**Internal Organisation Management System**

***A PROJECT REPORT***

***Under the guidance of***

***Details***

******

*Submitted By*

**Prince Arora**

**1402002098**

******

*In partial fulfilment of the requirement for the award of the degree*

**Of**

**Master of Computer Applications**

**Department of Information Technology**

****

February, 2017

****

**BONAFIDE CERTIFICATE**

**Certified that this project report titled “**Internal Organisation Management System**” is the bonafide work of “**Prince Arora, 1402002098**” who carried out the project work under my supervision.**

**SIGNATURE**

**GUIDE**

**<<Name >>**

**Abstract**

**Companies are following process for all the new joining. One of them is familiarising new employee to their internal system, which takes lots of time and most of the time it gets very confusing for the employees. This document/ Project is designed to make this process easy and self-understandable.**

**Some of the feature provided:**

**Email Client:** Email client to check or compose internal emails.

**Chat Box:** Chat box to communicate with any employee in team and share files.

**Project management:** Project management to check project state and milestones.

**Employee management/ Team management:** Manage team/ employee.

**Assets management:** Manage office assets and assets assigned to any employee.

**Notifications:** Notifications for events and other actions

**Table of Content**

|  |  |  |
| --- | --- | --- |
| **Chapter No** | **Title** | **Page No** |
|  | **Abstract** | **1** |
|  | **List of Figures** | **2** |
|  | **List of Tables** | **3** |
| **1.** | INTRODUCTION   * 1. Introduction   1.1.1 Problem Statement  1.1.2 Resolution  1.1.3 Purposed System | **5** |
| **2.** | REQUIRMENT SPACIFICATIONS  2.1 SRS Diagram  2.2 E-R Diagram  2.3 Class Diagram  2.4 Data Flow Diagram |  |
| **3.** | SYSTEM DESIGN  3.1 Modularization  3.1.1 Dojo Components  3.1.2 REST Services Components  3.2 Data Integrity & Design  3.2.1 Data Security  3.2.2 Design  3.3 Procedural Design  3.4 User Interface Design |  |
| **4.** | IMPLEMENTATION  4.1 Dojo Implementation  4.2 REST Implementation |  |
| **5.** | RESULTS  5.1 Input and Output Screens |  |
| **6.** | Limitations and Future Application of the project |  |
|  | Conclusion and Bibliography |  |
|  |  |  |

INTRODUCTION

1.1 INTRODUCTION

1.1.1 Problem Statement

After hiring any new employee in any organisation, its organisation’s responsibility to give proper training to the employees to make them understand all the internal portals and processes that is needed. Most of the time it’s hard for employee to understand all at once because most of the time these processes are divided into different portals and are not divided according to the user’s rights.

Working on different portals to perform these action is time consuming and also hard to understand for new employees and takes to for them to familiarize with the system. If we have all the options at same place it will be self-explanatory for these employees and also they can manage their dashboard according to their convenience. For example a manager can keep the project management part option and can hide other sections and they can easily access those from same place when needed. If we keep these information at same place then all the employees will know what they need to work on and what is the status for their project and also other events and updates that are mandatory for users. And if they need to find something they can get it from this dashboard.

Most of the time organisations are dependent on mails or any chat system for their internal communications. And it gets confusing for employees if they have to search for anything important and they are not sure how it was communicated with them either mail or chat. If we can integrate these both at one place then it will be easy to them to search and also manage their meetings and schedules.

1.1.2 Resolution

Resolution for the above issue if to provide all these functionalities at one place and give appropriate rights to user so that can manage the dashboard and they are aware of their rights. All the updates and notifications can be found at same place and all the communications will be at same place. A search that can find things from all over their dashboard will be defiantly helpful for them as they don’t have to look for thing in different places where they are not sure where exactly they will find it.

If all the activities that are mandatory for an employee can be handled at one place then it will be easy for new employees to pick up these processes and will save lots of time to them. And Since it is a web application it can run anywhere.

1.1.3 Purposed System

The system that is purposed here is a single page application that will contain section/ Modules based on user’s permissions and roles. These modules can include Mail client, Chat Box, Project management, Asset management, Team Management and Notifications.

There are lots on web toolkits and frameworks to create single page application (AngularJS, ReactJS, ExtJS etc) one of them is Dojo Toolkit. Dojo provides modular ways to create a single page web applications. It provides some predefined components also that can help creating an application ex: fx to add animation and effect in html, tree to add tree structure in html page.

This single page application will be communicating with a REST services that will be doing all the persistence operations and the performing all the tasks based on business logic defined. These REST services will be communicating with Data storage (MySql).

* 1. Architectural Diagram.

**Browser with single page application**

Employee

**MySQL**

**DataBase**

**REST Services**

REQUIRMENT SPECIFICATIONS

2.1 SRS Diagram

Client 2

Client 3

Client 1

**MySql Data Store**

Dojo XHR request to use REST services

**Application Server containing REST Services (Apache Tomcat)**

**Web Server (Single Page Application hosted here)**

2.2 E-R Diagram

2.3 Class Diagram

2.4 Data Flow Diagram

Validate Username and password from DB

Send username and password to Services

**User Login**

Login user based on token generated.

Send token back to client browser/ web server.

Check user details and create token

Request for Login.

**Database**

**Application Server**

**Web Server**

User

**Common Data flow:**

Request for data

Convert and display JSON data as html.

Required data for user in JSON format

Query data from Database

Validate login token from request and fetch required data from database

Return data to application server.

Send user request with login token to REST services

**Database**

**Application Server**

**Web Server**

User