MAPÚA MALAYAN DIGITAL COLLEGE

A college under Mapúa Malayan Colleges Laguna

MotorPH OneSuite System Documentation

Prepared and Presented by:

Lopez, Sherilou
Lao, Winston Ace
Bansas, Rhona Lynne
Bachelor of Science in information Technology
2nd Term, S.Y. 2023-2024

TABLE OF CONTENTS

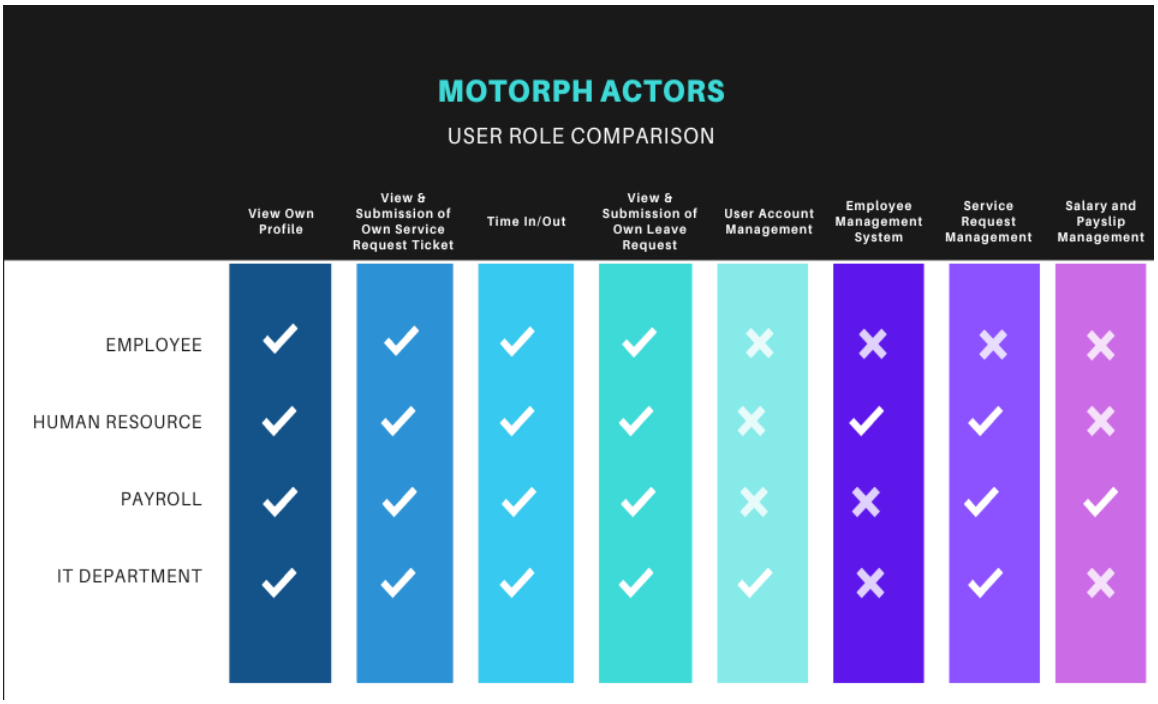
1. Introduction

The MotorPH OneSuite stands as a pinnacle of modern software engineering, meticulously engineered to revolutionize and simplify the intricate landscape of payroll management within any organization. With its robust features and intuitive design, this comprehensive solution serves as a beacon of efficiency, streamlining and automating every aspect of the payroll process. From centralized employee information management to seamless service request handling, meticulous time and attendance tracking, efficient leave and absence management, precise salary calculations, meticulous deduction management, and comprehensive withholding tax processing, MotorPH OneSuite is a one-stop solution for all payroll-related needs. Moreover, it serves as a secure repository for payslips, ensuring that every piece of vital information remains easily accessible yet impeccably safeguarded. By employing this system, MotorPH empowers itself to uphold impeccable data accuracy, navigate regulatory complexities with ease, and foster an ecosystem of unparalleled efficiency and compliance throughout the organization.

This user guide represents an indispensable resource for users delving into the intricacies of the MotorPH OneSuite. Crafted with precision and clarity, it aims to arm users with the knowledge and skills necessary to navigate the system adeptly, extracting maximum value from its myriad functionalities. Whether it's understanding the nuances of data entry, exploring the depths of reporting capabilities, or mastering the art of customizing settings to align with organizational needs, this guide provides comprehensive coverage. By empowering users with in-depth insights and practical guidance, it ensures that the MotorPH OneSuite becomes not just a tool but a strategic asset, driving efficiency, compliance, and success across the organization's payroll management endeavors.

1.1. User Roles

In the MotorPH OneSuite, four primary user roles interact with the software to facilitate efficient payroll management: employees, HR personnel, payroll administrators, and IT administrators. Employees utilize the system to access their personal information, submit service requests, and view payslips. HR personnel oversee employee data management, process leave requests, and generate reports for organizational analysis. Payroll administrators handle salary calculations, deductions, and tax management, ensuring accurate payroll processing. IT administrators maintain system integrity, handle user access permissions, and provide technical support to ensure smooth system operation. Each user role plays a vital role in the seamless functioning of the MotorPH OneSuite, contributing to streamlined payroll processes and organizational efficiency.



1.2. Navigating The User Guide

2. Getting Started

2.1. Hardware Prerequisites

Minimum System Requirements:

- CPU: Dual-core processor (e.g., Intel Core i3 or AMD equivalent)
- RAM: 4 GB DDR3 RAM
- Storage: 50 GB available hard disk space
- Internet Connectivity: Required for downloading and installing software dependencies, updates, and accessing online resources if necessary.
- Operating System:
    - Windows 7 or newer
    - macOS 10.12 Sierra or newer
    - Linux (Ubuntu 16.04 or equivalent)

Recommended System Requirements:

- CPU: Quad-core processor (e.g., Intel Core i5 or AMD equivalent)
- RAM: 8 GB DDR4 RAM
- Storage: 100 GB available hard disk space (SSD recommended for better performance)
- Internet Connectivity: Required for optimal performance, including downloading updates, accessing online resources, and community support.
- Operating System:
    - Windows 10 or newer
    - macOS 10.14 Mojave or newer
    - Linux (Ubuntu 20.04 LTS or equivalent)

2.2. Software Prerequisites

1. Visual Studio Code:

- Description: Visual Studio Code is a source code editor developed by Microsoft for Windows, Linux, and macOS.
- Version: Ensure you have the latest stable version installed.
- Download Link: **Visual Studio Code**

2. Node.js:

- Description: Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. It allows you to run JavaScript on the server-side.
- Version: Ensure you have the latest LTS (Long Term Support) version installed.
- Download Link: **Node.js**

3. Nodemon:

- Description: Nodemon is a utility that monitors for any changes in your source code and automatically restarts your server.
- Installation: After Node.js is installed, run the following command in your terminal:

```
npm install -g nodemon
```

4. Express:

- Description: Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.
- Installation: After Node.js is installed, you can install Express in your project by running the following command in your terminal within your project directory:

```
npm install express
```

5. NPM Packages:

- csv-parser: A streaming CSV parser that aims for maximum speed and minimal memory usage.
- csv-write: A CSV serializer for Node.js.
- cors: Cross-Origin Resource Sharing middleware for Express.js.
- Installation: After Node.js is installed, run the following commands in your terminal:

```
npm install csv-parser csv-write cors
```

6. Internet Browser:

- Description: Any modern internet browser such as Chrome, Safari, Edge, Firefox, etc.
- Version: Ensure you have the latest stable version installed.
- Download Links:
  - Google Chrome
  - Safari
  - Microsoft Edge
  - Mozilla Firefox

2.3. Accessing The System

1. **Launch Visual Studio Code:**
    - Locate and open the Visual Studio Code application on your computer. You can typically find it in your applications folder (on macOS) or in the Start menu (on Windows).

2. **Download the project folder:**
    - Visit the GitHub repository link: https://github.com/winsace/OOP/tree/main/motorph.
    - Navigate to the src folder by clicking on it.
    - Once you're inside the src folder, you should see the server.js file along with other files and folders.
    - To download the entire src folder, click on the green "Code" button located towards the right side of the screen.
    - In the dropdown menu, click on "Download ZIP". This will download the entire repository, including the src folder and its contents, as a ZIP file to your computer.
    - After the ZIP file is downloaded, locate it in your downloads folder or the directory where your browser saves downloads.
    - Extract the contents of the ZIP file. Inside, you'll find the src folder containing the necessary files, including server.js.

3. **Open the project folder:**
    - Within Visual Studio Code, navigate to the folder containing your Node.js project files. You can either use the "File" menu to open a folder or simply drag and drop the folder into Visual Studio Code's interface.

4. **Run the Node.js application with Nodemon:**
    - Open the integrated terminal in Visual Studio Code. You can do this by selecting "Terminal" from the top menu and choosing "New Terminal".
    - Navigate to the project directory within the terminal. You can use the cd command followed by the path to your project directory.
    - Instead of using the node command to run your Node.js application, use nodemon followed by the main script name. For example:

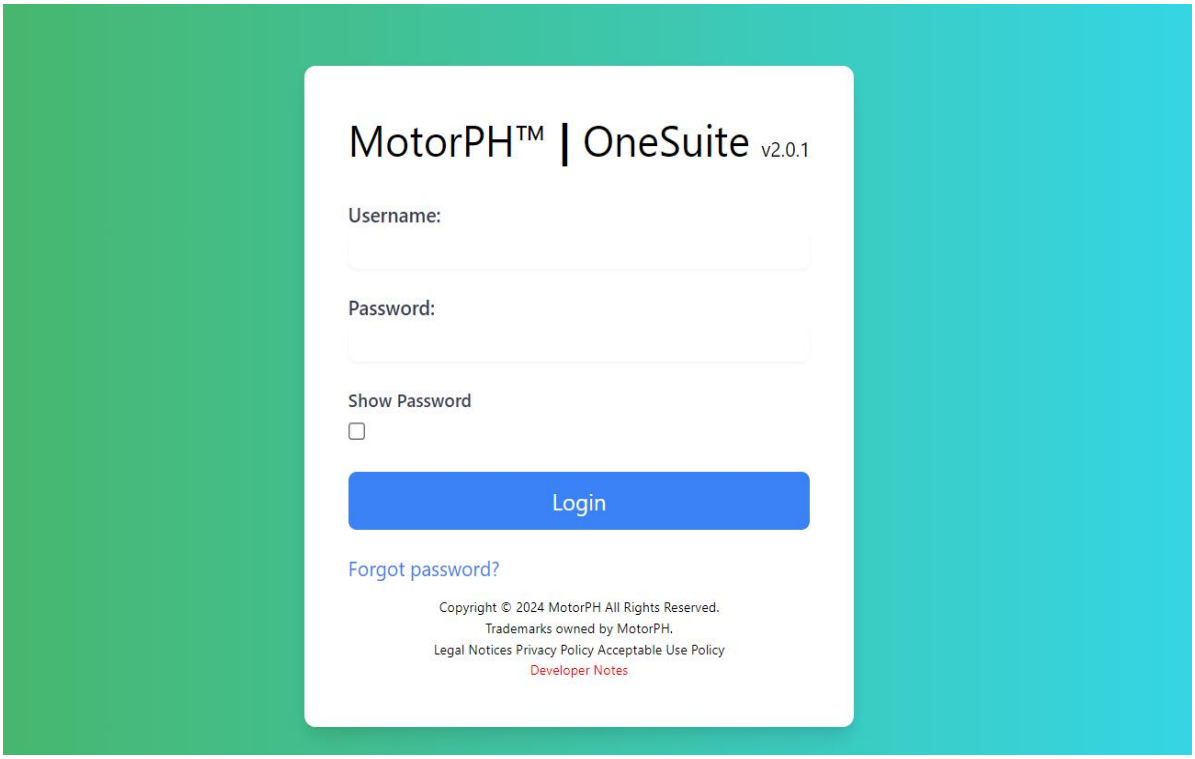
```
nodemon server.js
```

    - Replace server.js with the name of your main Node.js script file.
    - Nodemon will start the application and monitor for any changes in your code. Whenever you save changes to the code, Nodemon will automatically restart the application, making the development process smoother.

5. **Server Launch Confirmation:**
    - After running the command, you'll see the output message "Server running at http://localhost:3000" in the terminal, indicating that the server is up and running. You can then access the application by opening a web browser and navigating to http://localhost:3000. This URL corresponds to your local machine and the port on which the server is running. For example:

```
[nodemon] starting `node server.js`
Server running at http://localhost:3000
```

6. **Accessing the Application via Web Browser:**
    - Open any web browser on your computer.
    - In the address bar of the web browser, type http://localhost:3000 and press Enter.
    - This will open the application in the web browser, as the server is running locally on port 3000.
    - You should now see the homepage or landing page of MotorPH OneSuite, as served by the Node.js server.



3. Using The Payroll System

3.1. Secure Login Procedures

All user roles (Employees, HR, Payroll, and IT) would need a username and password to access the system.

- **Login Steps:**
    1. Launch the web interface.
    2. A login screen will appear prompting for username and password.
    3. Enter credentials and submit the login request.

- **Login Response:** Upon successful login with valid credentials, the system would grant access to the user interface corresponding to their assigned role (e.g., employee dashboard, HR dashboard, Payroll dashboard, or IT dashboard). If the username and password combination is incorrect, the system would display an error message indicating invalid login credentials.

- **Forgot Password:** The system offers a "forgot password" functionality. This involves a process to reset the password, through an email verification.

3.2. Salary and Deduction Calculations

<Depending on the scope of your project, provide instructions for calculating salaries. This can be as simple as a basic salary calculation or a more complex computation that considers various factors.>

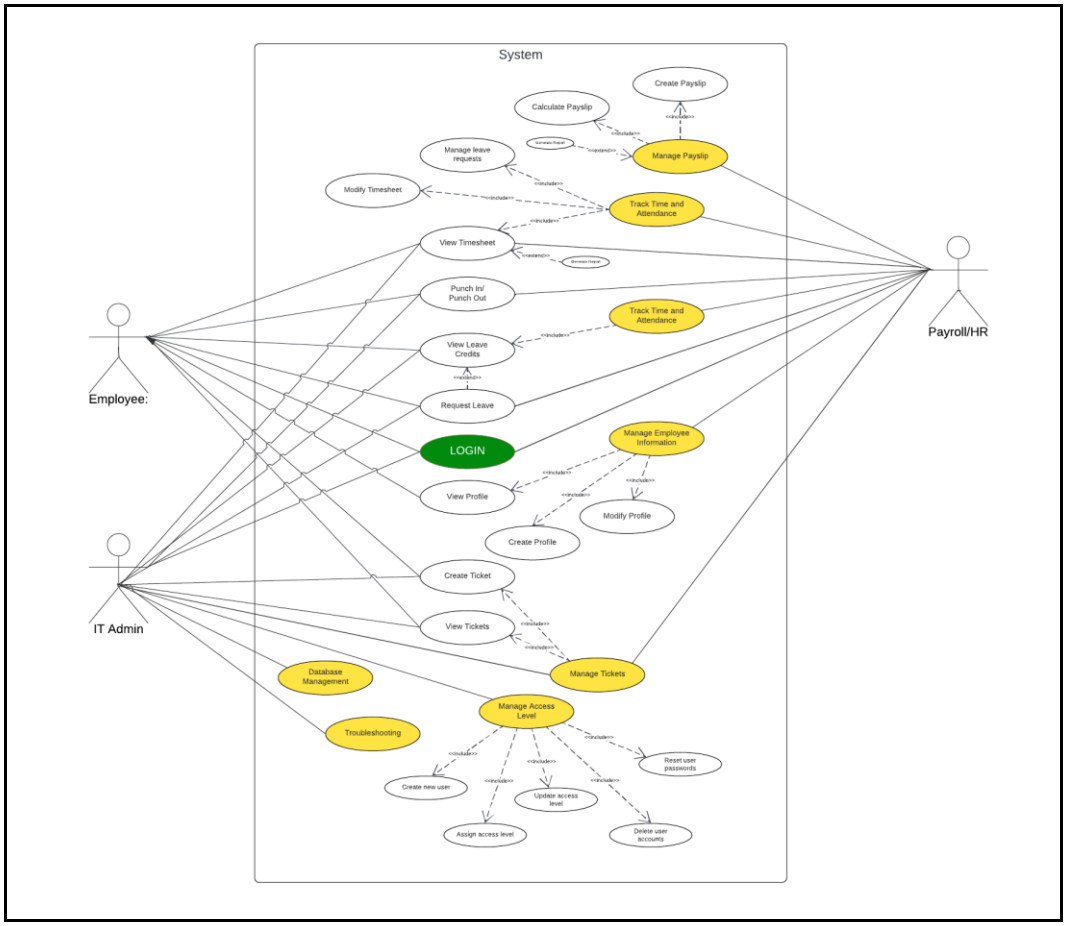
3.3. Leave Applications

<If your system includes leave management, explain how users can apply for leave. This can be a straightforward process or a more elaborate one, depending on your project's complexity.>

4. Technical Information

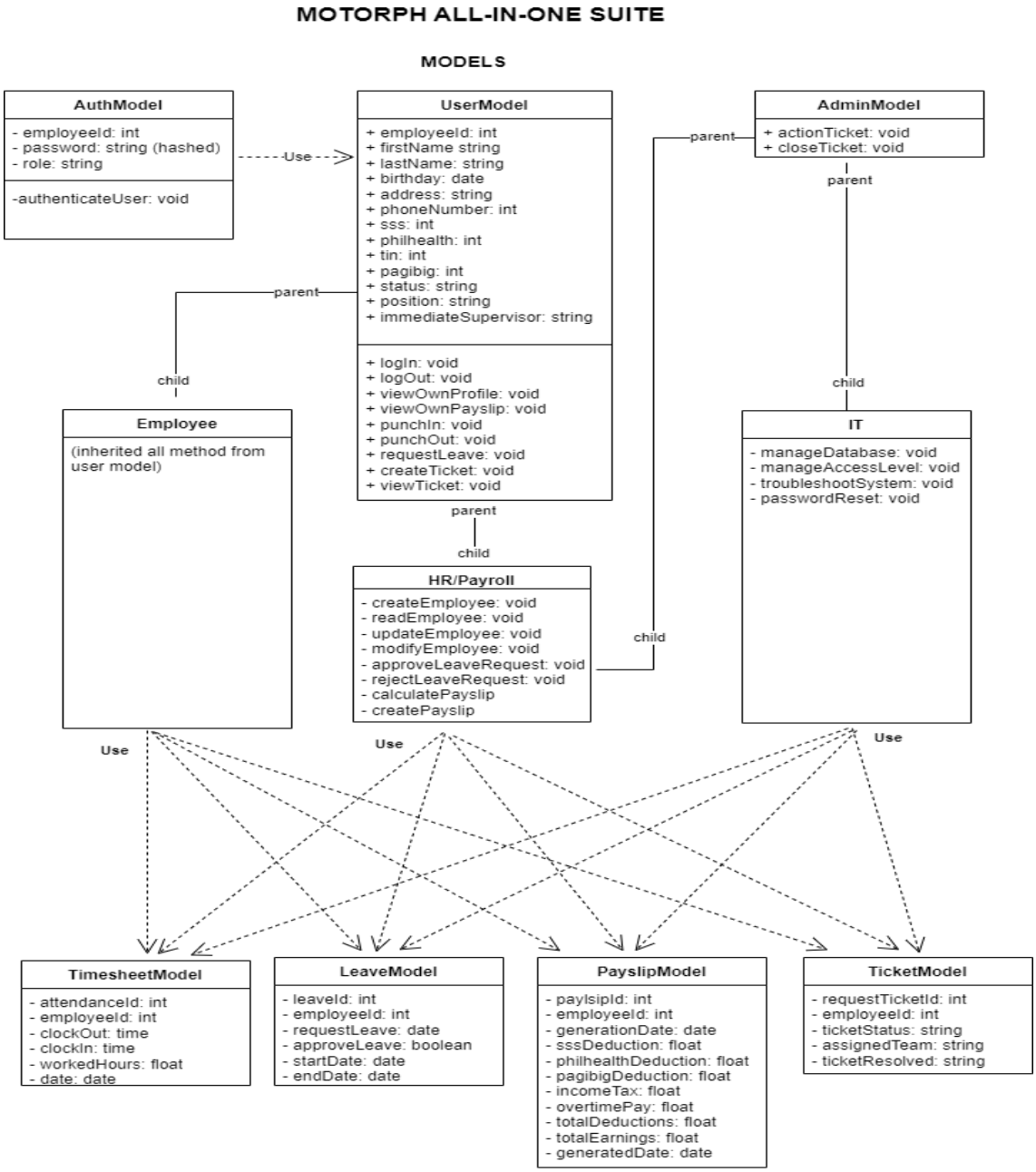
4.1. Use Case Diagram

The use case diagram offers a skeletal structure for a payroll system built using object-oriented principles. It identifies key classes like Employee, Payroll/HR, IT Admin, and Login. The diagram outlines inheritance for the Employee class and an aggregation relationship between Login and Employee. This indicates that Employee objects can exist independent of login functionality, while Login objects likely reference specific Employee data. Overall, the diagram provides a starting point for understanding potential object interactions within the payroll system design.



4.2. Class Diagram

The class diagram provides a foundational understanding of how object-oriented principles can be applied to design a payroll system. It highlights the core classes involved in employee data management, timekeeping, leave requests, payslip generation, payroll processing, system administration, and login functionalities. The inheritance and associations provide insights into potential collaborations between these classes to achieve the system's functionalities.



4.3. Testing