TECHNICAL REPORT

ON

HUMAN RESOURCE MANAGEMENT SYSTEM

Members and Contributors

Amit Prajapati
B00781164
m663039@dal.ca

Deeksha Bahara
B00784704
dk488492@dal.ca

Delisia Philip
B00782029
dl280077@dal.ca

Laxmi Yadav
B00762453
lx405836@dal.ca

Submitted To
Kirstie Hawkey and Robert Hawkey

CSCI 5308 - Quality Assurance

Faculty of Computer Science
Dalhousie University

August 03, 2018

ABSTRACT

This report explains the development process for Human Resource Management Portal. It contains information regarding what features the application contains and the process used to implement those. The report further explains the design patterns used for development, refactoring process and the technical debts for the Human Resource Management Portal application.

CONTENTS

1. INTRODUCTION

1.1 Purpose

Human Resource Management portal is a web application used by the corporate sector to communicate information concerned with the safety, compensation, benefits of the human resources and the employees at the company. The application technically acts as an interface that brings the managers and the employees to a single platform to share, request, retrieve personal HR-related concerns. HRM portal also serves as a self-service portal through which the employees can apply for their leaves conveniently using this application and view/ update their information without any additional hassle.

1.2 Scope

The project delivers a web portal that will be useful for the different kind of users in the corporate sector. The three kinds of users identified in this project are Web Admin, HRs, and the Employees. These users share common functionalities of logging in, editing the profile, applying for leave and accessing information like their CTC details and salary details. The HR employees have additional functionalities of processing the applied leave and adding new employees to the portal. The website also supports configurable logic like modifying default password and adding roles. which can be changed only by Web Admin users.

1.3 Background

The project was developed in seven sprints by strictly following the Agile methodology. The seventh sprint was utilized to implement code refactoring changes in UI, Java and SQL code. Apart from this, based on the review comments given by the professor, enhancements were added to the project.

2. APPLICATION DETAILS

2.1 Users

There are three types of user in the application, and they are Web Administrator, Admin and Employee.

2.1.1 Web Administrator

Web Administrator is the system administrator who can add or remove employees, view or update application configurations. Web administrator can also change the password.

2.1.2 Admin

Admin user can be HR and account department members. They can add or remove employees, upload documents like salary slip and CTC for employees. They also perform normal employee operations like viewing or updating their profile, downloading salary slip and CTC, and change their password.

2.1.3 Employee

Employee user is the employee from any department of the organization. They have features like viewing or updating their profile, downloading salary slip and CTC, and change their password.

2.2 Features

This application has five major modules Login, Profile, Leave, Admin, and Account. The module-wise feature description is as follows.

2.2.1 Login

The login functionality along with change password and log out comes under this module. The login module helps in creating and managing user session. The login feature in this application would allow the user to use the application features. The login page is the starting point of this application. The user will be authenticated using employee id and password. Different types of users like Web admin, Admin and Employees will use the same login screen but they will be redirected to different home pages on successful login. The functionalities which are available to them are controlled by using access control.

The change password feature will allow the user to change their password. The password will only be changed if the current password is known.Throughout the session, user will have an option to logout of the application. The logout operation will terminate the session and user will be redirected to the login page.

2.2.2 Profile Module

The profile module manages the profile related activities. Users can view their profile information using “View/Edit profile” option present in the profile tab. A profile page will be displayed with all the information regarding the employee. User can update certain parts of information like alternate email id, address and phone number. Phone number and email address will be validated before it is updated. If there are any errors while updating the information, then the update will not be saved and the user will be notified. If the new information entered by the user is valid then the data will be updated and the user will be informed about the same.

.

2.2.3 Leave Module

The features related to applying for a leave request and responding to a leave request comes under the leave module. Employees can apply for leaves and view their leave status using "Apply Leave" and "View Leave Status" options present in Leave tab. To help users in the decision-making process during leave application, the users will be shown the balance leaves available to him/her for that year. Another significant feature made available to the user is to view the status of the applied leaves. The users should be able to see a list of leaves they have applied in the past including approved and rejected leaves.

Admin have an additional feature in their portal to approve and reject the leaves applied by other employees. They can do that using "Process leave" option provided in the Leave tab.

2.2.4 Admin Module

The admin (HR) related operations like adding and removing employee comes under this module. Admin can add employees into the system using "Add Employee" feature. To add an employee, admin need to provide employee details like employee id, name, role, email id, alternate email id, phone number and address. The information will be validated before forwarding it to the server for processing. If any information is incomplete or incorrect, then the user will be notified.

Admin can remove employees from the system using "Remove Employee" feature. For removing any employee, admin need the Employee Id. Upon entering the Employee Id, admin will be able to view employee details and remove employee from the system. When an employee will be removed from the system, the saved data for that employee like salary slips, and credentials will be removed from the Employee info and Employee Login tables.

2.2.5 Account Module

This module includes upload and download document features. Admin can upload employee documents like salary slip and CTC. Employees will be able to download their salary slips and CTC documents using this module.

2.3 Application Workflow

The following figure shows the actions user can perform on the home page.

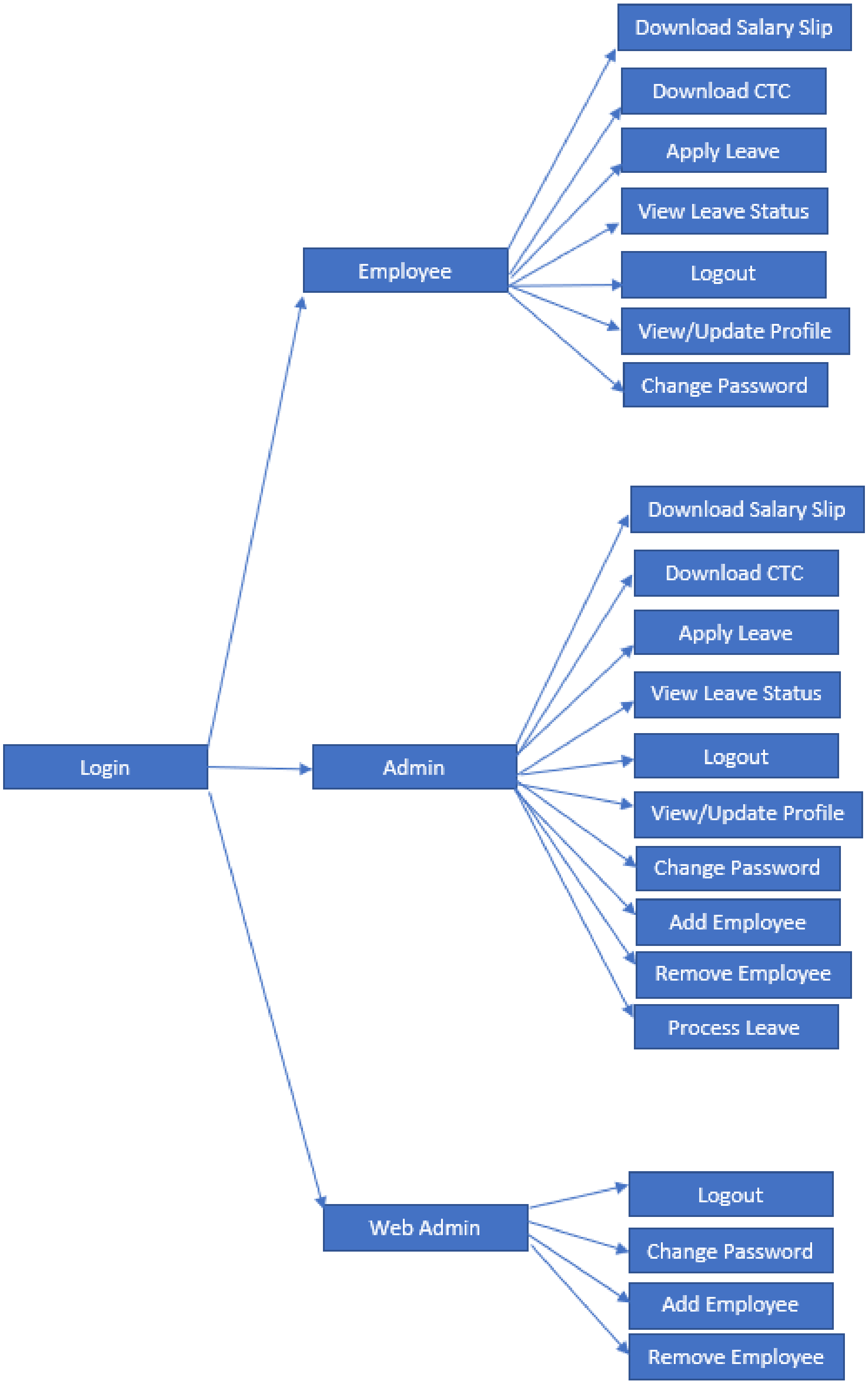

Figure 1: Sitemap

3. APPLICATION DEVELOPMENT PROCESS

We followed the agile methodology to develop the application and the application was implemented in seven sprints. Initially, we started off with the project by setting up the environment. Our work breakdown structure was divided as follows.

Sprints	Modules-Front End Development	Modules-Back End Development
Sprint 1	• Setting up the Environment	• Setting up the Environment. • Table Creations
Sprint 2	• Add Employee • Login Page • Remove Employee	• Add Employee • Remove Employee • Login • Logout
Sprint 3	• UI Template • Profile • Upload Salary Slip	• View Profile • Update Profile • Upload Salary Slip
Sprint 4	• Download Salary Slip and CTC • Apply Leave	• Download Salary Slip & CTC • Apply Leave
Sprint 5	• Process Leave • Role Configuration	• Stored Procedures • Embedding Stored Procedures • Role Configuration • Process Leave
Sprint 6	• Change Password • View Leave • Configurable Logic	• Change Password • Session Handling • View Leave Status • Exception Handling
Sprint 7	• Refactoring	• Refactoring • Configurable Logic

3.1 Continuous Integration

We used Heroku for continuous integration. Heroku account was set up followed by adding the contributors. There were three applications launched from Heroku which was linked to the three corresponding branches of the Dalhousie-Group1 GitHub branches which are development, test and production-release. Any code changes/ merges were deployed automatically to Heroku in the respective applications. Email notification was enabled from each of the branches to keep a track of the failed and succeeded deployments. The configurations for switching between the environments were not hardcoded and were read dynamically from a property file.

3.2 Project Management Tools.

GitHub and Trello were the project Management tools used for this project.

1. GitHub brought the entire team together onto a single platform. Each of us had individual feature branches where our work was done primarily and merged to develop. After every retrospective meeting and code reviews, the work was merged to develop branch. After every Fortnight, the code was released.
2. Trello was used as the project management tool to track the status of the user story. It helped people to communicate with each other and keep a track of the status of a module which is worked on by multiple people.

3.3 Used Conventions

During the initial project meetings, it was decided that we followed the following convention:

- Class names begin with capital letters
- Method names followed camel convention.
- Constants declared static and capitalized throughout
- Names of the packages is all small.
- Names of the Test Classes is the same as the original class names with the suffix “Test” followed.
- Using Tabs for indentation.

3.4 Layers in Application

We followed the `Model-View-Controller` architecture in developing our application. The model classes are related to the tables in the database and follows the same structures as the tables. The controller class acts as an interface between the model and the view and finally the view is the presentation layer which consists of all the user interfaces.

3.5 Configurable Logic

As discussed in the previous section, there are three kind of users for the HRM application namely –Employee, HR and Admin. HR and Admin are authorized to add or remove users of the application. The users are given a default password when they are added. The configuration logic built into the system allows the ‘admin’ user to change the default password assigned to the created users using API calls to the backend. The created user will be able to login to the system using the modified password without deploying the application again.

3.6 Technologies Used

The technologies mentioned below were used to build the application.

Front-end Languages: HTML, CSS, JavaScript, jQuery.

Front-end Framework: Bootstrap.

Back End Technologies: Java.

Framework: JAVA SPRING.

4. APPLICATION IMPLEMENTATION

4.1 Design Patterns

Design patterns help in creating highly cohesive modules. This application implements the following design patterns.

4.1.1 Singleton

Singleton pattern helps in creating classes which do not need to be initialized every time for using. This application implements the Singleton pattern for several classes like DatabaseAccessorFactory, DatabaseConnector, and DatabaseProperties.

4.1.2 Builder

We are using builder pattern for creating model instances for database and the database accessor classes which handle the database operations. The ModelFactory class handles the model construction and the DatabaseAccessorFactory handles the accessor class creation.

4.1.3 Strategy

Different types of users have different access in the application. This application uses the strategy pattern for implementing this feature. The application has three types of user, Employee, Admin, and Web Administrator and strategies for each user. The strategy classes decide whether to return the page that the user has requested and if not then where to redirect the user.

4.1.4 Decorator

Decorator Designer pattern allows an object to change its behavior dynamically by incorporating additional logic added to the original class.

In the project the throwable Exception class is decorated with features like status code and custom messages. This creates a throwable custom exception in the code rather than absorbing the exception within.

4.1.5 Chain of Responsibility

Chain of responsibility is a design pattern that simplifies object interconnections. In this pattern, the objects form a part of a chain and requests are sent from one object to another across the chain.

This pattern was implemented in HRM portal to fulfill the validation logic of the 'Leave module'. A series of validation logic which includes checking for a valid employee ID, various date validations, a check for validating the remaining leaves, and identifying overlapping leaves for an employee were implemented in a chain. The validation request is sent from one object to another across the chain until an object fails in the validation chain.

4.1.6 Template

Template design pattern defines a set of standards that must be followed by the sub classes which are implementing them. This pattern is suitable for objects which share similar functionalities.

The form obtained from the UI for employee CTC and employee salary slip follow similar yet different methods for validation. Template design pattern was implemented to validate the forms individually.

4.2 Refactoring

We started refactoring the code, once we had enough code. We refactored the database layer, where we were first using a class for getting the query and another class for executing it. We changed our approach and started using stored procedures with callable statements.

We also reconfigured the resource control logic and implemented a strategy pattern over there. Created three access strategies for the users and described which user can access which page. If the user tries to access a restricted page, then the user is redirected to homepage.

4.3 Technical Debts

We have finished all the features that we originally planned to implement. The module structure is not finalized and is a part of the technical debt. Our technical debt also includes checking the code if it has code smells and refactoring it. The user interface enables the user to perform operations, but it needs to be tweaked to look better. The last part of our technical debt includes changing alert into message boxes.

5. TEAM MEMBER CONTRIBUTIONS

5.1 Amit Prajapati

Amit implemented the following features:

- Maven project and defining the folder structure.
- The initial DatabaseConnector class for establishing connection to the database.
- Connected the application to the three environments on Heroku.
- Created the initial web controller for accessing html pages.
- Implemented logger class for logging in application.
- Implemented the SessionController for handling session related operations.
- Implemented the addEmployee and removeEmployee features in java.
- Implemented the logout feature in java.
- Implemented configurable logic java code for fetching and storing the configuration from database.
- Implemented builder pattern for creating model instances and accessor instances.
- Designed the view and update profile user interface.
- Designed the process leave user interface.

5.2 Deeksha Behara

- Implemented Employee Login Back end feature.
- Implemented Password Encryption for Login In
- Implemented Front end feature for Upload CTC
- Implemented Back end feature for Upload CTC
- Implemented Front end feature for Upload Salary Slip
- Implemented Back end Feature for Upload Salary Slip
- Designed View Leaves front end.
- Implemented View Leave Back end.
- Implemented Stored Procedures and embedding of stored procedures.
- Implemented Custom HRM Exception and decorator pattern to design exceptions.
- Implemented form validations for the form captured from the backend and used template pattern for form validations.

5.3 Delisia Philip

- Designed the template for HRM portal UI design.
- Created a wrapper for ajax calls.
- Designed the home page user interface.
- Designed the login module user interface with exception handling.
- Implemented exception handling for the user interfaces for all screens.
- Implemented leave application feature in java.
- Implemented process leave application logic in java.
- Implemented chain of responsibility design pattern for leave application.
- Designed the configurable logic user interface.
- Handled leave validations and exceptions in Java.
- Refactored UI implementations in several files.
- Partly designed the upload CTC user interface.

5.4 Laxmi Yadav

- Implemented backend logic for View Profile
- Implement backend logic for Update Profile Page
- Designed the UI for Change Password
- Implemented backend logic for Change password.
- Designed UI for download Salary Slip
- Designed UI for download CTC.
- Implemented backend logic for download CTC.
- Implemented backend logic for download salary slip.
- Implemented Configurable logic for Roles.
- Implemented strategy design pattern for Access Control.

6. CONCLUSION

The document illustrates the process followed for the implementation of the Human Resource Management portal. It includes the execution plan followed, features delivered by the team and the technical debts incurred.

The project helped us understand the best practices of using the Agile methodology. The test plan approach followed for the implementation of the project ensured that the enhancements we took further in the project did not hamper the original functionality of the project. The structured approach followed in developing the project also helped us deliver a quality product within the stipulated amount of time. With successful planning and management, we are herewith delivering the HRM portal with all the features implemented as mentioned in the project plan document as a deliverable for the Quality assurance course.