and efficient.
3. Minimized time & effort to lodge the leave application, map it to the concerned authority, and provide feedback.
4. Leave record maintenance.

1.5.2 Academic objectives:

1. Developing a web based application.
2. Use of cloud based service as back end support for data storage, synchronization and computation.
3. Mapping of the leave application against each concerned authority, based on the nature of leave application.
4. Introducing a feedback mechanism to enable the applicant to keep track of his application.

1.5.3 Application / End – goal objectives:

1. To facilitate the students to launch their leave applications efficiently.
2. To enable the concerned authorities to be informed timely and enable them to provide their response effectively.
3. To save the precious time and efforts by automating the system.
4. To enable authorities to keep a track of all the leave records in a convenient way.
5. To enable the applicant to keep track of his applications.

1.6 Deliverables

1. Complete working project
2. Web based application
3. Documentation
4. Tutorials
5. User Manual

Chapter 2

2 Literature Review

2.1 Introduction

As an Army institution, MCS follows a strict policy and hierarchy in the case of leave applications. The current procedure is carried out manually by forwarding the application to the concerned authority's office and waiting for the response. Each office has a Runner who brings the application to the authority's office. The length of the chain of command is based upon the nature of leave and the SOPs as defined in Rules.

With the modernization and automation of everything else, the leave process was lagging behind since its almost the only process carried out manually amongst the rest. We carried out interviews of administration and students to identify pain points and deficiencies in the current system.

2.2 Problem Domain

This entire process inherits many problems including:

1. The delay and inefficiency in the leave process.
2. Extra effort to maintain and see leave records of personnel.
3. Check and balance on the personnel on leave.
4. Non-transparent leave process.
5. Absence of concerned authority

It necessitates the use of technology to minimize the effects of this current disparity by building a web–based application to enable the students to lodge their applications to the concerned authority for responding application promptly and efficiently.

2.3 Related Work

No work of similar nature has been carried out within MCS.

2.4 Shortcomings/issues

1. Internet connection is mandatory to login the application on your platform.

2. The contents of the application will be in English language only.

3. The server will be unavailable in case of maintenance and testing issues. No backup server configuration is provided.

2.5 Proposed Project

Every day many faculty members and students have to apply for leave for different reasons. Its hectic job for the authorities and the applicants as well to follow a manual system that is currently in practice in MCS. The applicant first has to visit the administration to write their name in the registers or get an application printed, which are then carried by the staff members to each authority that is included in the hierarchy for recommendation and approval. The authority maybe absent or sometimes the staff member may take longer than usual to get the applications approved. The applicant may have to wait for days, not knowing about the status of his/her application. The system is inconvenient and inefficient.

The LRAS will automate this entire system making it more efficient and convenient for applicant, staff members and authorities. The system will be transparent as the user will be able to keep a track of the application status. The authorities will be notified on their account about any application they have to recommend or sign. If the authority is not present due to any reason, the rights will be shifted to the higher authority. The authorities will not have to go through papers to keep a track of all the faculty or student members on leave. An overall leave record will be maintained for each user showing their leave history. The system will reduce the workload of leave approval in MCS to none.

2.6 Deliverables

2.6.1 Software Requirement Specification (SRS)

The purpose of this document is to present a detailed description of the Leave Recommendation and Approval System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, its entire process, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the users. It will explain how the system will particularly help faculty and students in the leave approval process.

2.6.2 Software Architecture Document

In this document, the overall architecture of the system is discussed, including the introduction of various components and subsystems. It is mainly supported by system Architecture diagram which shows an insider's perspective of the system by describing the high-level software components that perform the major functions to make the system operational.

2.6.3 Software Design document

The design document captures all our functional requirements and shows how they interact with each other conceptually. The low-level design also shows as to how we have been implementing how we are going to implement all of these requirements.

2.6.4 Implementation code Document

The implementation code document provides details about the pseudo code for the application and project prototype.

2.6.5 Software Testing Document

This document has testing modules in which there are certain test cases which depicts the correctness and accuracy of the project.

2.6.6 Final Project Report

This is the thesis report which compiles all the previous and current working for the project. Thesis report provides the whole summary for the project and also give details about each and every aspect of the project starting from introduction of the project, literature review, requirements leading to design discussions then testing and lastly future work and conclusion.

2.6.7 User Manual

User Manual gives details about the use of the end product. It contains details as how to use the product. Its functionalities and details of every aspect as how that works and how to use it. User Manual is for users to get to know the product.

2.7 Technological Requirements

LRAS system requires following software and hardware requirements.

2.7.1 Software Interfaces

1. LRAS should be able to run on any version of the following Web browsers: Microsoft Internet Explorer, Mozilla Firefox, Netscape, Opera, Safari and Google Chrome.

2. Primary Operating System supported by LRAS Interface will be Windows 8.1
3. LRAS should be able to run on Apache Web server configured in a stable Linux/Unix/MAC/Windows machine.
4. LRAS should be able to work with PHPmyAdmin database management system.

2.7.2 Hardware Interfaces

2.7.2.1 Computer System

1. System shall have keyboard input.
2. System shall have mouse input.
3. System shall have a monitor.
4. System shall have a working internet connection and the hardware requirements that come with it (Network card, Ethernet Port, Modem etc.)

2.7.2.2 Mobile Device

1. Any platform that supports a web browser.

2.7.2.3 Web and Database Server

1. To process requests and retrieve/store data.

2.7.3 Communications Interfaces

1. System shall be connected to the web services that we will create.
2. To access the application data, SQL queries will be used.
3. Communication between the Web Interface and the server will be through HTTP over a web browser.

2.7.4 Programming Interface

Programming interfaces for project are:

1. XAMPP
2. WAMPP

Chapter 3

3 Overall Description

3.1 Product Perspective

Every day many faculty members and students have to apply for leave for different reasons. Its hectic job for the authorities and the applicants as well to follow a manual system that is currently in practice in MCS. The applicant first has to visit the administration to write their name in the registers or get an application printed, which are then carried by the staff members to each authority that is included in the hierarchy for recommendation and approval. The authority maybe absent or sometimes the staff member may take longer than usual to get the applications approved. The applicant may have to wait for days, not knowing about the status of his/her application. The system is inconvenient and inefficient.

The LRAS will automate this entire system making it more efficient and convenient for applicant, staff members and authorities. The system will be transparent as the user will be able to keep a track of the application status. The authorities will be notified on their account about any application they have to recommend or sign. If the authority is not present due to any reason, the rights will be shifted to the higher authority. The authorities will not have to go through papers to keep a track of all the faculty or student members on leave. An overall leave record will be maintained for each user showing their leave history. The system will reduce the workload of leave approval in MCS to none.

3.2 Product Functions

1. Account creation
2. Leave application Template
3. Application forwarding based on Application category
4. Application Tracking

5. Hierarchy Generation
6. Updating the Application status
7. Role of Administrator
8. Application Printing
9. Check and Balance on the Leaves
10. Ability to Leave Remarks
11. Text message system

3.3 User Classes and Characteristics

Leave recommendation and approval system is being developed for different classes of the users which are:

1. **Administrator User: -** The administrator user will be able to maintain database and provide any other assistance in case of inconvenience to the users.

2. **Faculty User: -** The faculty can be civil or military. They can apply for applications and provide response to application if happen to be in chain of command

3. **Student User: -** All types of students can apply for leave

There are certain job or rights of each and every user.

End Users	Job/rights
Administrator	1. Login 2. Add User 3. Generate Hierarchy 4. Remove User 5. View/Update Profile

	6. Check Record
Faculty User	1. Login 2. Recommend Application 3. Approve Application 4. Check Record 5. Check Application Status 6. Apply for leave
Student User	1. Login 2. Apply for Leave 3. Check Application Status 4. Check Record

Table 2 Users and Characteristics

3.4 Operating Environment

OE-1: LRAS should be accessed using any popular versions of the Web browsers including Microsoft Internet Explorer, Mozilla Firefox, Netscape, Opera, Safari and Google Chrome.

OE-2: The Web based system of LRAS shall run on the computer system with following specifications:

1. Pentium 4 or Higher CPU
2. At least 512MB of RAM
3. At least 1 GB of free disk space
4. Windows XP or later operating system
5. Chrome, Firefox or Internet Explorer
6. Color monitor and a working internet connection.

OE-3: LRAS should be managed with PHPmyAdmin database management system.

OE-4: LRAS will be able to run on any Mobile/Computer with a working internet connection.

3.4.1 Technology Platform:

3.4.1.1 Web-Based Front End:

Application will be developed for web browsers including Chrome, Firefox and Opera.

3.4.1.2 Cloud Service:

Application's core part would be deployed on the cloud for the computation and data mapping. web-hosting service like www.codexify.com would be used for the database handling.

3.4.1.3 Programming languages:

1. HTML
2. PHP
3. JavaScript
4. CSS

3.4.1.4 Programming Environment

1. XAMPP
2. WAMPP

3.4.1.5 Database

1. PHPmyAdmin

3.5 Design and Implementation Constraints

CO-1: Web browser is required to run the web application for the end-user.

CO-2: All HTML code should conform to the HTML 5 standard.

CO-3: Lack of user-expertise in using web browsers**.**

CO-4: Internet connection needed**.**

CO-5: Use of English language as the only means of communication in the system

CO-6: Database queries should be written using standard methods.

3.6 User Documentation

UD-1: Final release will be accompanied with a user guide to inform users how to use Leave Recommendation and Approval System(LRAS). User documentation that would be delivered along with the final product

1. User manual

3.7 Assumptions and Dependencies

AS-1: Basic assumption for development of LRAS is that system should be available 24x7 since leave may be applied at any time.

AS-2: The users will not misuse the application to apply for leave for inappropriate reasons.

AS-3: The administration officials will be honest and will register every user with verifiable credentials.

AS-4: The user should be willing to stand by his/her application.

AS-5: The server will be able to handle a large number of request.

AS-6: Users of Leave Recommendation and Approval System (LRAS) should be assumed to have a laptop or a mobile phone with internet access.

D-1: There will be a permanent dependency on the internet, as without it the system will not work at all.

D-2: System is dependent on the online servers from where all the users will be accessing their data.

Chapter 4

4 Software Requirements Specification

4.1 System Features

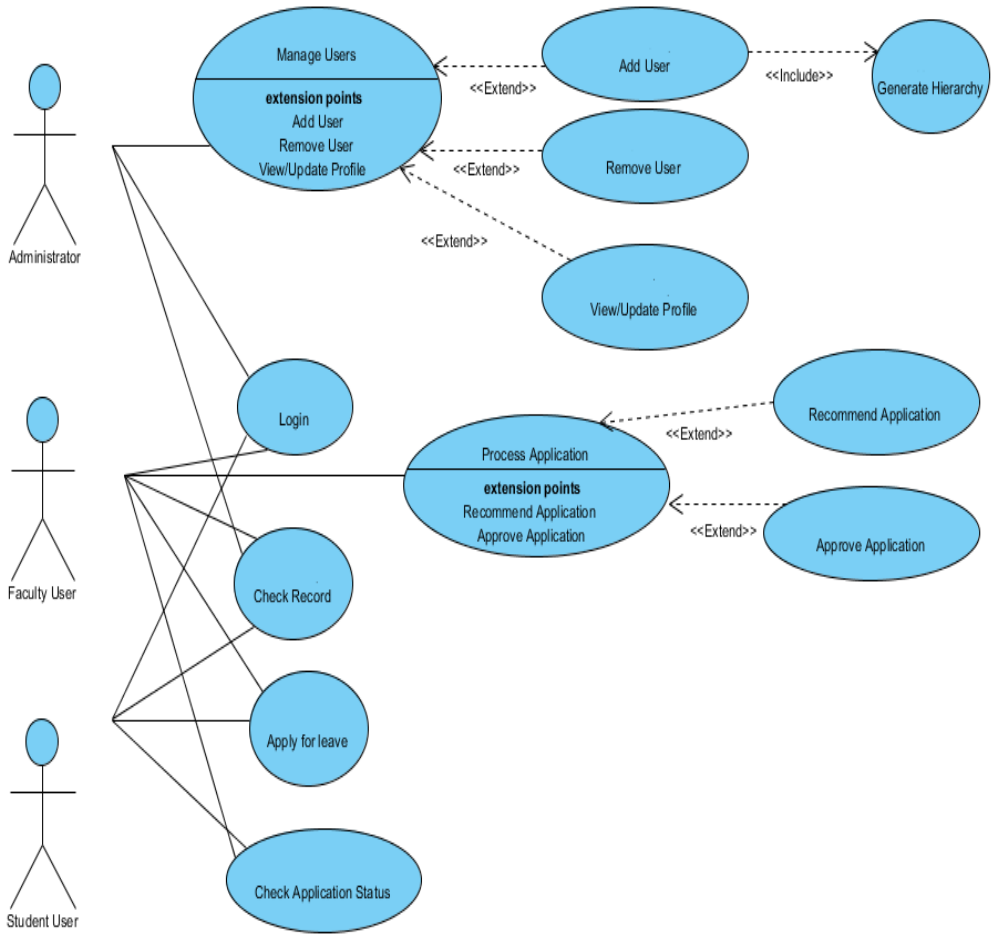

Figure 1 Use Case Diagram

4.1.1 Account creation

4.1.1.1 Description and Priority

The system will enable the system admin to create the accounts, while providing them with the access rights based on their hierarchy in MCS. The account creation is for the MCS faculty/students only. This account will enable them to login and process the applications. The priority of this system feature is high.

4.1.1.2 Stimulus/Response Sequence

Input: The MCS staff will provide the credentials of all the users to create a user

Output: The account with specific rights to each person will be created.

4.1.1.3 Functional Requirements

REQ-1: The LRAS shall allow the administrator to generate requests for account-creation.

REQ-2: The LRAS shall proceed the requests of account-creation to the system administrator.

REQ-3: The LRAS shall allow the administrator to create another administrator through the permission of concerned authorities.

4.1.2 Login/access rights

4.1.2.1 Description and Priority

The system will enable the users to login. The users will provide their unique ID and password to access their account so that they would be able to submit, recommend or approve of the application based on their roles in the system.

4.1.2.2 Stimulus/Response Sequences

Input: User will enter login credentials to the system.

Output: The system will grant the valid user, the access to the features according to his/her access rights.

4.1.2.3 Functional Requirements

REQ-1: The LRAS shall allow the users to login to their account.

REQ-2: The LRAS shall grant the logged in user, the access to the system features.

4.1.3 Log maintenance

4.1.3.1 Description and Priority

A log of all application will be maintained on the database, accessible to the admin and the user. The log will contain the record of all the applications and their current status.

4.1.3.2 Stimulus/Response Sequences

Input Admin will populate the database with the users and appointments.

Output: The log of the all the users and respective appointments will be maintained.

4.1.3.3 Functional Requirements

REQ-1: The LRAS will maintain the log of all the applications.

REQ-2: The LRAS will allow the admin to see all the users.

REQ-3: The LRAS will allow the user to check the status of his/her application.

4.1.4 Application Details

4.1.4.1 Description and Priority

The applicant should be able to enter a description of his/her application and category.

4.1.4.2 Stimulus/Response Sequences

Input: The user will lodge an application and will add a description.

Output: The application will be registered along with the user description.

4.1.4.3 Functional Requirements

REQ-1: The LRAS shall allow the applicant to enter the description of his/her application.

REQ-2: The LRAS shall allow the applicant to specify the category of application, under which the leave application falls.

4.1.5 Task assignment to the concerned authority

4.1.5.1 Description and Priority

Application lodged by the user will be assigned to the concerned authority, based on the nature of the application.

4.1.5.2 Stimulus/Response Sequence

Input: Users will populate the applications.

Output: The system will forward the application to respective authority.

4.1.5.3 Functional Requirements

REQ-1: The LRAS shall assign the registered application to the concerned authority.

4.1.6 Leave history

4.1.6.1 Description and Priority

The system will maintain a record of applications and show the user his/her history.

4.1.6.2 Stimulus/Response Sequences

Input: The user will lodge an application.

Output: The system will assign an application number to the application and maintain a record.

4.1.6.3 Functional Requirements

REQ-1: The LRAS shall generate an application number against a lodged application.
REQ-2: The LRAS shall maintain a record of all the applications.

4.1.7 Post Application Approval Feedback

4.1.7.1 Description and Priority

The user will be notified about the progress of the application. Each time a recommendatory authority passes on the application, the user will be notified.

4.1.7.2 Stimulus/Response Sequences

Input: Whenever the concerned authorities recommend or approve the application.

Output: The system logs will be updated and the user will be notified.

4.1.7.3 Functional Requirements

REQ-1: The LRAS will allow automatic update the application status.

REQ-2: The LRAS will update the number of students on leave. It will be visible to the concerned authorities.

4.1.8 Leave Remarks

4.1.8.1 Description and Priority

The authority will be able to provide remarks with the response.

4.1.8.2 Stimulus/Response Sequence

Input: The authority will give remarks with the response.

Output: Remarks will be attached with application and will be shown to everyone in the hierarchy.

4.1.8.3 Functional Requirements

REQ-1: The LRAS shall allow the authority to give remarks with application.

4.1.9 Text Messages Reminder

4.1.9.1 Description and Priority

If the concerned authority does not provide a response for a specified time period then a text message will be generated which will be sent to authority's phone.

4.1.9.2 Stimulus/Response Sequence

Input: The specified timeframe for the application-resolution is passed whereas no response is received.

Output: A remainder will be generated by the system and sent to the concerned authority.

4.1.9.3 Functional Requirements

REQ-1: The LRAS shall send a reminder to the concerned authority if the application does not receive any response.

4.2 Other Non-functional Requirements

4.2.1 Performance Requirements

Certain functionalities will be required, based on the performance and response of LRAS. LRAS has to be efficient in its response and operation. The product domain requires that the software is optimized in terms of performance. The data flow should happen in the most efficient way.

4.2.2 Safety Requirements

SF-1: Once the system is set up, the database will be exported after every 3 days.

4.2.3 Security Requirements

SE-1: Users shall be required to log in to the LRAS for their own credential information.

SE-2: The system shall permit only authorized members to do administrator's task.

SE-3: The system shall permit users to view only their own profile and data that are intended for them.

SE-4: The system must perform an encoding technique such as hashing to save all passwords securely.

SE-5: The System will provide confidentiality and integrity.

4.3 Software Quality Attributes

Quality attributes of LRAS are described below. By following these attributes, the quality of LRAS will be improved.

4.3.1 Runtime System Qualities

Some of the runtime qualities that should be considered in the development of LRAS are described here.

4.3.1.1 Functionality

LRAS must provide the functions of authentication of user.

4.3.1.2 Availability

LRAS should be available 24x7 since the application can be lodged at any time.

4.3.1.3 Usability

Usability is an important criterion in the development of LRAS. The system should present all functionalities in such a way that nothing is missed by the user.

4.3.2 Non-Runtime System Qualities

These are qualities of LRAS which are required to make this software useful for further enhancements. It will also be helpful in future development as well as extending system to different environments.

4.3.2.1 Modifiability

LRAS must support modifiability so any further improvements or features are easy to incorporate.

4.3.2.2 Portability

The system should work on WIFI as well as 3G network.

LRAS should be able to run in different computer environments. The LRAS server should be a platform-independent and should support interoperability.

4.3.2.3 Interoperability

The system shall work on Personal computers as well as mobile devices and the specified web browsers.

4.3.2.4 Testability

Different quality tests should be performed so that LRAS is free from faults and perform according to requirements.

4.4 Other Requirements

The system will be providing a scalable solution with expected increase in the number of users.

Chapter 5

5 System Design Specifications

5.1 Overview of the module

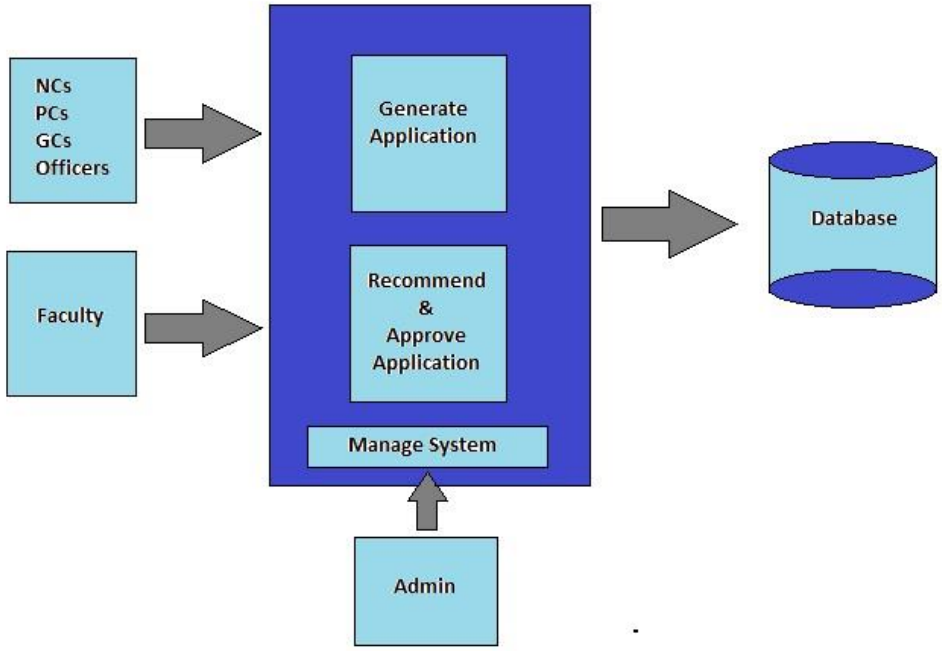

Figure 2 Overview of the Module

There are four types of Users in LRAS

- Student
- Officers
- Faculty

Each User will have his account which will be added by the administrator. Users will choose the type of application and specify the reason along with it.

In case of faculty, they will be serving dual-purpose i.e. apply for leave and grant leave. Admin will be managing the database i.e. add or delete users and courses. Manage the hierarchy. Based on the type of user and type of application, the hierarchy will be generated. The application will be sent to the respective authority. The authority that receives the application will either approve or recommend the application based on his role and type of application. All the login information, application data, user accounts, hierarchy tables and leave records will be stored in the database. Manage system is the function of Admin. It defines the rights of admin.

5.2 Architecture

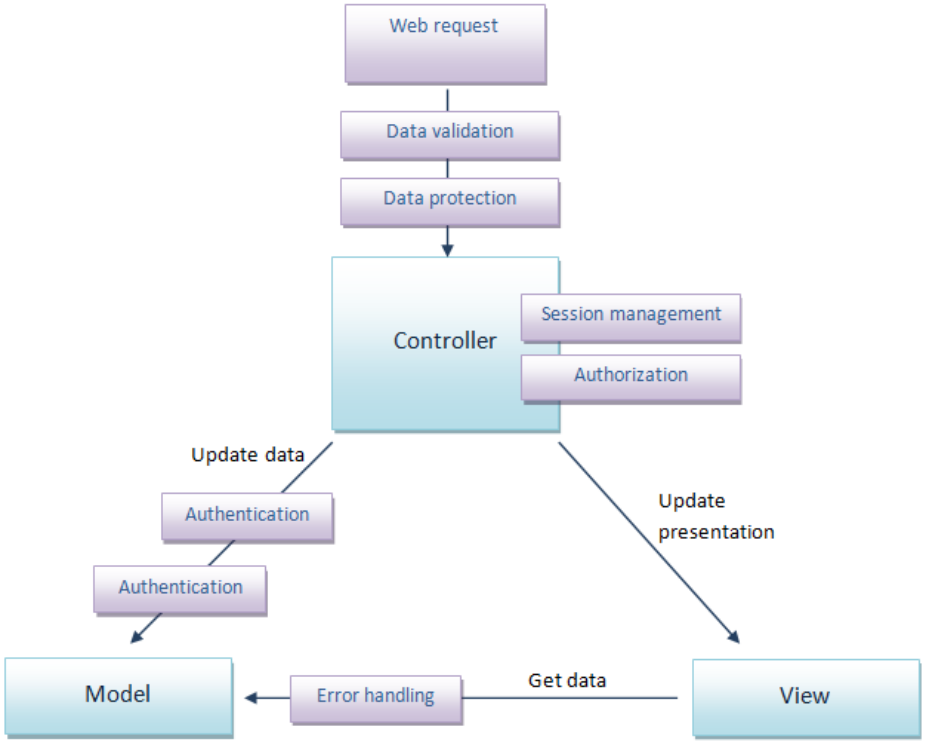

Figure 3 Architecture Diagram

Model View Controller (MVC) will be used to implement LRAS. There will be three major components of our application. Essentially, it allows us to isolate these very separate pieces of code into their own domain, which makes code maintenance and debugging much

simpler than if all of these items were chunked into one massive piece. Error tracking in this case will be very efficient. Development can be carried on concurrently and integrated later on.

5.3 Overall structure of the system

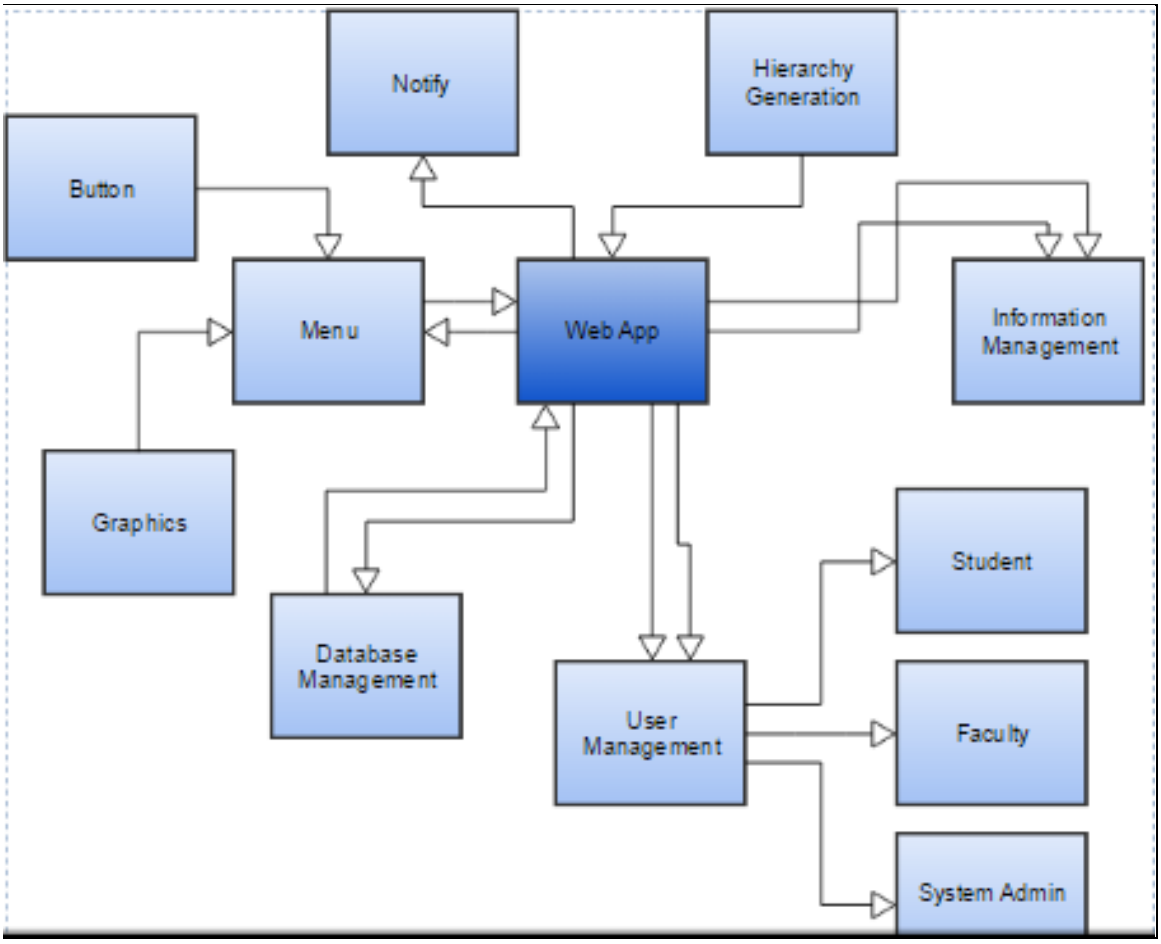

Figure 4 overall Structure of the system

5.4 Brief Description of the components

Following is a brief overview of the major modules / components of the system, mentioned in the above diagram:

Module name/ components	**Purpose**
Menu	Menu has buttons having links of other menus.
Button	Touch buttons as input class. To select a particular category
User management	All types of users are managed in this via login names and password according to their access rights.
Students	Students include all the NCs, GCs, PCs, Officers
Faculty	Faculty include both Military and Civil Faculty.
System Admin	System admin is treated as a user because he too will have an account, just with more rights.
Database Management	Database management will be carried out by the admin and the database will save all the information about the student and hierarchy tables.

Graphics	It is a module to draw and display everything on the screen.
Web App	Stores the information about the app such as all the applications, current and pending application.
Notify	It provides the notification to the users and authority about applications in their respective domain.
Hierarchy Generation	Hierarchy generation is carried out based on the information provided to the system. It varies based on the type of user.
Module name/ components	**Purpose**
Menu	Menu has buttons having links of other menus.
Button	Touch buttons as input class. To select a particular category

Table 3 Components and purpose

5.5 Class Diagram

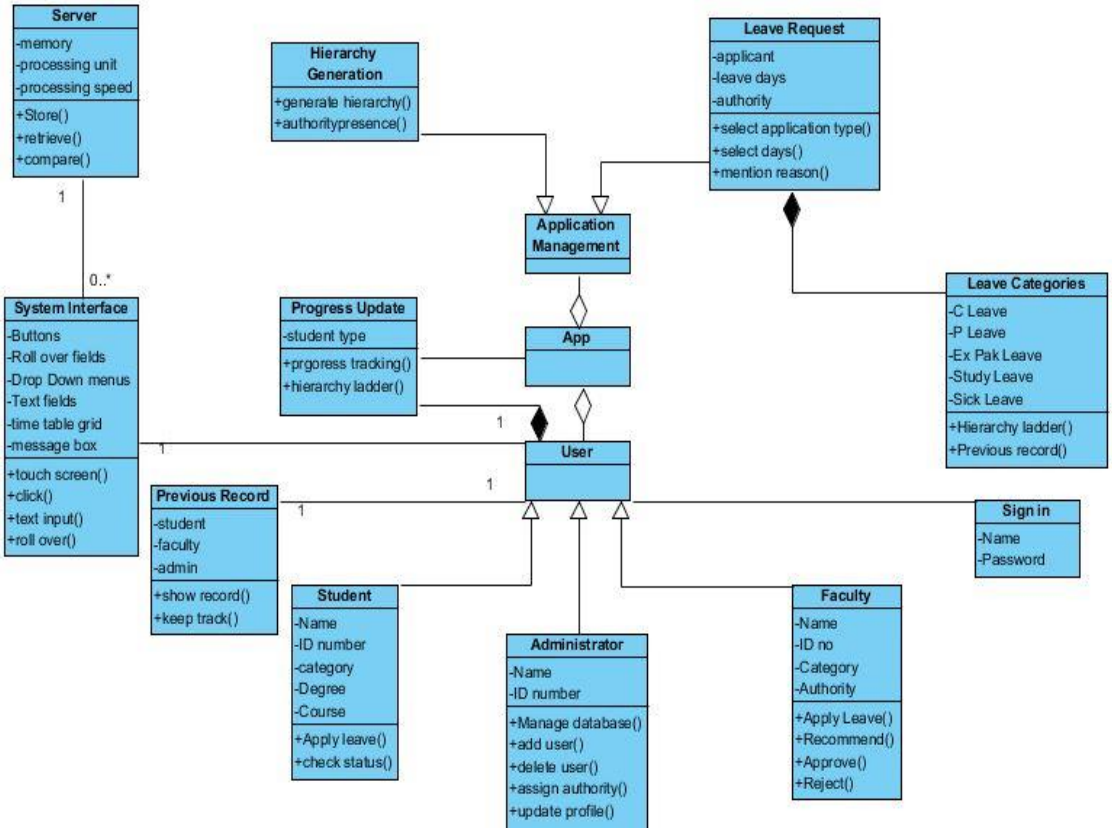

Figure 5 Class Diagram

Class name	**Description**
App	App class contains all the information that LRAS has to perform. It is the main class which will be acting as a gateway to all the other classes
Application management	Application management class will be handling all the functions and other classes related to applying for leave and generating hierarchy.

User	User class contains all the information related to user management. Generalization with other classes of user categorization and the functions that performs all the user management functions.
System interface	It contains all the information to enable user to interact with LRAS. It has links to all the functions of different classes that on selection lead to different actions.
Server	This class contains the database objects and server information. It provides access to all the database objects and also saves the data
Leave Request	The class provides all the data and options to apply for leave.
Hierarchy Generation	This class creates the hierarchy that is to be followed for the leave approval.
Progress Update	This class contains all the functionality to update the leave application progress status.
Number of Days	This class plays a significant role to determines the hierarchy ladder
Leave Categories	This class offers the user/faculty the types of leave they want to submit an application for.

Users	This class handles the types of users and their interaction with the system.
Faculty	This class handles the faculty users of the system and their role in the hierarchy.
Administrator	The class handles all the functions that an administrator can perform and maintains the rights of an administrator.
Student	This class handles all kinds of users i.e. NCs/PCs/GCs/ASCs and Officers.
Sign In	This class handles the login system for LRAS.
Previous Record	This class contains the leave history of all the users of the LRAS

Table 4 Class description

5.6 Entity Relationship Diagram

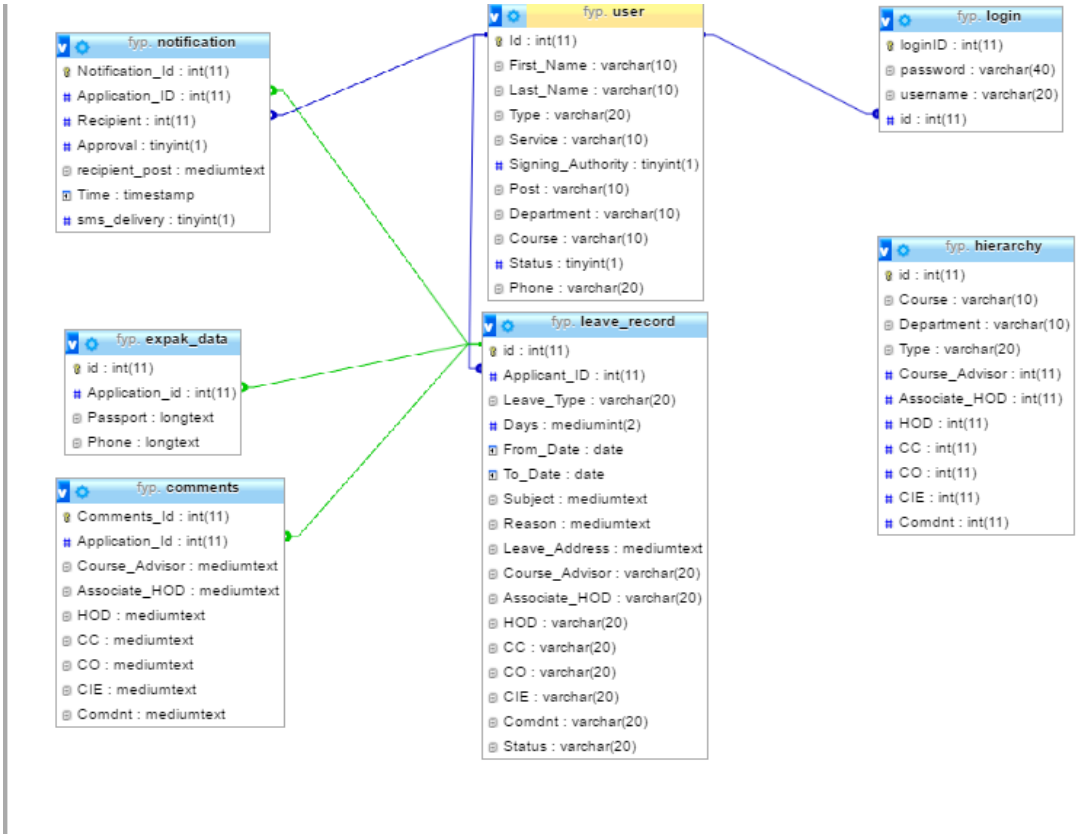

Figure 6 Entity Relationship Diagram

Leave record is directly linked with the Comments by the authorities and Notification of the application. It contains the data from both the tables and then maps the data to users through the User table. Ex-Pak Leave table is created separately since it goes out of the domain of MCS. Hierarchy table contains the hierarchy of each course respectively.

5.7 Use Case Diagram

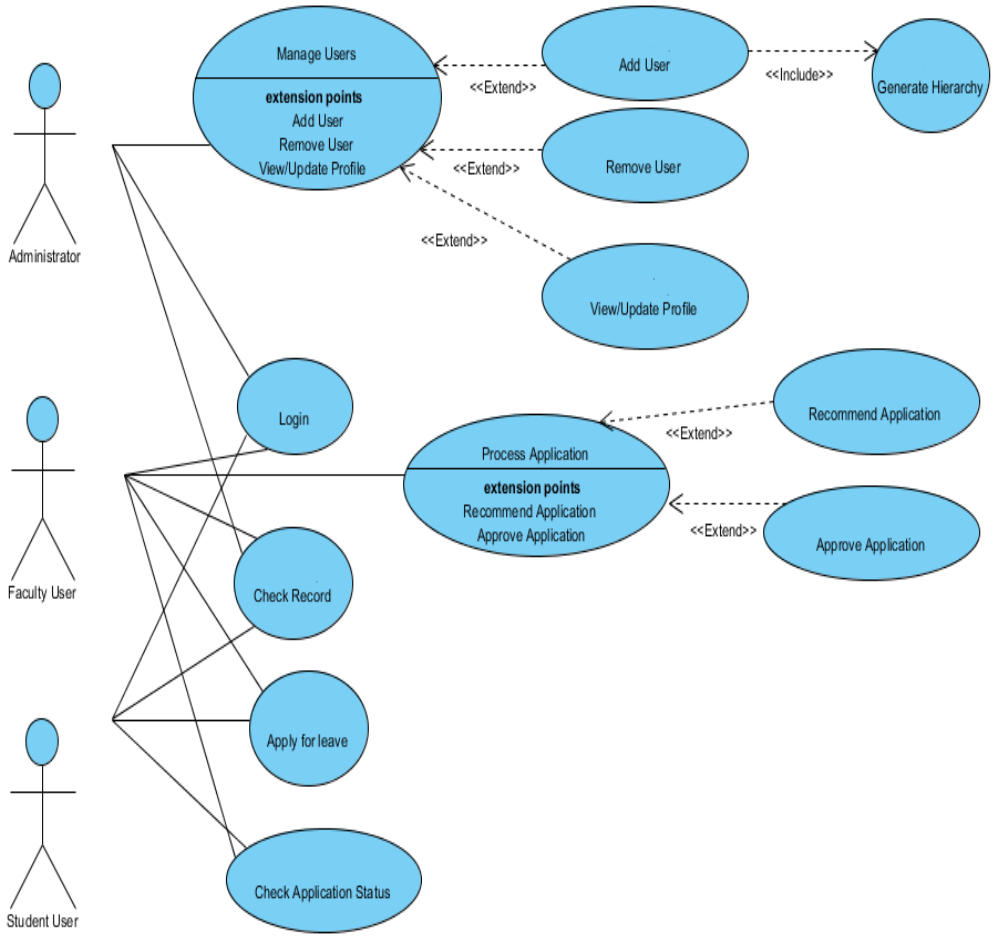

Figure 7 Use Case

5.8 Use Cases Description

5.8.1 Manage Users

Use Case ID:	1
Use Case Name:	Manage Users

Actors:	Administrator		
Created By:	Uzair	Created By:	Uzair
Date Created:	3/1/2017	Date Created:	3/1/2017
Description:	1. Admin has to login to the system to manage users i.e. create, view, update and delete user profiles.		
Preconditions:	1. Admin has to login.		
Post conditions:	1. The System must record the change.		
Normal Flow (primary scenario):	1. The actor creates, views, updates and deletes the user details. 2. Click on create or view or update or delete button as required.		
Alternative Flows:	The actor will contact the system maintenance team to check if there is some error with database systems and has to resolve the error.		

Table 5 Manage Users

5.8.2 Login

Use Case ID:	2		
Use Case Name:	Login		
Actors:	Administrator, Faculty Users, Student Users		
Created By:	Uzair	Created By:	Uzair
Date Created:	3/1/2017	Date Created:	3/1/2017
Description:	A user tries to login to the system.		
Preconditions:	1. User has to open the login page first.		

Post conditions:	1. If the use case was successful, the actor is now logged into the system. If not the system state remains unchanged.
Normal Flow (primary scenario):	This use case starts when an actor wishes to log into the System. 1. The system requests that the actor enter his/her name and password. 2. The actor enters his/her name and password. 3. The system validates the entered name and password and logs the actor into the system.
Alternative Flows:	1. **Invalid Name / Password** If in the *Basic Flow* the actor enters an invalid name and/or password, the system displays an error message. The actor can choose to either return to the beginning of the *Basic Flow* or cancel the login, at which point the use case ends.

Table 6 Login

5.8.3 Process Application

Use Case ID:	3		
Use Case Name:	Process Application		
Actors:	Faculty Users		
Created By:	Uzair	Created By:	Uzair
Date Created:	3/1/2017	Date Created:	3/1/2017
Description:	The system starts processing application.		
Preconditions:	1. User has to generate application first.		

Post conditions:	1. The system must verify the user and start processing the generated application.
Normal Flow (primary scenario):	1. The user must fill up the application form completely. 2. The user must click on the button “generate application”. 3. The system must record the application in the database. 4. The system must send the application to the concerned authorities as per the hierarchy assigned by the administrator. 5. The system must approve or recommend the application as per the domain of the authority.
Alternative Flows:	1. The user fills up the application form. 2. The system checks for the empty/left over credentials in the form. 3. The system displays an error report if the user didn’t fill the form completely and tells the user to fill up completely.
Use Case ID:	3

Table 7 Process Application

5.8.4 Edit/View Profile

Use Case ID:	4		
Use Case Name:	Update/View Profile		
Actors:	Administrator		
Created By:	Uzair	Created By:	Uzair
Date Created:	3/1/2017	Date Created:	3/1/2017

Description:	Admin tries to update/view the profile of a user.
Preconditions:	1. Admin has to log in first.
Post conditions:	1. The System must show the profile and record the changes made therein.
Normal Flow (primary scenario):	1. The admin logs in to the system. 2. The admin selects the user whose profile needs to be updated/viewed. 3. The admin clicks profile options. 4. The system enables the admin to view or modify the profile information of a user. 5. The admin can change all the info of the user also the username and password of the user upon request. 6. The system records the changes made in the profile.

Table 8 Edit or View profile

5.8.5 Check Application Status

Use Case ID:	5		
Use Case Name:	Check Application Status		
Actors:	Faculty Users, Student Users		
Created By:	Uzair	Created By:	Uzair
Date Created:	3/1/2017	Date Created:	3/1/2017
Description:	A user has to login to the system to check the status of the application.		
Preconditions:	1. User has to open his/her profile first. 2. User has to click on the “check application status” button.		

	3. User has to know the Date / ID of the application to check the status.
Post conditions:	1. The System shows the status of the application.
Normal Flow (primary scenario):	1. User clicks on the “check application status” button. 2. System shows the status of the application.
Alternative Flows:	**No Application Generated Yet!** 1. User clicks on the “check application status” button without generating any application before. 2. The system displays that “No application is generated yet!”.

Table 9 Check Application Status

5.8.6 Check Record

Use Case ID:	6		
Use Case Name:	Check Record		
Actors:	Administrator, Faculty User, Student User		
Created By:	Uzair	Created By:	Uzair
Date Created:	3/1/2017	Date Created:	3/1/2017
Description:	The system will enable the users to check their leave record.		
Preconditions:	1. User has to login to the system and click on the “Check Leave Record” button.		
Post conditions:	1. The system will generate the detailed report of leave record.		

Normal Flow (primary scenario):	1. The user will click on the “Check Leave Record” button. 2. The system generates the report of the user’s leave record. 3. The system allows the user to print the report of his/her leave record.
Alternative Flows:	1. The user has not even generated a single application yet. 2. The system returns message displaying “No application generated yet!”.

Table 10 Check Record

5.8.7 Apply for leave

Use Case ID:	7		
Use Case Name:	Apply for leave		
Actors:	Faculty Users, Student Users		
Created By:	Uzair	Created By:	Uzair
Date Created:	3/1/2017	Date Created:	3/1/2017
Description:	The system will enable the user to apply for leave.		
Preconditions:	1. User has to login to the system. 2. User has to click on the “Apply for Leave” button.		
Post conditions:	1. The system will update the status of the application.		
Normal Flow (primary scenario):	1. The user clicks on the “Apply for Leave” button. 2. The system records the changes and updates the status of application.		

Alternative Flows:	**Limited Connectivity** In case of connectivity issues with the database, the system will not be able to process, generate and update the status of the application and will show an error message “Connectivity Issues, please try later!”.

Table 11 Apply for Leave

5.8.8 Generate Hierarchy

Use Case ID:	8		
Use Case Name:	Generate Hierarchy		
Actors:	Administrator		
Created By:	Uzair	Created By:	Uzair
Date Created:	3/1/2017	Date Created:	3/1/2017
Description:	The system generates the hierarchy as selected by the administrator on the type of the user.		
Preconditions:	1. Administrator must be logged in. 2. Administrator must have added the new user.		
Post conditions:	1. The system generates the hierarchy for the user.		
Normal Flow (primary scenario):	1. The administrator adds the user. 2. The user type is specified on the time of adding of a new user. 3. The administrator selects the hierarchy for the user. 4. The system generates the hierarchy for the new user.		

Alternative Flows:	**Limited Connectivity** In case of connectivity issues with the database, the system will allow the admin to refill the credentials.

Table 12 Generate Hierarchy

5.9 Sequence Diagrams

5.9.1 Application Processing

The below diagram defines the sequence of actions that happens when an application has been requested by a user. All the actions are in time lined.

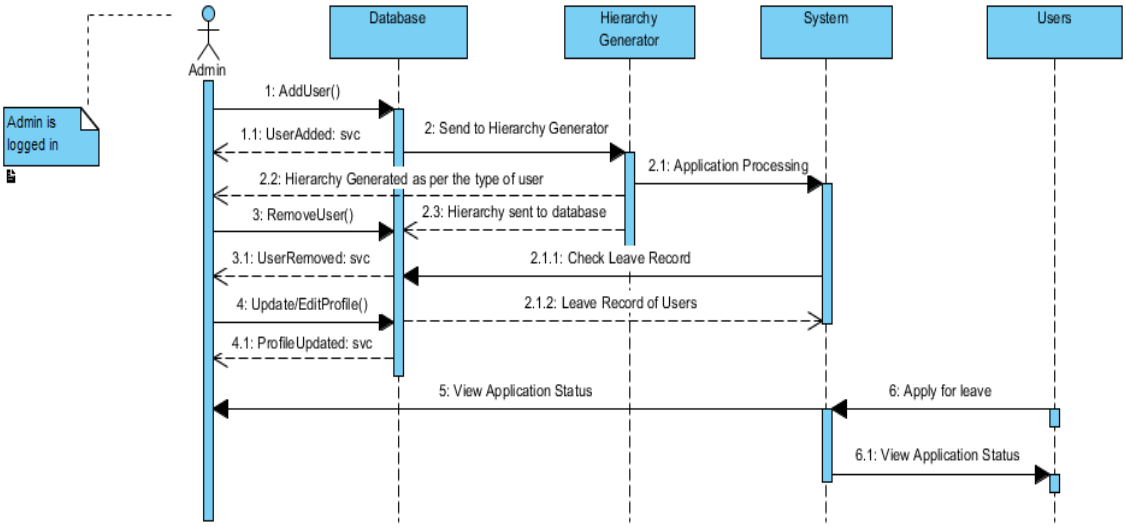

Figure 8 Application Processing

5.9.2 Student User

The below diagram defines the sequence of actions that happens when a student user applies for leave through LRAS.

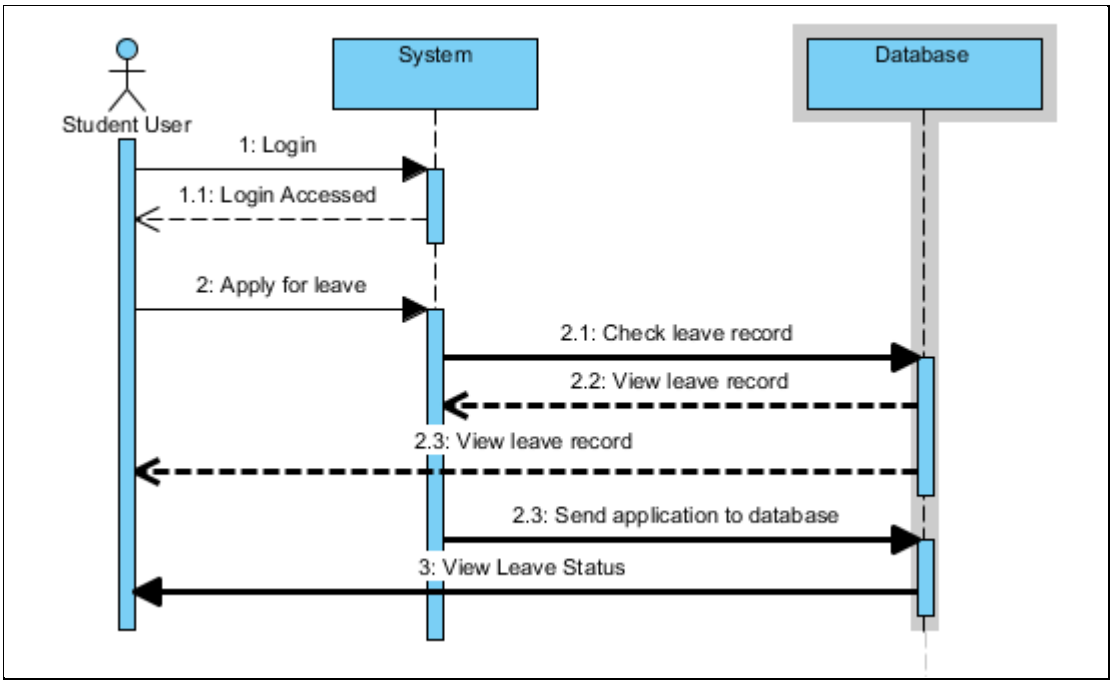

Figure 9 Student User

5.9.3 Edit/View Profile

The below diagram defines the sequence of actions that happens when a User tries to edit or view his/her profile. There is also defined optional outlook in the sequence of editing or viewing profiles.

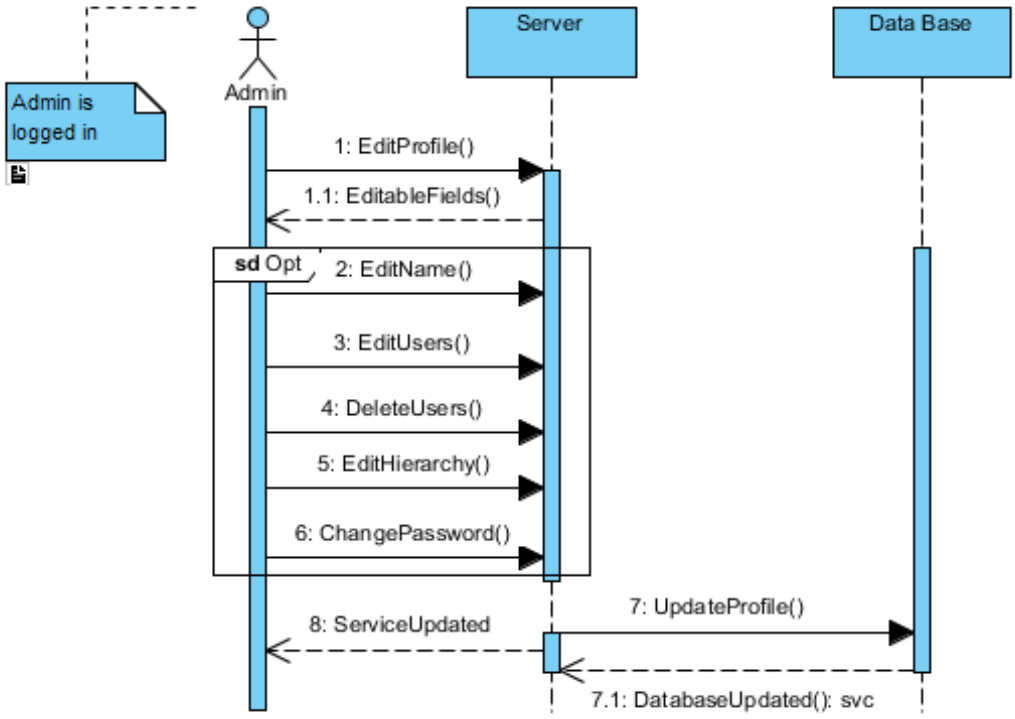

Figure 10 Sequence diagram Edit/View Profile

5.9.4 Faculty User

The below diagram defines the sequence of actions that happens when a faculty user applies for leave or being an authority he/she recommends or approves the applications of student users.

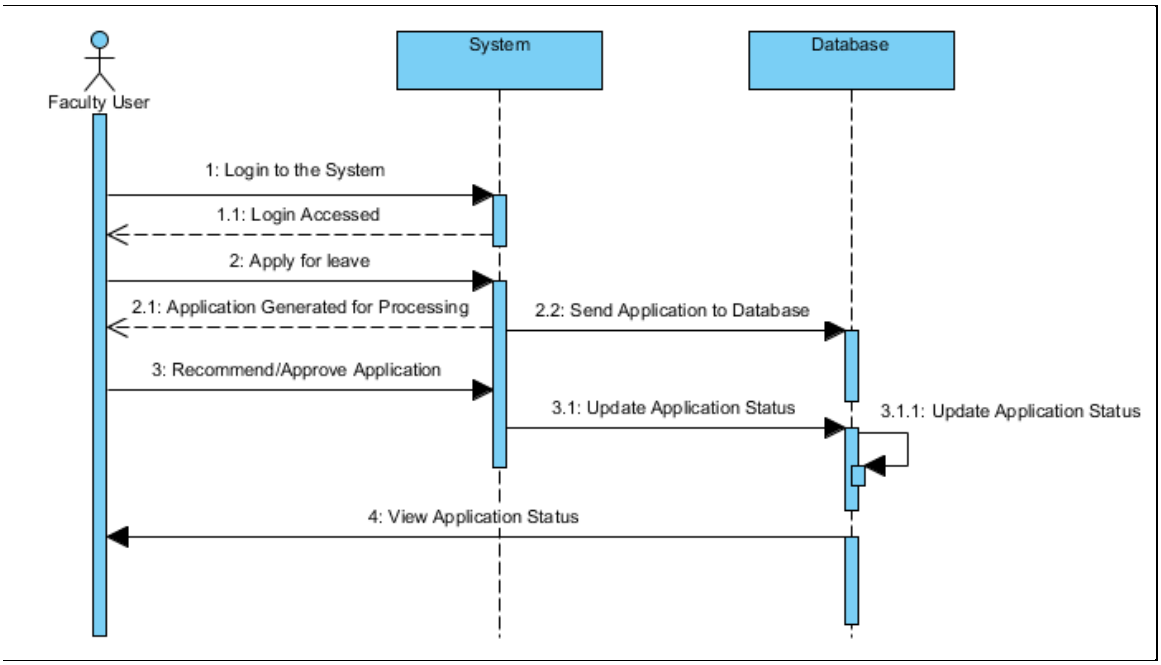

Figure 11 Sequence diagram for Faculty User

5.10 Activity Diagrams

5.10.1 Apply for Leave(Student/Faculty)

The below diagram defines the stream of activities that a user need to execute while applying for leave.

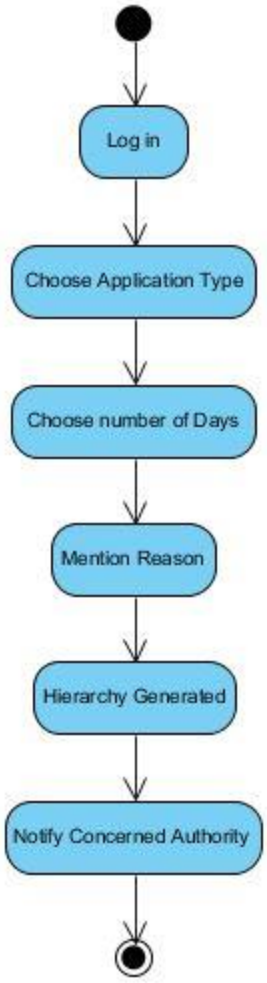

Figure 12 Apply for leave

5.10.2 Approve/Recommend Leave(Faculty)

The below diagram defines the stream of activities that a user(faculty) needs to execute while approving/recommending the leave application.

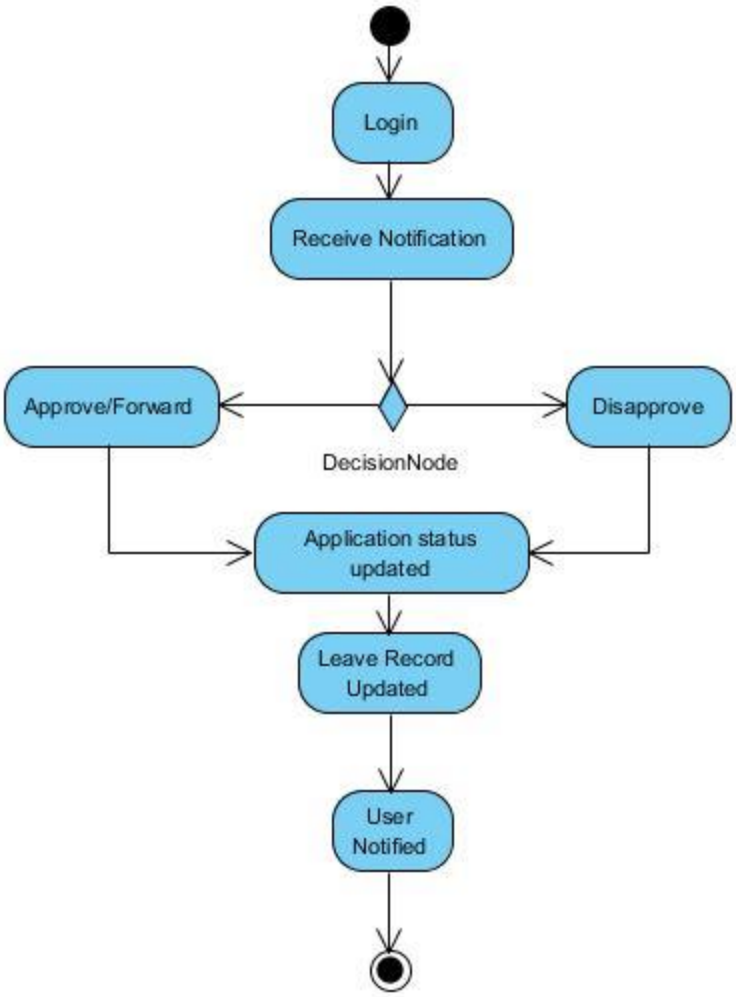

Figure 13 Approve/Recommend Leave

5.10.3 Add/Delete User(Admin)

The below diagram defines the stream of activities that are carried out by the admin for adding/deleting or assigning new roles to the users.

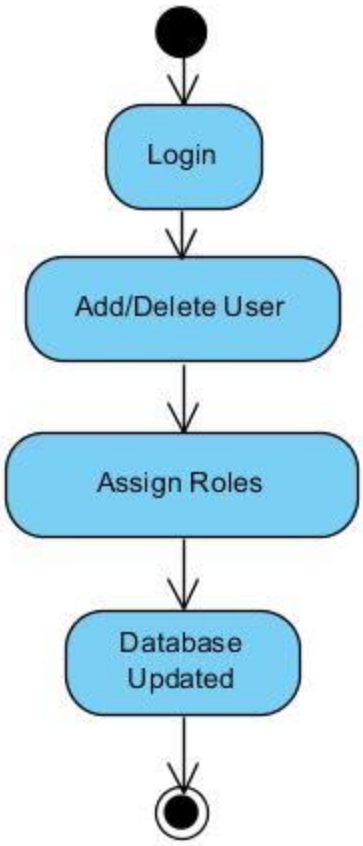

Figure 14 Add/Delete User

5.10.4 Absence of Authority

The below diagram defines the stream of activities that will be carried out when an authority is absent.

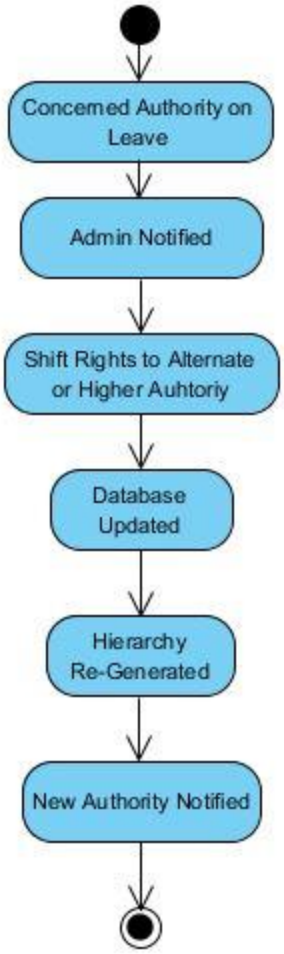

Figure 15 Absence of Authority

5.10.5 View Application Status

The below diagram defines the stream of activities that are followed when checking the application status.

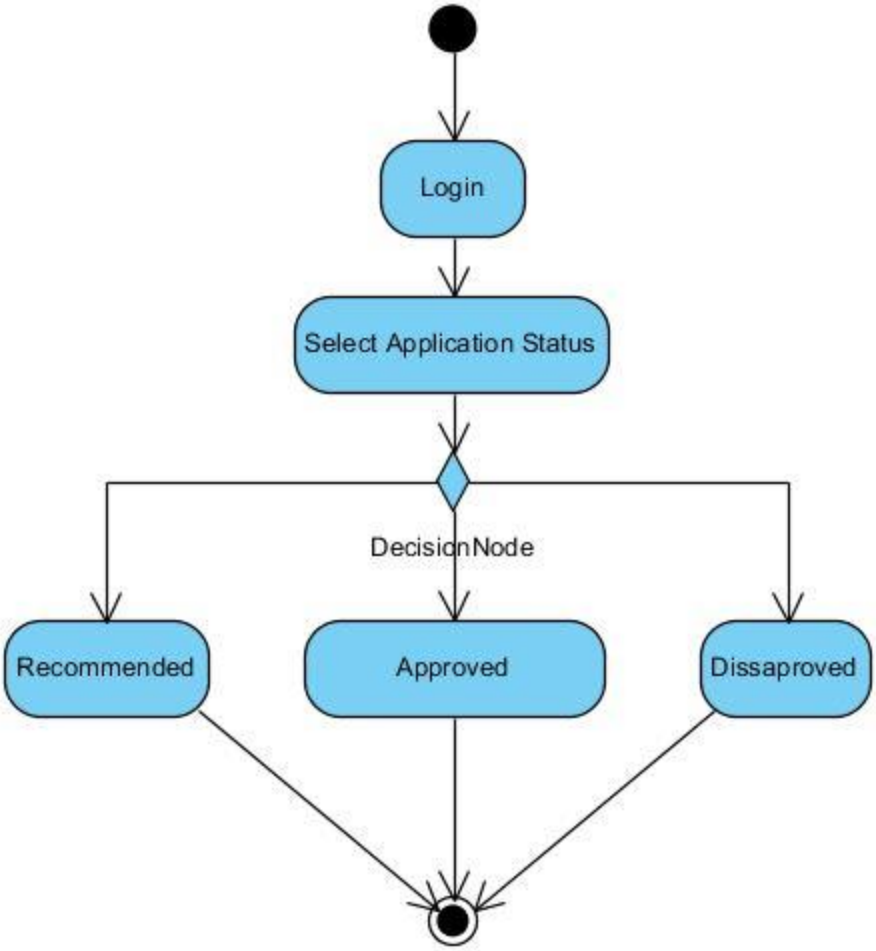

Figure 16 Update application status in web interface

Chapter 6

6 System Implementation

6.1 Technology Used

6.1.1 Programming Language Used

LRAS uses typical web development languages such as PHP, JavaScript, HTML5, CSS being used for configuration of the services and database, should be accessed using any popular versions of the Web browsers including Microsoft Internet Explorer, Mozilla Firefox, Netscape, Opera, Safari and Google Chrome.

6.1.2 Development Tools

XAMPP was used to create server and system was tested directly on web browsers (Chrome and Firefox).

6.1.3 Database

The systems Database was developed and managed using MySQL

6.1.4 Operating System

The Web Interface will be running on Microsoft Windows and can run on Windows XP and all future iterations of it. The mobile interface runs on all operating systems supporting web browsers such as Chrome, Firefox, Safari, Opera, etc.

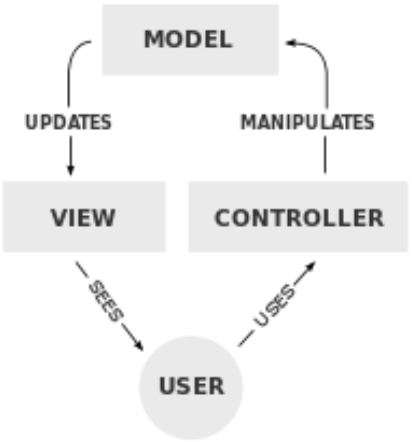

Figure 17 MVC Architecture

6.2 Complete System Implementation

The system contains of two main parts. One is for admin which will be able to manage all the admin tasks and the other one is for the students and faculty. The major modules of the web application will be discussed in detail in the subsequent sections.

6.2.1 Login Module

Module is linked with the database and forms the data access layer of the application. Business Layer consists of all the functions that are then accessed from the view. Usernames and Passwords stored in the data base are compared with Username and Password entered by the user.

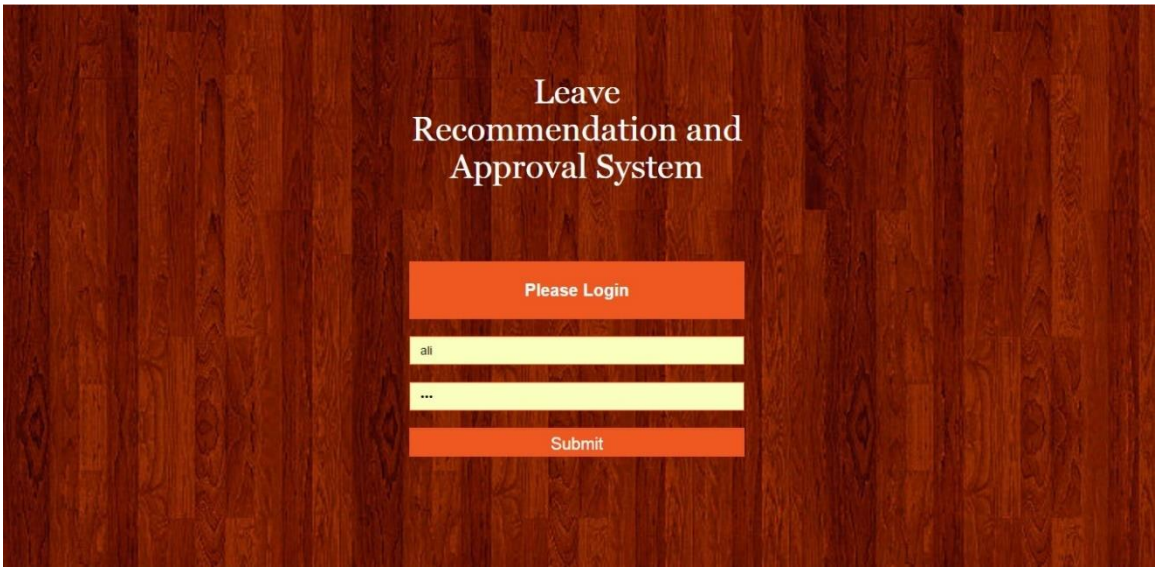

Figure 18 Login Module

6.2.2 Home Module

This is the main module, which greets the user following successful login. This is used to access all the different functionality of the application and can be considered as a main menu. It can be used to Generate applications, view previous applications generated by the user, view application status and to logout.

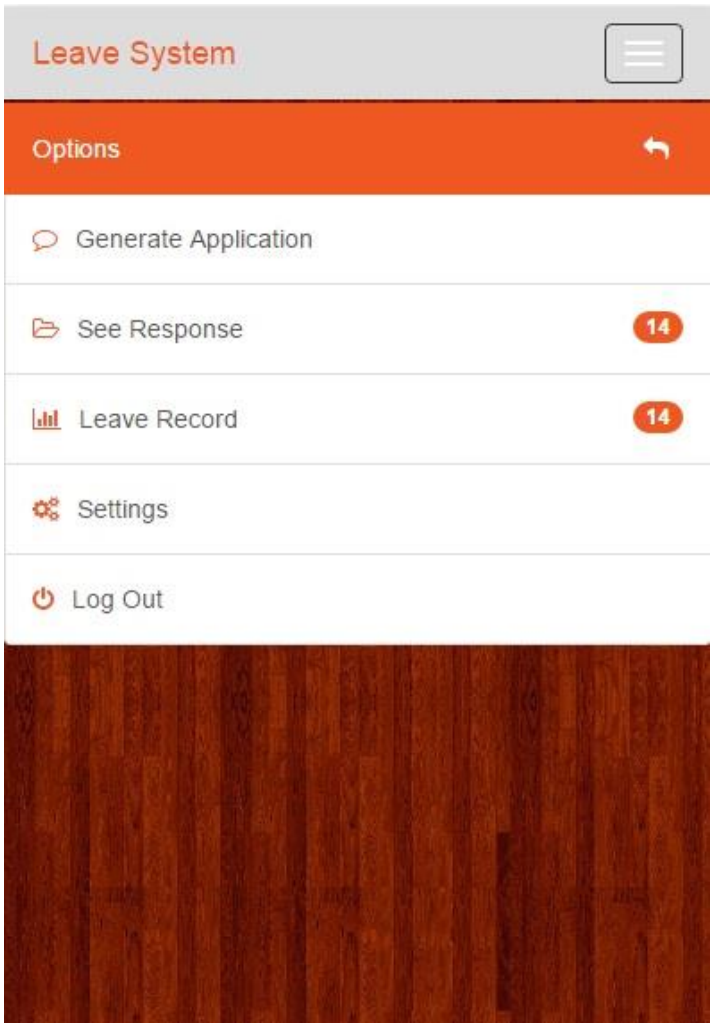

Figure 19 Home Module

6.2.3 Generate Application Module

This is the module which contains all the functionality relating to generation of an application, including viewing leave history. All these details are stored in the database, and a unique tracking Id for the application is issued to the user.

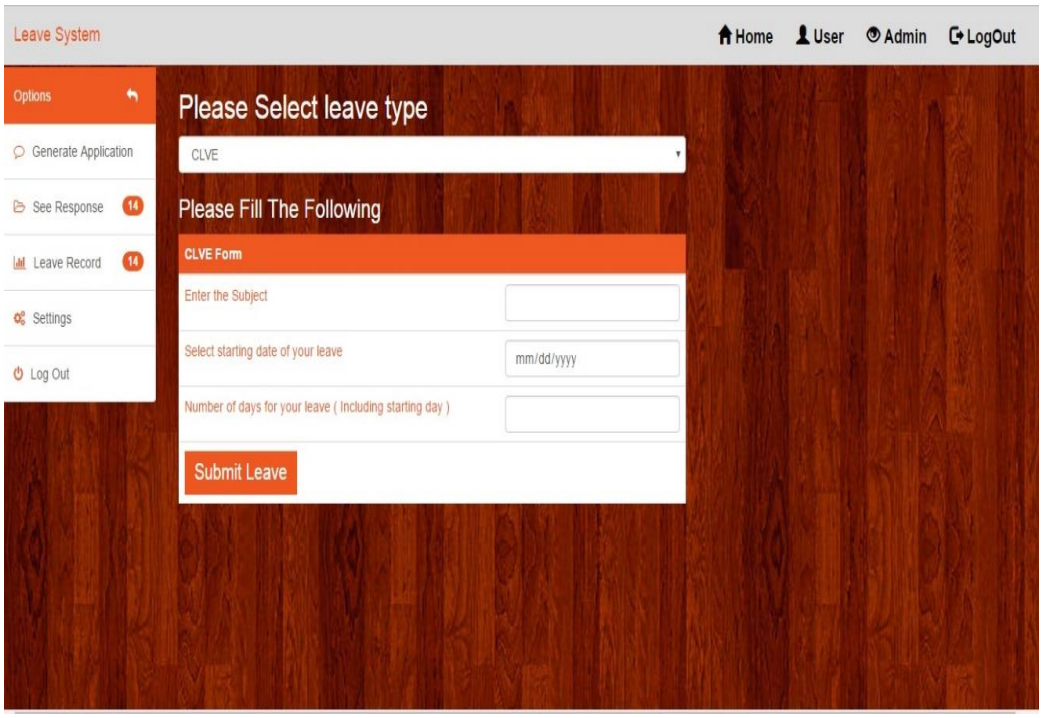

Figure 20 Generate Application Module

6.2.4 View Leave Application Module

This module enables logged in users to view the applications that they have generated previously and view the status that has taken place for those particular applications. It does so by seeing the tracking Id's linked with the account currently logged in to the application, then displays these. Upon selecting one, the details of that application are displayed.

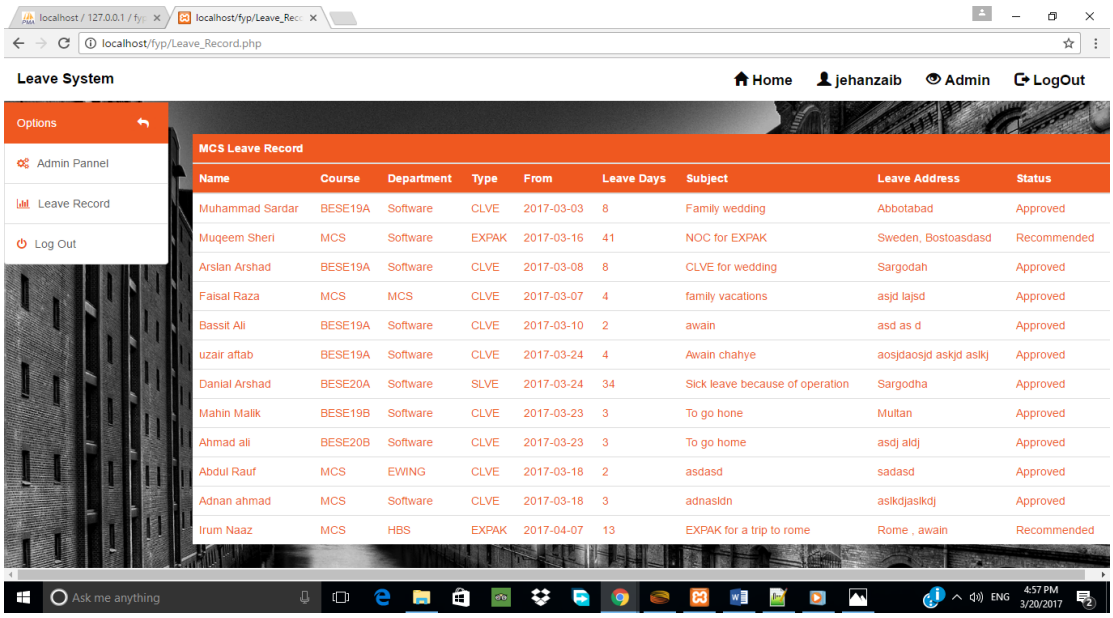

Figure 21 View Application Module

Chapter 7

7 System Testing

7.1 Over View

Testing of software projects involve different levels of testing to make sure that the software which is being developed is error and fault free. The different levels at which testing was done is discussed here.

7.2 Unit Testing

It involves the testing of each module at completion.

7.2.1 Login Feature Testing

Test Case Name	Accessing login screen	
Test Case No	1	
Description	Testing login screen	
Testing Technique Used	Unit Testing	
Preconditions	The user must be registered on the system	
Input Values	User name and password	
Valid Inputs	Valid user name and password	
Steps	1. Go the LRAS web application 2. Fill the required fields 3. Click “login”	
Expected Output	User have access to the main panel	
Test Procedure Valid Input Invalid Input		**Output** User has access to the main Panel Error message pops up

Table 13 Login Testing Feature

7.2.2 Add User Testing

Test Case Name	Add User	
Test Case No	2	
Description	Testing to add user to the system	
Testing Technique Used	Integration Testing	
Preconditions	Admin must be logged in to the system	
Input Values	Enter User Name	
Valid Inputs	Only alphabets	
Steps	1. Select Add User from the drop down in Admin Panel 2. Select the Name text fields 3. Input user names	
Expected Output	The system should approve the format of the names and allow the admin to proceed with the procedure.	
Test Procedure Valid Format Invalid Format		**Output** User added Error Message

Table 14 Add User Testing

7.2.3 Add New Hierarchy Testing

Test Case Name	Add new hierarchy
Test Case No	3
Description	Testing to add new hierarchy to the system
Testing Technique Used	Integration Testing
Preconditions	The system should be connected to the internet. The user should be logged into the system
Input Values	Mouse Clicks

Valid Inputs	--
Steps	1. Log into the system as admin 2. Click on Admin Panel 3. Select Add hierarchy from the drop-down menu
Expected Output	New Hierarchy should be added to the system
Actual Output	New hierarchy is added to the system

Table 15 Add New Hierarchy Table

7.2.4 Modify User Testing

Test Case Name	Modify User
Test Case No	4
Description	Testing to modify user.
Testing Technique Used	Integration testing
Preconditions	The system should be connected to the internet. The admin should be logged into the system
Input Values	Mouse Click Numeric Values
Valid Inputs	Phone number format specified
Steps	1. Log into the system as admin 2. Click on Admin Panel 3. Select Modify User from the drop-down menu
Expected Output	The User info should be modified
Actual Output	The user info is modified

Table 16 Modify User Testing

Chapter 8

8 Conclusion and Future Work

8.1 Conclusion

Our team set out to develop a system that would find an innovative and creative solution to a problem we face daily in our institution regarding the leave application generation and processing. We did so by creating a system that enables the users to generate applications in a formalized manner, while enabling the concerned authorities to receive these applications and respond to them in a systematic and organized way.

We achieved our objectives, successfully developing a Web Application that enables users to generate applications wherever they are, whenever they want as long as they have a data connection, alongside a web interface that enables concerned authorities to receive these applications and respond to them accordingly while updating the progress for the user to view.

Due to limitations in time and team size, the scope of the project was kept small. However, in the future the Project can be expanded to be useful in multiple institutions by making it generic rather than particular for the Military College of Signals.

If used correctly and adopted sincerely by the relevant authorities, we believe our project could genuinely benefit the institution.

8.2 Future work

Due to certain inherent limitations in terms of project development time and team size, a lot of things had to be excluded from the scope of this project. However, this leaves room for a multitude of enhancements, expansions and functionality add-on’s.

First of all, at the moment the Project only caters for the Military College of Signals, therefore the user base for the system will be extremely limited. In the future however, functionality could be expanded to include all major institutions of Pakistan.

At the moment, the system only caters for limited categories i.e. leave application generation and processing. In the future, this could be extended to other fields, such as administration and campus management etc. The program might be expanded to entertain all the campus related problems.

Among minor changes, the application's User Interface could be modified to be even more user friendly and the application could be optimized to run faster and improve performance on lower end devices. Functionality could be added to allow for prioritization of applications, or filtration and merging of similar applications.

Glossary

API	Application Programming Interface
App	Application
AS	Assumption
Black box Testing	Testing emphasizes on the external behavior of the software entity
CO	Constraints
LVE	Leave
DBMS	Database Management System
DEP	Dependency
FRs	Functional Requirements

GUI	Graphical User Interface
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol (HTTP) is a widely-used communications protocol for communication over a computer network, with especially wide deployment on the Internet
IDE	Integrated Development Environment
iOS	Mobile Operating System created and developed by Apple
JavaScript	Client side scripting language used to create dynamic web content and user interface
LRAS	Leave Recommendation and Approval System
MCS	Military College of Signals
NFRs	Non-Functional Requirements
NUST	National University of Science and Technology
OE	Operating Environment
OS	Operating System
Parse	Cloud Server
REQ	Requirement
SQL	Structured Query Language
SE	Security Requirements
SR	Safety Requirements
SRS	Software Requirements Specification

UD	User Documentation
UML	Unified Modeling Language
C/LVE	Casual Leave
P/LVE	Paid Leave
White Box Testing	Testing emphasizes on the internal behavior of the software entity

Table 17 Glossary

Bibliography

https://www.coursera.org/

https://bytehost.com

https://www.w3schools.com

https://www.ldeo.columbia.edu/~martins/sen_sem/thesis_org.html

Wikipedia (2015), “Cloud computing”, available from:

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