**Vibhaag – College Management, Scheduling and Analytics**

**Title: Vibhaag - Enterprise Management, Scheduling and Analytics**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl. No** | **Name** | **Role** | **Degree, Semester** | **College** |
| 1 | Sudhanva N | Project Lead | MCA, 4th Semester | PES University |
|  | **Phone**: 9632350260  **Email**: [nsudhanva@gmail.com](mailto:nsudhanva@gmail.com)  **SRN:** PES1201702260 |  |  |  |
| 2 | Karthik Devaraj | Developer | MCA, 4th Semester | PES University |
|  | **Phone**: 8660310176  **Email**: [karthikdevaraj1017@gmail.com](mailto:karthikdevaraj1017@gmail.com)  **SRN:** PES1201702449 |  |  |  |
| 3 | Shreedhar Hegde | Project Lead | MCA, 4th Semester | PES University |
|  | **Phone**: 8660452642  **Email**: [shreedharhegade96@gmail.com](mailto:shreedharhegade96@gmail.com)  **SRN:** PES1201702323 |  |  |  |
| 4 | Harsha K Y | Developer | MCA, 4th Semester | PES University |
|  | **Phone**: 8971677453  **Email**: [harshaky@gmail.com](mailto:harshaky@gmail.com)  **SRN:** PES1201801839 |  |  |  |
| 5 | Aadarsha V | Developer | MCA, 4th Semester | PES University |
|  | **Phone**: 8660310176  **Email**: [adarshav957@gmail.com](mailto:adarshav957@gmail.com)  **SRN:** PES1201702022 |  |  |  |

**Project Faculty in charge and Mentor:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No** | **Name** | **Designation** | **College** |
| 1 | Dr. Veena S | Chairperson, MCA | PES University |
|  | **Email**: [sveena@pes.edu](mailto:sveena@pes.edu) |  |  |

**VIBHAAG – COLLEGE MANAGEMENT, SCHEDULING AND ANALYTICS**

1. **Introduction**

**Purpose** - A Software as a service (SaaS) web application where in the idea is to allow enterprises such as schools, colleges, banks and other offices to manage vital day-to-day processes such as employee management, customer management, meeting and schedule management, analytics and reporting.

**Document conventions -**

DB - Database

DDB - Distributed Database

**Intended Audience and Reading Suggestions -** The project is currently under the development phase where it is being tested for accuracy and reliability. It will help every enterprise in making their day to day processes easier and accountable. Further plan is to deploy the product in any enterprise organization which strive for quality.

**Project Scope -** The purpose of this courier delivery system is to ease the college management and to create a convenient and ease to use application for administrators to manage and maintain their faculty. The system is based on a non-relational database with its analytics and notification functionalities.

**References -**

* GitHub
* Stack Overflow
* React JS, Redux documentation
* Node JS, Express JS, MongoDB documentation
* Udacity – React and React Native course

1. **Overall Description**

**Product Perspective -** The application consists of two main components. One is the web application and the other being the mobile application. The web application is responsible for authentication, user role management and database management whereas the mobile application is responsible for users’ attendance or presence via scan-to-confirm strategy. It provides customizable options for creating users with role management and secure authentication.

**User classes and characteristics -** The schedule of activities are recorded by the administrator at the beginning of the session. The mobile app is responsible for recording the actual events occurred when the session is held. It records the time, date and location of the incident by a QR Code and sends the data securely to the web application which in turn stores and processes the information. Reports can be generated based on processed information and made available to different users based on their roles. Whenever there is a change in schedule, it will be notified to the employees through the app.

The Admin should have the following management functionalities:

* User authorization and authentication with role management
* User should have restricted access, permission-based creation of departments, faculty or any other entity
* Customized schedules for sessions for different user or group along with notifications
* Create departments, sessions, add faculty, add subjects

Operating Environment

Operating environment for the delivery system is as listed below:

* Distributed database
* Client / Server system
* Operating system : Windows/Linux/MacOS
* Database : NoSQL
* Platform : Node.js

Design and Implementation Constraints

* The global schema, fragmentation schema and allocation schema.
* NoSQL commands for above queries/applications
* Implement the database at least using a centralized management system.

User Documentation:

As the product is CMS, user documentation becomes a critical component of the system as it shall provide specific guidelines to a user for using the system.

Assumptions and Dependencies

Let us assume that this is a deployed courier delivery system and it is used in the following application:

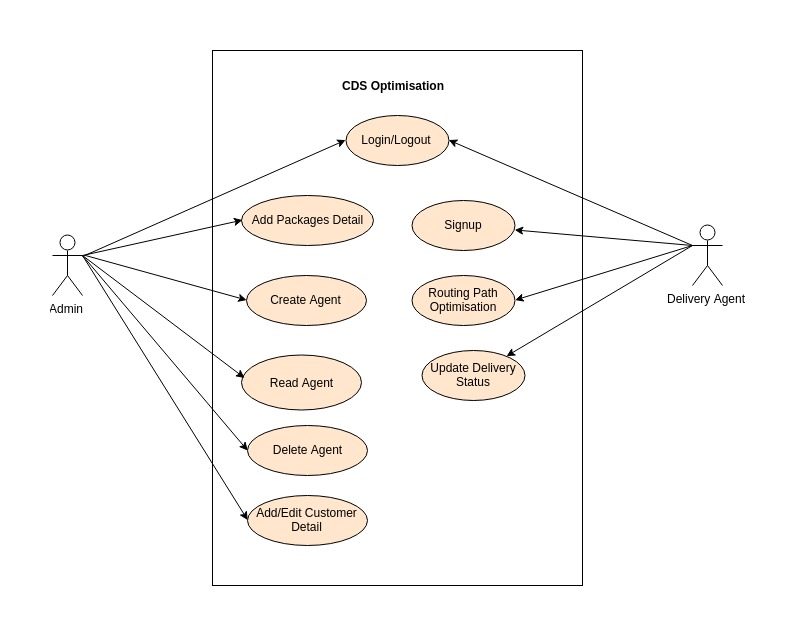
* Get real time traffic updates that will allow the courier delivery executive to deliver the package within a shortest span of time.
* Reroute the delivery route of the package to be delivered according to the traffic on route.

**3) System Features**

The courier delivery system maintains information about number of delivery executives and their details, packages to be delivered and their details and tracking the movement of package.

**Functional requirements**

Other system features include:



Client / Server System:

The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application - frontend (Here the client is the delivery executives) and the server is the DBMS (backend).

**4) External Interface Requirements**

User interfaces:

Front-end software: HTML, CSS and React js

Back-end software: MongoDB (NoSQL)

Hardware interfaces:

* Windows / Ubuntu 18.06
* Web or mobile browser which supports CGI, HTML and JavaScript

Software interfaces:

|  |  |
| --- | --- |
| **Software used** | **Description** |
| Operating System | We have chosen Windows Operating System for its best support and user-friendliness |
| Database | To save the delivery details and delivery agent details, we have chosen NOSQL(MongoDB) |
| JavaScript | To implement the project, we have chosen JavaScript language for its more interactive support |
| Heroku Cloud Deployment | The platform for deploying our app, we choose Heroku NodeJS Platform. |
| MLab MongoDB cloud Deployment | MLab MongoDB cloud service to store data in NoSQL database. |

Communication interfaces:

The project supports on Chrome, Firefox, Safari, and Opera.

**5) Other Nonfunctional Requirements**

Performance Requirements:

The steps involved to perform the implementation of courier delivery system are listed below

**E-R Diagram:**

The E-R Diagram constitutes a technique for representing the logical structure of a database in a pictorial manner. This analysis is then used to organize data as a relation, normalizing relation and finally obtaining a relation database.

ENTITIES: Which specify distinct real-world items in an application.

PROPERTIES / ATTRIBUTES: Which specify properties of an entity and relationships.

RELATIONSHIPS: Which connect entities and represent meaningful dependencies between them.

Safety Requirements:

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

Security Requirements:

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

Software Quality Attributes:

* AVAILABILITY: The courier should be delivered within the stipulated date and time.
* CORRECTNESS: The delivery executive should start from the correct place at the right time and reach the designated destination.
* MAINTAINABILITY: The admin must be able to maintain the correct schedules of the packages to be delivered.
* USABILITY: The delivery schedules should satisfy a maximum number of customers as the courier must be delivered according to the convenience of the person receiving it.