**Mini Project Case Study for Resource Management**

**Topic: Resource Management**

**Table of Contents**

**1 Requirement Specification**

**2 Design**

**2.1.High Level Design**

**2.2Business Flow**

**3 Coding**

**3.1Error Handling**

**3.2Environment**

* **Requirement Specification**

1.1  Background

First, we will be writing the node js program which will convert the excel to json format.

Next, we will be writing the test cases to test the API which we will be creating

We will create the api which will do 2 things. Return all json data, and return fitered data.

Once the api is created and tested, we will start with the UI.

React routers will be used to routing to redirect the user specific pages.

Once routing is configured we will be creating the individual pages and components.

We will simultaneously commit the code in gitbhub.

Once everything is done, we will deploy the application in cloud and provide a url where it can be accessed.

1.2 Functional Requirements

Description : Resource Management Application mainly used to display user information,

it consist of

1) Home page - The home page contains the entire information about the users.

2) Profile page - Profile page consist the information regarding a particular user.

1.3 Non Functional Requirements

* As much as possible, code and screen should be reused
* It should be compatible IE and Netscape browsers

The application should run without any issues in clustering environment. The team needs to document all the action items taken for the aspect. However, they don’t need to test in clustering environment

1.4 Purpose

The purpose of the project is to develop and deploy a web based system that is capable of displaying employees information.

* **Design**

The application is sole web based application and has the below high level architecture

**2.1. Web Application**

* **High Level Design**

Design an overall view of how the whole application should work and the top level components that will be made up of the proposed solution. It should have very little detail on implementation. In this stage we need to expand our requirements to “requirements and design” that includes a discussion of which design patterns we will apply to resolve the design forces identified in our requirements (If any new ones we notice it has to be added to that document during that high level design stage itself), and structure diagrams, interaction diagrams or other representations of the object-level structure of our design resulting from the application of those patterns. Based on the implementation design, we should also document a high level design for the tests we will use to evaluate our implementation, including interaction diagrams or any other representation suitable to capture that design as well.

***Architecture***

The overall architecture of the system is depicted here.

Over view of the architecture of the application:-

* Application UI has been built using HTML5, CSS, Bootstrap and React.js
* Node is used as server to handle the parsing the project and creating json file.
* Express is used for Routing purpose.
* Mocha & Chai is used for Testing.

* **Business Flow**
* **Coding**

* **Error Handling**

Appropriate error handling has been done in the project. All errors are routed to the pages which process the error , and display appropriate messages. The following errors are detected and error messages are displayed.

* A user cannot ‘copy paste’ the link (URL) of an old session and try to run it again.

* **Environment**

* The application should run on any node compliant web servers.
* Client Browser could be Internet Explorer, Mozilla Fire Fox or Chrome. However, application team should target only IE and Chrome for testing
* Use React.js framework for web application UI is also used.
* Node server will take the request from users.
* Atom editor can be used as an development environment
* Express can be used for Routing purpose.
* Mocha & Chai are used for Testing purpose.

**Development & Testing Environment**

Platforms

* Windows 8.1, Windows 10, Windows XP

Application Server

Node.

**System Requirements**

**Operating System**

* Windows: Windows 2000, XP, or Windows 8.1

**Hardware**

* Intel® Pentium® III 1.0 GHZ or higher recommended
* 512 RAM minimum