

Human Resource Management System (HRMS)

P. Vinay Bhushan¹, P. Durga Prasad², V. Sai Kiran³, K. Raja Shekar⁴

B. TECH Scholars, Department of Computer Science and Engineering^{1,3,4}

Assistant Professor, Department of Computer Science and Engineering²

Sreenidhi Institute of Science & Technology, Hyderabad, India - SNIST

Abstract: Human Resource Management System Project is an basically software designed to keep track of hand information in any company. It stores data similar as their workers' particular information. The thing of " Human Resource Management System" is to produce a work center scheduling system. Scheduling is a technology that makes the process of informing conditioning and announcements in the company where it's enforced simple and indeedonline. The Human Resource Management System design gives directors a better idea of their workers and helps them plan and manage their work hours in order to cut costs and boost productivity. It gives applicable directions and supervisions for workers. It also secures and manages information that are important to the workers including particular and work- related information. This design contains modules like Hand information operation, hand schedule monitoring, cover working days and leaves, set leave reporting and induce leave report.

Keywords: Effectiveness, Caregivers, Care, STP, Knowledge, Hemodialysis

I. INTRODUCTION

The Project is made using: MongoDB, Node.js, Express.js, React.js, Web Sockets.

1. MongoDB: MongoDB is a popular open-source NoSQL database that is designed for scalability, flexibility, and ease of development. It is a document-oriented database that stores data in JSON-like documents with dynamic schemas, making it ideal for handling unstructured or semi-structured data.
2. Node.js: Node.js is an open- source, cross-platform, garçon- side JavaScript runtime terrain that's erected on Chrome's V8 JavaScript machine. It's used to develop scalable, high- performance operations for web waiters, command-linetools, and native operations. Node.js was developed by Ryan Dahl in 2009, and it has since gained a massive following and relinquishment from inventors around the world.
3. Express.js: Express.js is a popular web framework for Node.js, designed to build robust and scalable web applications. It provides a set of features and tools to build web applications, including middleware support, routing, and templating engines. Express.js is open source, which means that developers can use it for free and customize it to fit their specific needs.
4. React.js: React.js is an open- source JavaScript library that's used to make stoner interfaces(UI). It was developed by Facebook in 2011 and was released to the public in 2013. React.js has gained popularity due to its simplicity, performance, and reusability. It provides a declarative approach to building UI components and allows developers to easily manage the state of the application.
5. Web Sockets: Web Sockets is a communication protocol that enables bidirectional communication between customer and garçon over a single TCP connection. It provides a patient, low- quiescence, full- duplex communication channel over a single socket, allowing real- time data to be transmitted between web cybersurfers and web waiters. Web Sockets is designed to work with any web garçon and is supported by all major web cybersurfers. In this report, we will give a brief overview of Web Sockets, including its history, advantages, and operation.

II. BACKGROUND STUDY

HRM emerged in the early 20th century as a response to the growing complexity of workforce management in large organizations. The need for HRM systems became more apparent as organizations grew in size and complexity, and the workforce became more diverse. HRM systems were initially developed to automate routine administrative tasks, such

as payroll processing and record-keeping. However, as HRM evolved, the scope of HRMS expanded to include other functions such as recruitment, training, performance management, and benefits administration. The concept of HRMS was further popularized with the advent of computer technology in the 1960s and 1970s. This allowed organizations to automate HR functions and manage employee data more efficiently. The development of the Internet and cloud computing in the 1990s and 2000s further transformed HRMS, allowing organizations to manage HR functions from anywhere in the world. The proposal of HRMS is driven by the need for organizations to manage their workforce more effectively and efficiently. As organizations face increasing competition and pressure to improve performance, HRMS can provide a strategic advantage by streamlining HR processes, reducing costs, and improving the quality of HR services.

III. EXISTING SYSTEM

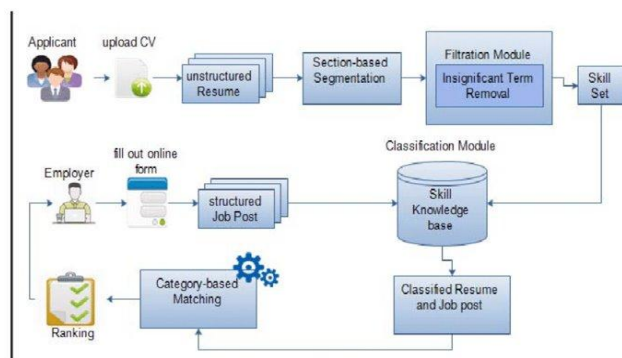
The existing system for human resource management can vary widely across organizations. Some organizations may use manual systems for managing their HR activities, while others may use standalone software solutions for specific HR functions such as recruitment or performance management. The existing system may also be a combination of different software solutions that are not integrated with each other. These systems can be inefficient, time-consuming, and error-prone, leading to problems such as duplicate data entry, data inconsistencies, and difficulties in generating reports.

IV PROPOSED SYSTEM

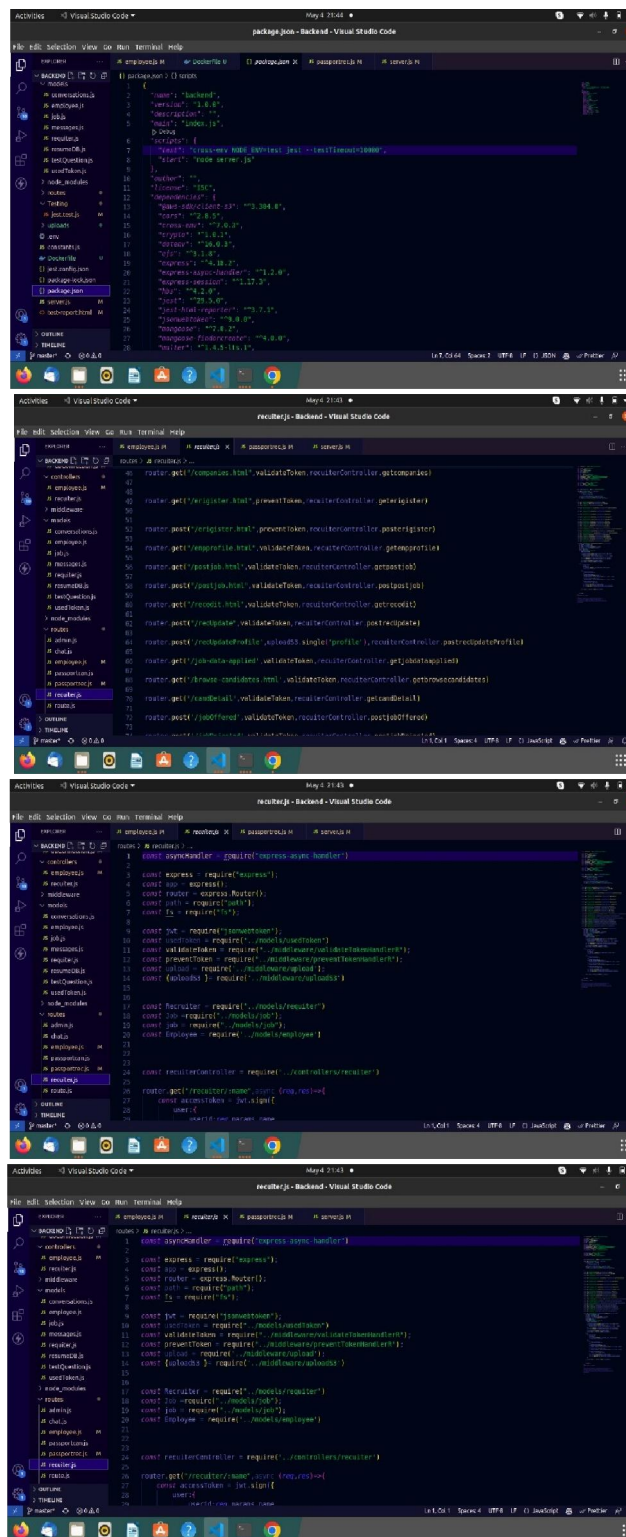
The proposed system for human resource management is a comprehensive software solution that integrates all HR functions into a single platform. It aims to automate and streamline HR processes, reduce administrative burden, and provide real-time data insights to help make informed decisions. The proposed system will include modules for various HR functions, including recruitment, onboarding, performance management, payroll, and employee self-service. The system will also provide data analytics and reporting capabilities to enable managers to make data-driven decisions. The proposed system will be a web-based application that can be accessed from anywhere, at any time, using any device with an internet connection. It will be scalable and flexible, allowing organizations to customize it to meet their specific needs. The system will be designed with data privacy and security in mind, and it will comply with local legal and regulatory frameworks.

VI ARCHITECTURE

Systems design could be seen as the operation of systems proposition to product development. There's some imbrication with the disciplines of systems analysis, systems armature and systems engineering. Architectural Design The architectural design of a system emphasizes the design of the systems armature that describes the structure, geste and further views of that system and analysis. This is frequently conducted via modelling, using anover-abstract(and occasionally graphical) model of the factual system. In the environment of systems, designs are included. Physical Design The physical design relates to the factual input and affair processes of the system. This is explained in terms of how data is input into a system, how it's vindicated authenticated, how it's reused, and how it's displayed. In physical design, the ensuing conditions about the system are decided.



VII IMPLEMENTATION



```

// package.json - Backend - Visual Studio Code
{
  "name": "backend",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "start": "node server.js"
  },
  "dependencies": {
    "express": "^4.18.2",
    "mongoose": "^6.9.0",
    "cors": "^2.8.5",
    "dotenv": "^16.0.3",
    "jsonwebtoken": "^8.5.1"
  }
}

// routes.js - Backend - Visual Studio Code
const express = require('express');
const router = express.Router();
const jwt = require('jsonwebtoken');

const {
  register,
  login,
  validateToken,
  getAllEmployees,
  getEmployeeById,
  updateEmployee,
  deleteEmployee,
  assignDepartment,
  updateEmployeeStatus
} = require('../controllers/employees');

router.post('/register.html', validateToken, register);
router.post('/login.html', validateToken, login);
router.get('/employees.html', validateToken, getAllEmployees);
router.get('/employees/:id', validateToken, getEmployeeById);
router.put('/employees/:id', validateToken, updateEmployee);
router.delete('/employees/:id', validateToken, deleteEmployee);
router.put('/employees/:id/department', validateToken, assignDepartment);
router.put('/employees/:id/status', validateToken, updateEmployeeStatus);

module.exports = router;

// app.js - Backend - Visual Studio Code
const express = require('express');
const bodyParser = require('body-parser');
const mongoose = require('mongoose');
const cors = require('cors');

const app = express();

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));
app.use(cors());

const config = require('../config');
const { connect } = require('../db');
const { create } = require('../models');
const { routes } = require('../routes');

connect();
create();

app.use('/', routes);

const PORT = config.port;
app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});

// routes.js - Backend - Visual Studio Code
const express = require('express');
const router = express.Router();
const jwt = require('jsonwebtoken');

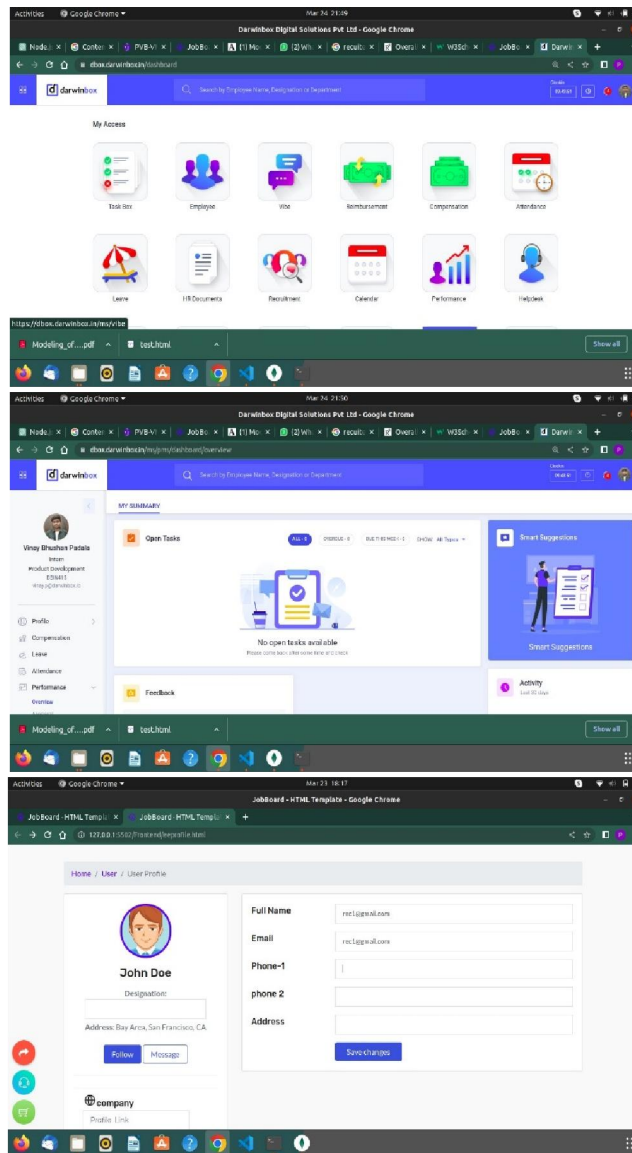
const {
  register,
  login,
  validateToken,
  getAllEmployees,
  getEmployeeById,
  updateEmployee,
  deleteEmployee,
  assignDepartment,
  updateEmployeeStatus
} = require('../controllers/employees');

router.post('/register.html', validateToken, register);
router.post('/login.html', validateToken, login);
router.get('/employees.html', validateToken, getAllEmployees);
router.get('/employees/:id', validateToken, getEmployeeById);
router.put('/employees/:id', validateToken, updateEmployee);
router.delete('/employees/:id', validateToken, deleteEmployee);
router.put('/employees/:id/department', validateToken, assignDepartment);
router.put('/employees/:id/status', validateToken, updateEmployeeStatus);

module.exports = router;

```

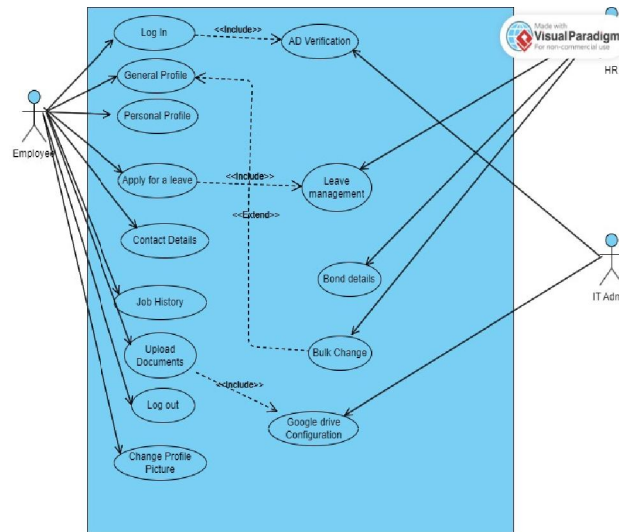
VIII. OUTPUT SCREENS



IX UML DIAGRAMS

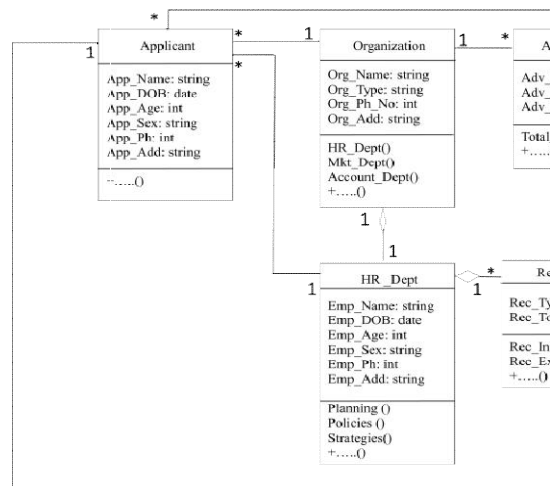
Usecase Diagram

A use case illustration is used to represent the characteristics of your system's druggies and how they interact with it. Actors in a use case are system druggies. We can produce one by employing a set of symbols. An excellent use case demonstrates to the platoon how individualities or organizations interact with software or operations. With the help of your system or operation, assists actors in achieving the compass of your system. A use case illustration doesn't go into great depth; for illustration, it doesn't show the order in which way are performed. A proper use case illustration depicts the commerce between use cases and actors.



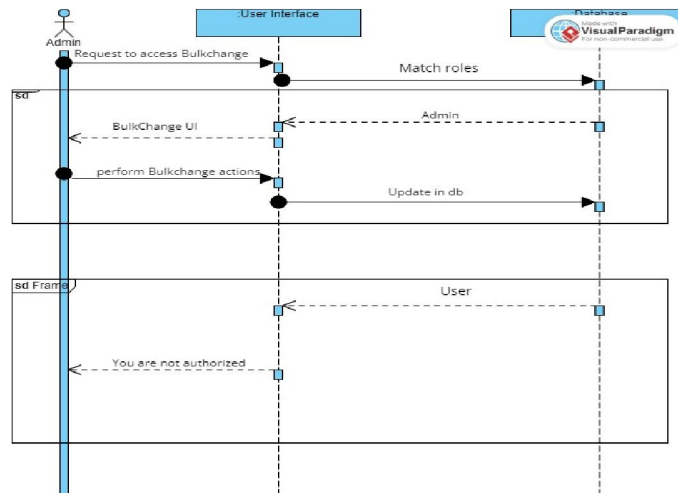
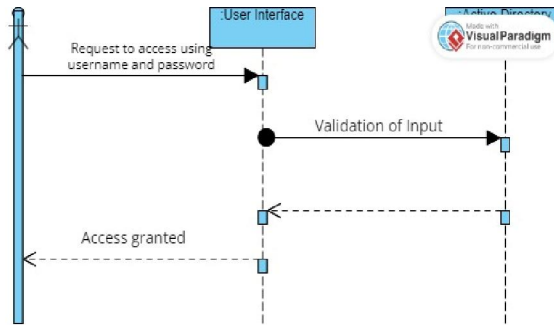
Class Diagram

A class illustration is a type of structural illustration used in the field of software engineering and object- acquainted design to visually represent the connections and association of classes within a system. As a core element of the Unified Modeling Language(UML), class plates serve as a design for the development of software operations, furnishing a clear and terse view of the system's armature.



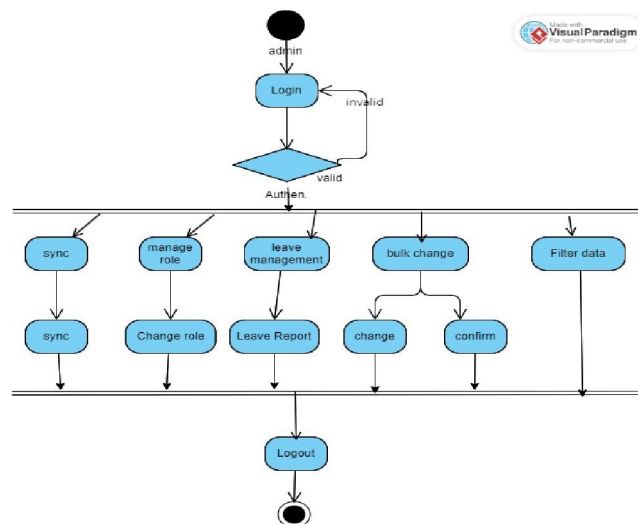
Sequence Diagram

The sequence illustration depicts the relations of particulars or rudiments in a logical order. This illustration is constantly employed by software masterminds and business experts to understand the conditions of a system. This illustration can be used to validate the procedure as well. They're also appertained to as event plates. This illustration depicts use cases in great detail. The quantum of time it takes an object to fulfil its duty is represented by activation. The dashed line under an object indicates the thing's successional tasks.



Activity Diagram

The inflow of conditioning in the system is depicted by an exertion illustration. An exertion illustration is a graphical representation of what causes a particular event. In UML, there are three feathers of plates structure plates, geste plates, and commerce plates



X CONCLUSION

Based on the information presented in your project report on the HRMS (Human Resource Management System), it can be concluded that the project is a comprehensive solution that streamlines HR-related activities, from recruitment to employee management and payroll processing. The system is designed to improve the efficiency and effectiveness of

HR operations, reducing the workload for HR staff while providing employees with a user-friendly interface for accessing their personal information, benefits, and company policies.

Overall, the HRMS is a valuable tool for any organization seeking to optimize its HR operations and increase the satisfaction and engagement of its workforce. Your project report provides a thorough analysis of the system's features, benefits, and potential challenges, highlighting the importance of careful planning and effective implementation to ensure a successful outcome.

XI FUTURE SCOPE

The HRMS project has immense potential for future expansion and development. As businesses and organizations continue to evolve and embrace technology, the demand for advanced HR solutions will increase. Therefore, the HRMS project can be further enhanced to include additional features and functionalities.

One potential area of future scope for the HRMS project is the integration of artificial intelligence and machine learning algorithms. By incorporating these technologies, the system can become even more intelligent and capable of predicting employee behavior, identifying trends, and providing personalized insights and recommendations.

Another area of future scope for the HRMS project is the integration of blockchain technology. This can enhance the security and privacy of employee data, and enable more efficient and secure payment processing.

In addition, the HRMS project can be expanded to include modules for employee engagement, performance management, and talent development. These modules can provide a comprehensive framework for managing and developing employees, ensuring that they have the necessary skills and competencies to meet business needs

REFERENCES

- [1]. Ulrich, D., & Brockbank, W. (2005). *The HR value proposition*. Harvard Business Press.
- [2]. Martinsons, M. G., Davison, R. M., & Tse, D. K. (2014). The balanced scorecard: A foundation for the strategic management of information systems. *Decision Support Systems*, 71, 1-14.
- [3]. SHRM. (2018). SHRM HR jobs. Retrieved from <https://www.shrm.org/hr-jobs/Pages/default.aspx>
- [4]. Bureau of Labor Statistics. (2018). Occupational outlook handbook. Retrieved from <https://www.bls.gov/ooh/>
- [5]. HR Technologist. (2019). HR software. Retrieved from <https://www.hrtechnologist.com/hr-software/>
- [6]. Bureau of Labor Statistics. (2018). Occupational outlook handbook. Retrieved from <https://www.bls.gov/ooh/>
- [7]. Stone, D. L., & Deadrick, D. L. (2015). Challenges and opportunities affecting the future of human resource management. *Human Resource Management Review*, 25(2), 139-145.
- [8]. Jackson, S. E., Schuler, R. S., & Jiang, K. (2014). An aspirational framework for strategic human resource management. *Academy of Management Annals*, 8(1), 1-56.