# MÔ TẢ

|  |
| --- |
| Xây dựng trang web quản lý timesheet với các mô tả nghiệp vụ như sau:  1.1. Mỗi timesheet có các nội dung như sau:   * Ngày (ví dụ 2018-08-14) * Các công việc đã làm trong ngày (multiple line). Mỗi line có dạng:   + Task ID (nếu không có thì để là "N/A")   + Nội dung task   + Thời gian sử dụng * Các khó khăn gặp phải (text) * Các dự định sẽ làm trong ngày tiếp theo (text)   1.2. Mô tả use case của nhân viên   * Nhân viên có trang cá nhân, có thể sửa đổi các thông tin sau:   + password, avatar, description * Mỗi ngày nhân viên cần phải vào trang web để tạo timesheet * Hệ thống sẽ tự động ghi nhận:   + Số lần nhân viên đăng ký timesheet trong tháng (kể cả làm đúng giờ hay làm bổ sung)   + Số lần mỗi nhân viên chậm làm timesheet theo tháng * Sau khi tạo timesheet, nhân viên có quyền sửa nội dung timesheet * Nhân viên có thể truy cập trang danh sách, liệt kê các timesheet của mình theo tuần / tháng.   **Yêu cầu**  2.1. Đọc hiểu mô tả nghiệp vụ. Vẽ biểu đồ use case cơ bản.  2.2. Xây dựng cấu trúc database. Tạo file GoogleSheet theo mẫu.  <https://docs.google.com/spreadsheets/d/1yo7DOcFnZ939w4OnVug2Pgaa1EVY7XquwHXn3pt_eZ8/edit?usp=sharing>  2.3. Tạo code base cơ bản:   * PHP 8.1, MariaDB * Laravel 10 * migrations * layouts   2.4. Code chức năng nhân viên   * Màn hình nhân viên login * Màn hình tạo timesheet * Màn hình danh sách timesheet * Màn hình view chi tiết timesheet * Màn hình chỉnh sửa timesheet   2.5. Các chức năng nâng cao   * Chức năng phân quyền trong hệ thống (Yêu cầu tìm hiểu về Policy trong Laravel):   + Tạo thêm field role trong table users, định nghĩa quyền cho các user trong hệ thống như sau   A screenshot of a phone  Description automatically generated   * Khi user truy cập vào các page mà mình ko có quyền truy cập thì xử lý lỗi 403      * Thêm màn hình hiển thị list time sheet bằng calendar (Yêu cầu sử dụng 1 plugin js) * Sử dụng bootstrap 5 để customize lại giao diện của hệ thống |

THỰC HIỆN

### 04/06/2024- Dấu thời gian.

1. Dataset: [TimeSheetProjectDB - Google スプレッドシート](https://docs.google.com/spreadsheets/d/1f6Q7JY2kzSRoVI1VYjr4jo2Q7YQ1st3VYpY1OlrVqxw/edit#gid=0)
2. Mã nguồn: [Github](https://github.com/ds-gioi-duong/training_intern)
3. Công nghệ sử dụng

* PHP 8.1, MariaDB

|  |
| --- |
| **PHP 8.1**  PHP (Hypertext Preprocessor) là một ngôn ngữ kịch bản máy chủ mã nguồn mở được thiết kế đặc biệt cho phát triển web. PHP 8.1 là một phiên bản của ngôn ngữ PHP, được phát hành vào tháng 11 năm 2021. Phiên bản này mang lại nhiều cải tiến và tính năng mới, bao gồm:   1. **Fibers**: Cung cấp cơ chế quản lý luồng nhẹ giúp cho việc viết mã bất đồng bộ trở nên dễ dàng hơn. 2. **Enums**: Giới thiệu kiểu dữ liệu liệt kê (enumerations), giúp đơn giản hóa việc làm việc với các tập hợp giá trị cố định. 3. **Readonly Properties**: Cho phép khai báo các thuộc tính chỉ đọc, không thể thay đổi sau khi được khởi tạo. 4. **Intersection Types**: Hỗ trợ kiểu giao nhau, cho phép khai báo biến hoặc tham số với nhiều kiểu kết hợp. 5. **Performance Improvements**: Cải thiện hiệu suất so với các phiên bản trước. |

|  |
| --- |
| MariaDB là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở, được phát triển bởi những người đã tạo ra MySQL. MariaDB được sử dụng để lưu trữ dữ liệu trong các bảng, với khả năng xử lý các giao dịch, truy vấn, và quản lý dữ liệu phức tạp. MariaDB nổi tiếng với:   1. **Tương thích với MySQL**: MariaDB tương thích với MySQL về API và các câu lệnh SQL, giúp việc chuyển đổi giữa hai hệ thống trở nên dễ dàng. 2. **Hiệu suất cao**: MariaDB thường có các cải tiến về hiệu suất và tối ưu hóa so với MySQL. 3. **Tính năng nâng cao**: Bao gồm các engine lưu trữ tiên tiến như Aria, XtraDB, và Cassandra, hỗ trợ tốt hơn cho các ứng dụng lớn và yêu cầu hiệu năng cao. 4. **Bảo mật và ổn định**: Thường xuyên được cập nhật và bảo trì để đảm bảo tính bảo mật và ổn định của hệ thống.   Cả PHP và MariaDB thường được sử dụng cùng nhau trong phát triển web, với PHP đảm nhiệm phần logic ứng dụng và MariaDB đảm nhiệm phần lưu trữ dữ liệu. Cặp đôi này thường xuất hiện trong các ngăn xếp công nghệ phổ biến như LAMP (Linux, Apache, MySQL/MariaDB, PHP).  Config:  root-12345678  MariaDB: 127.0.0.1:3306    Cài đặt xong truy cập heidiSQL |

|  |
| --- |
| **Migrations**  Migrations là một khái niệm trong phát triển phần mềm, đặc biệt phổ biến trong quản lý cơ sở dữ liệu. Nó đề cập đến quá trình thay đổi cấu trúc cơ sở dữ liệu theo một cách có kiểm soát và có thể theo dõi được. Các migration giúp bạn duy trì và cập nhật cấu trúc cơ sở dữ liệu một cách nhất quán trên các môi trường phát triển, thử nghiệm, và sản xuất.  Một số đặc điểm và lợi ích của migrations:   1. **Quản lý thay đổi**: Ghi lại và quản lý mọi thay đổi cấu trúc cơ sở dữ liệu như tạo bảng, thêm cột, hay thay đổi kiểu dữ liệu. 2. **Khả năng hoàn nguyên (Rollback)**: Cho phép hoàn nguyên lại thay đổi nếu có lỗi xảy ra, giúp đảm bảo tính nhất quán và bảo mật. 3. **Dễ dàng chia sẻ**: Các migration được lưu trữ dưới dạng mã nguồn, có thể được chia sẻ và áp dụng bởi các thành viên trong nhóm phát triển. 4. **Tự động hóa**: Các công cụ migration thường có khả năng tự động áp dụng thay đổi, giúp giảm thiểu công việc thủ công và lỗi do con người.   Ví dụ, các framework như Laravel (PHP), Django (Python), và Ruby on Rails (Ruby) đều cung cấp hệ thống migration tích hợp để quản lý cơ sở dữ liệu. |

* layouts

|  |
| --- |
| Layouts là một khái niệm trong thiết kế giao diện người dùng (UI) và phát triển web. Nó đề cập đến cách sắp xếp và tổ chức các phần tử trên một trang web hoặc trong một ứng dụng để tạo ra một giao diện nhất quán và dễ sử dụng.  Một số khía cạnh của layouts:   1. **Consistency**: Giúp tạo ra sự nhất quán trong cách trình bày thông tin và tương tác, giúp người dùng dễ dàng sử dụng và điều hướng. 2. **Responsiveness**: Đảm bảo rằng giao diện hiển thị tốt trên các kích thước màn hình khác nhau, từ điện thoại di động đến máy tính để bàn. 3. **Hierarchy and Flow**: Sắp xếp các phần tử để hướng dẫn người dùng qua các bước và thông tin một cách logic và trực quan. 4. **Aesthetics**: Tạo ra một giao diện hấp dẫn về mặt thị giác, kết hợp màu sắc, kiểu chữ và khoảng cách một cách hài hòa.   Trong phát triển web, frameworks như Bootstrap, Foundation, và các hệ thống grid thường được sử dụng để tạo ra layouts dễ dàng và nhanh chóng. Trong các framework MVC (Model-View-Controller) như Laravel (PHP) hoặc Ruby on Rails (Ruby), layouts cũng được sử dụng để tạo ra các trang chủ đạo mà các trang con kế thừa, giúp duy trì một giao diện nhất quán. |

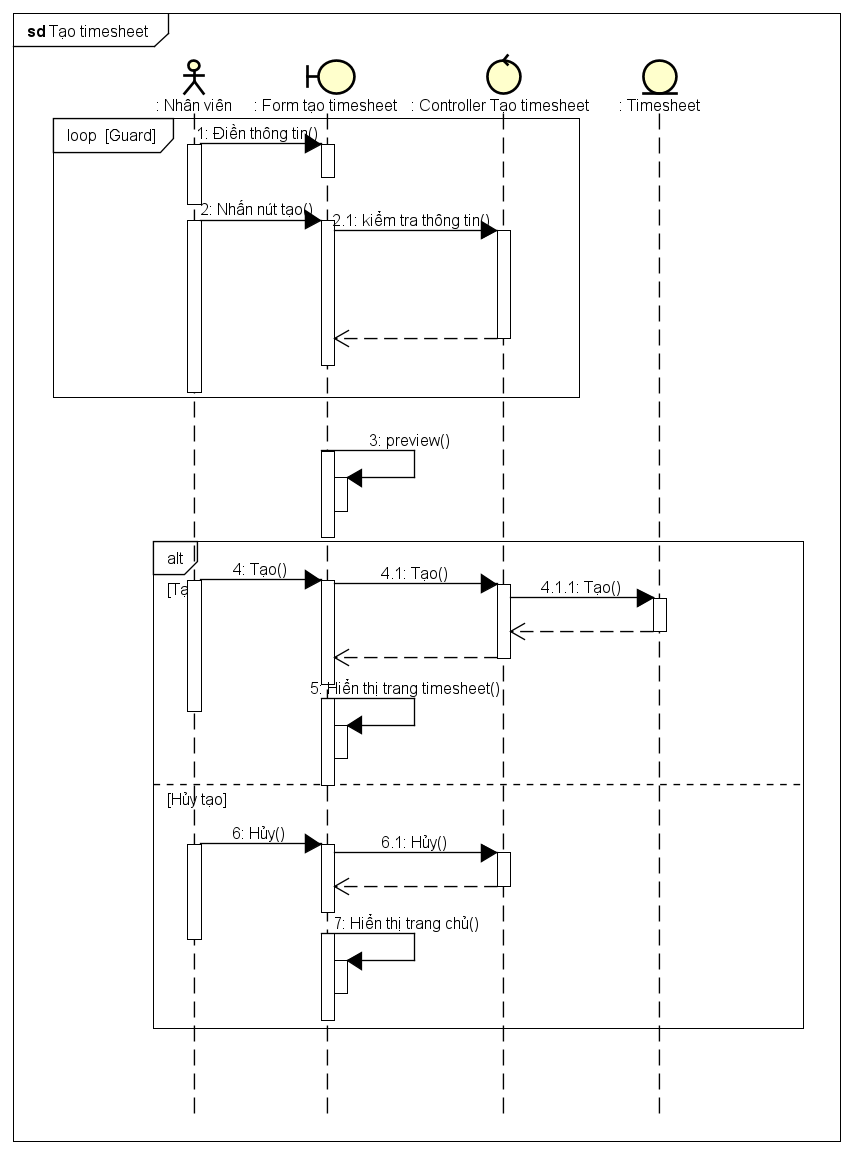
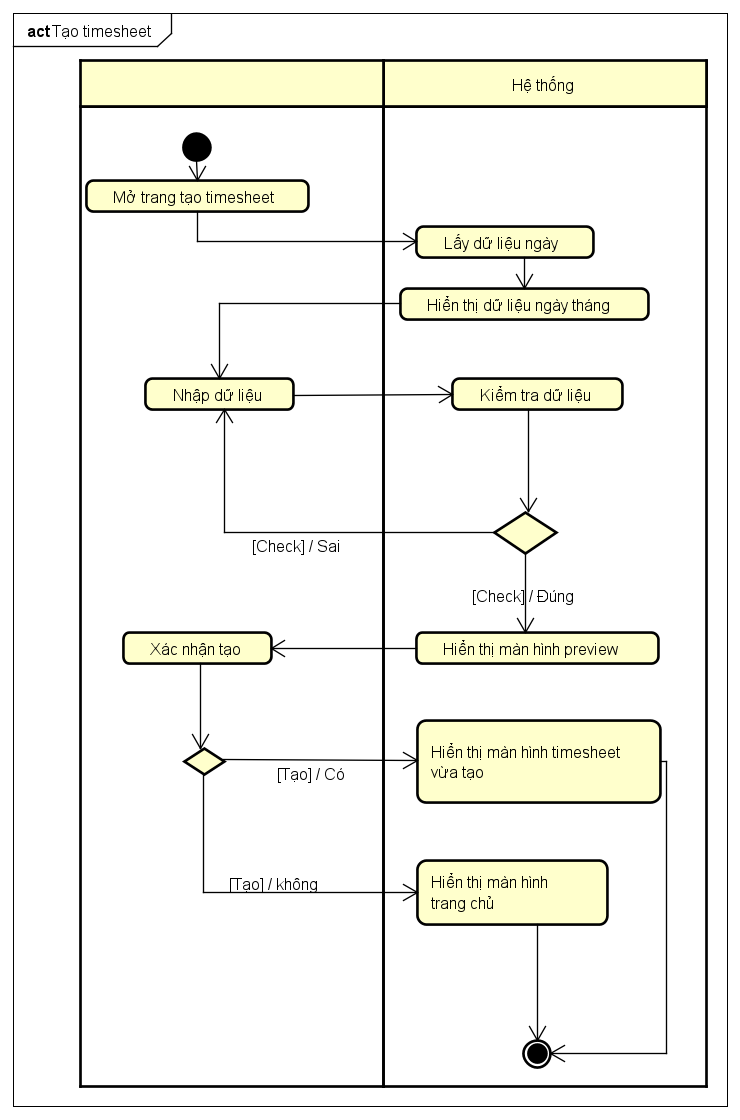
1. Cài đặt môi trường
   1. MariaDB: Config:

* root-12345678
* MariaDB: 127.0.0.1:3306
  1. PHP : cài xampp -> set env var cho php
     1. Cách 1: Cấu hình chạy trên xampp
* Chọn : httpd.config để cấu hình
* VD: thay đổi port, thay đổi DocumentRoot , <Directory> ( nơi lưu mã nguồn)
  + 1. Cách 2: Chạy với dòng lệnh: ` php -S localhost:8000`
  1. UI

1. Phát triển:
   1. Thu thập yêu cầu
      1. Xây dựng timesheet với các mô tả nghiệp vụ như sau:

* Ngày ( 2024-06-20)
* Các công việc đã làm trong ngày ( muitiple line). Mỗi line có dạng:
  + Task ID ( nếu không có thì “N/A”)
  + Nội dung task
  + Thời gian sử dụng
* Các khó khăn gặp phải (text)
* Các dự định sẽ làm trong ngày tiếp theo ( text)
  + 1. Mô tả use case của nhân viên
* Nhân viên có trang cá nhân, có thể sửa đổi thông tin:
  + Password, avatar, description
* Mỗi ngày nhân viên cần phải vào trang web để tạo time sheet
* Hệ thống sẽ tự động ghi nhận:
  + Số lần nhân viên đăng ký timesheet trong tháng ( kể cả làm đúng giờ hay làm bổ sung)
  + Số lần mỗi nhân viên chậm làm timesheet theo tháng
* Sau khi tạo timesheet , nhân viên có quyền sửa nội dung timesheet
* A diagram of a person's work flow

  Description automatically generatedNhân viên có thể truy cập trang danh sách, liệt kê các timesheet của mình theo tuần/ tháng.
  + 1. Usecase Phân rã
       1. Quản lý user
       2. Sửa thông tin cá nhân
  1. Phân tích thiết kế tổng quan
     1. Usecase: Đăng nhập
     2. Usecase: Tạo timesheet



* + 1. Usecase: Sửa timesheet

A diagram with text and black dots

Description automatically generated with medium confidence

A screenshot of a diagram

Description automatically generated

* + 1. Usecase: Xem danh sách timesheet
    2. Usecase: Xem chi tiết timesheet
    3. Usecase: Sửa thông tin cơ bản
  1. Thiết kế giao diện
* Màn hình nhân viên login
* Màn hình tạo timesheet
* Màn hình danh sách timesheet
* Màn hình view chi tiết timesheet
* Màn hình chỉnh sửa timesheet
  1. Thiết kế chi tiết lớp
  2. Mô hình hóa dữ liệu
  3. Kiểm thử
  4. Triển khai và vận hành

# TEST MÔI TRƯỜNG